# Manuel PHP 4.3.0 − Version Française

http://www.nexen.net

# Sommaire

**Manuel PHP....1**
1.1 Les auteurs....1
1.2 L'éditeur....1
1.3 Note légale....1

**2 Préface....2**

**3 Comment commencer....3**
3.1 Configuration....3
3.1.1 Le fichier de configuration....3
3.1.2 Comment modfier la configuration....3
3.1.3 Directives de configuration....5
3.2 Introduction....10
3.2.1 Qu'est ce que PHP?....10
3.2.2 Que peut faire PHP?....11
3.3 Une introduction à PHP....13
3.3.1 Le nécessaire....13
3.3.2 Votre première page PHP....13
3.3.3 Trucs pratiques....15
3.3.4 Utiliser un formulaire....16
3.3.5 Utiliser des codes anciens avec les nouvelles versions de PHP....17
3.3.6 Et après?....18
3.4 Installation....18
3.4.1 Considérations générales sur l'installation....18
3.4.2 Installation sous UNIX....19
3.4.3 Installation sous Linux/Unix....20
3.4.4 Installation sous HP−UX....20
3.4.5 Installation sous Solaris....21
3.4.6 Installations Unix/OpenBSD....22
3.4.7 Installation sous Unix et Mac OS X....23
3.4.8 Liste des options de configuration diverses....24
3.4.9 Installation sous Windows....28
3.4.10 Installation avec le serveur Apache....36
3.4.11 Installation avec les serveurs Apache 2.0....39
3.4.12 CGI/ Installation pour exécution en ligne de commande....41
3.4.13 Installation avec les serveurs fhttpd....42
3.4.14 Installation sur le serveur Caudium....42
3.4.15 Installation avec les serveurs IIS/PWS....43
3.4.16 Installation sous Netscape et iPlanet Enterprise Serveur....45
3.4.17 Installation pour les serveurs OmniHTTPd....49
3.4.18 Installation Oreilly Website Pro Server....50
3.4.19 Serveurs Sambar....50
3.4.20 Installation Xitami....50
3.4.21 Autres serveurs web....51
3.4.22 Des problèmes?....51

**4 Référence....52**
4.1 La syntaxe de base....52
4.1.1 Passer du HTML au PHP....52
4.1.2 Le séparateur d'instructions....53
4.1.3 Commentaires....53
4.2 Les types....54

# Sommaire

**4 Référence**
4.2.1 Introduction....54
4.2.2 Booléens....55
4.2.3 Entiers....56
4.2.4 Les nombres à virgule flottante....58
4.2.5 Les chaînes de caractères....59
4.2.6 Les tableaux....64
4.2.7 Les objets....70
4.2.8 Ressources....71
4.2.9 La valeur NULL....71
4.2.10 Pseudo−types utilisés dans cette documentation....72
4.2.11 Définition du type....72
4.3 Les variables....74
4.3.1 Essentiel....75
4.3.2 Variables prédéfinies....76
4.3.3 Portée des variables....77
4.3.4 Les variables dynamiques....81
4.3.5 Variables externes à PHP....81
4.4 Les constantes....84
4.4.1 Syntaxe....84
4.4.2 Constantes prédéfinies....85
4.5 Les expressions....85
4.6 Les opérateurs....88
4.6.1 La précédence des opérateurs....88
4.6.2 Les opérateurs arithmétiques....88
4.6.3 Les opérateurs d'assignation....89
4.6.4 Opérateurs sur les bits....89
4.6.5 Opérateurs de comparaison....90
4.6.6 Opérateur de contrôle d'erreur....90
4.6.7 Opérateur d'exécutions....91
4.6.8 Opérateurs d'incrémentation/Décrémentation....91
4.6.9 Les opérateurs logiques....92
4.6.10 Opérateurs de chaînes....92
4.6.11 Opérateurs de tableaux....92
4.7 Les structures de contrôle....93
4.7.1 if....93
4.7.2 else....94
4.7.3 elseif....94
4.7.4 Syntaxe alternative....94
4.7.5 while....95
4.7.6 do..while....96
4.7.7 for....96
4.7.8 foreach....97
4.7.9 break....99
4.7.10 continue....99
4.7.11 switch....100
4.7.12 declare....102
4.7.13 return....103
4.7.14 require....103
4.7.15 include....104
4.7.16 require_once....106
4.7.17 include_once....107

# Sommaire

**4 Référence**
4.8 Les fonctions....108
4.8.1 Les fonctions....108
4.8.2 Les arguments de fonction....109
4.8.3 Les valeurs de retour....111
4.8.4 Variables fonctions....111
4.8.5 Fonctions internes....112
4.9 Les classes et les objets....112
4.9.1 Les classes : class....112
4.9.2 extends : héritage....114
4.9.3 Constructor : constructeur....114
4.9.4 Opérateur ::....116
4.9.5 parent....117
4.9.6 Sauvegarde d'objets − cas des sessions....117
4.9.7 Les fonctions magiques __sleep et __wakeup....118
4.9.8 Références dans un constructeur....119
4.9.9 Comparer des objets en PHP 4....120
4.9.10 Comparer des objets en PHP 5....122
4.10 Les références....123
4.10.1 Qu'est ce qu'une référence?....123
4.10.2 Que font les références ?....124
4.10.3 Ce que les références ne sont pas....124
4.10.4 Passage par référence....125
4.10.5 Retourner des références....125
4.10.6 Détruire une référence....126
4.10.7 Repérer une référence....126

**5 Caractéristiques....127**
5.1 Identification HTTP avec PHP....127
5.2 Cookies....128
5.3 Gestion des chargements de fichier....129
5.3.1 Chargements de fichiers par méthode POST....129
5.3.2 Explication sur les messages d'erreurs de chargement de fichiers....131
5.3.3 Erreurs classiques....131
5.3.4 Télécharger plusieurs fichiers simultanément....131
5.3.5 Chargement par méthode PUT....132
5.4 Utilisation des fichiers à distance....133
5.5 Gestion des connexions....134
5.6 Connexions persistantes aux bases de données....135
5.7 Safe mode....136
5.7.1 Securité et Safe Mode....136
5.7.2 Fonctions désactivées par le Safe Mode....139
5.8 Utiliser PHP en ligne de commande....141

**6 Index des fonctions....149**
6.1 Apache....149
6.1.1 Introduction....149
6.1.2 Installation....149
6.1.3 Configuration à l'exécution....149
6.1.4 Types de ressources....149
6.1.5 Constantes prédefinies....149
6.1.7 apache_get_version() : Retourne la version d'Apache....150

# Sommaire

**6 Index des fonctions**
6.1.8 apache_lookup_uri() : Effectue une requête partielle pour l'URI spécifiée et renvoie toutes les informations....150
6.1.9 apache_note() : Affiche ou affecte le paramètre "apache request notes"....151
6.1.10 apache_request_headers() : Lit tous les en−têtes HTTP de la requête....151
6.1.11 apache_response_headers() : Lit tous les en−têtes de réponse HTTP....152
6.1.12 apache_setenv() : Modifie une variable de subprocess_env Apache....152
6.1.13 ascii2ebcdic() : Transforme une chaîne ASCII en EBCDIC....152
6.1.14 ebcdic2ascii() : Transforme une chaîne EBCDIC en ASCII....152
6.1.15 getallheaders() : Lit tous les en−têtes de la requête HTTP....152
6.1.16 virtual() : Effectue une sous−requête Apache....153
6.2 Tableaux....154
6.2.1 Introduction....154
6.2.2 Pré−requis....154
6.2.3 Installation....154
6.2.4 Configuration à l'exécution....154
6.2.5 Types de ressources....154
6.2.6 Constantes prédefinies....154
6.2.7 Voir aussi....155
6.2.9 array_chunk() : Sépare un tableau en tableaux de taille inférieure....156
6.2.10 array_combine() : Crée un tableau à partir de deux autres tableaux....158
6.2.11 array_count_values() : Compte le nombre de valeurs dans un tableau....158
6.2.12 array_diff_assoc() : Calcule la différence de deux tableaux, en prenant en compte les clés....159
6.2.13 array_diff_uassoc() : Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction utilisateur....159
6.2.14 array_diff() : Calcule la différence entre deux tableaux....160
6.2.15 array_fill() : Remplis un tableau avec une même valeur....161
6.2.16 array_filter() : Filtre les éléments d'un tableau....161
6.2.17 array_flip() : Remplace les clés par les valeurs, et les valeurs par les clés....162
6.2.18 array_intersect_assoc() : Calcule l'intersection de deux tableaux avec des tests sur les index....163
6.2.19 array_intersect() : Calcule l'intersection de tableaux....163
6.2.20 array_key_exists() : Vérifie si une clé existe dans un tableau....164
6.2.21 array_keys() : Retourne toutes les clés d'un tableau....164
6.2.22 array_map() : Applique sur fonction sur des tableaux....165
6.2.23 array_merge_recursive() : Combine plusieurs tableaux ensembles, récursivement....168
6.2.24 array_merge() : Fusionne plusieurs tableaux....168
6.2.25 array_multisort() : Tri multi−dimensionnel....169
6.2.26 array_pad() : Complète un tableau jusqu'à la longueur spécifiée, avec une valeur....170
6.2.27 array_pop() : Dépile un élément de la fin d'un tableau....171
6.2.28 array_push() : Empile un ou plusieurs éléments à la fin d'un tableau....171
6.2.29 array_rand() : Prend une ou plusieurs valeurs, au hasard dans un tableau....172
6.2.30 array_reduce() : Réduit itérativement un tableau....172
6.2.31 array_reverse() : Inverse l'ordre des éléments d'un tableau....173
6.2.32 array_search() : Recherche dans un tableau la clé associée à une valeur....174
6.2.33 array_shift() : Dépile un élément au début d'un tableau....174
6.2.34 array_slice() : Extrait une portion de tableau....175
6.2.35 array_splice() : Efface et remplace une portion de tableau....175
6.2.36 array_sum() : Calcule la somme des valeurs du tableau....176

# Sommaire

**6 Index des fonctions**
6.2.37 array_udiff_assoc() : Calcule la différence entre des tableaux avec vérification sur les index. Les données sont comparées en utilisant une fonction callback....177
6.2.38 array_udiff_uassoc() : Calcule la différence de deux tableaux associatifs, avec une fonction utilisateur....178
6.2.39 array_udiff() : Calcule la différence entre deux tableaux en utilisant une fonction callback pour comparer les données....179
6.2.40 array_unique() : Dédoublonne un tableau....180
6.2.41 array_unshift() : Empile un ou plusieurs éléments au début d'un tableau....180
6.2.42 array_values() : Retourne les valeurs d'un tableau....181
6.2.43 array_walk() : Exécute une fonction sur chacun des membres d'un tableau....181
6.2.44 array() : Crée un tableau....182
6.2.45 arsort() : Trie un tableau en ordre inverse....184
6.2.46 asort() : Trie un tableau en ordre....184
6.2.47 compact() : Crée un tableau contenant les variables et leur valeur....185
6.2.48 count() : Compte le nombre d'éléments d'une variable....185
6.2.49 current() : Transforme une variable en tableau....186
6.2.50 each() : Retourne chaque paire clé/valeur d'un tableau....186
6.2.51 end() : Positionne le pointeur de tableau en fin de tableau....187
6.2.52 extract() : Importe les variables dans la table des symboles....188
6.2.53 in_array() : Indique si une valeur appartient à un tableau....189
6.2.54 key() : Retourne une clé d'un tableau associatif....190
6.2.55 krsort() : Trie un tableau en sens inverse et suivant les clés....190
6.2.56 ksort() : Trie un tableau suivant les clés....191
6.2.57 list() : Transforme une liste de variables en tableau....191
6.2.58 natcasesort() : Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse....192
6.2.59 natsort() : Tri d'un tableau avec l'algorithme à "ordre naturel"....193
6.2.60 next() : Avance le pointeur interne d'un tableau....194
6.2.61 pos() : Retourne l'élément courant d'un tableau....194
6.2.62 prev() : Recule le pointeur courant de tableau....194
6.2.63 range() : Crée un tableau contenant un intervalle d'éléments....195
6.2.64 reset() : Remet le pointeur interne de tableau au début....195
6.2.65 rsort() : Trie en ordre inverse....196
6.2.66 shuffle() : Mélange les éléments d'un tableau....196
6.2.67 sizeof() : Alias de count....196
6.2.68 sort() : Trie un tableau....197
6.2.69 uasort() : Trie un tableau en utilisant une fonction de comparaison personnalisée....197
6.2.70 uksort() : Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur....197
6.2.71 usort() : Trie un tableau en utilisant une fonction de comparaison personnalisée....198
6.3 Calendrier....200
6.3.1 Introduction....200
6.3.2 Installation....200
6.3.3 Configuration à l'exécution....200
6.3.4 Types de ressources....200
6.3.5 Constantes prédefinies....200
6.3.7 cal_from_jd() : Convertit le nombre de jours Julien en un calendrier et retourne des détails....201

# Sommaire

**6 Index des fonctions**
6.3.8 cal_info() : Retourne des détails sur un calendrier....201
6.3.9 cal_to_jd() : Convertit un calendrier en nombre de jours Julien....202
6.3.10 easter_date() : Retourne un timestamp UNIX pour Pâques, à minuit....202
6.3.11 easter_days() : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée....202
6.3.12 frenchtojd() : Convertit une date du calendrier français républicain en nombre de jours du calendrier Julien....203
6.3.13 gregoriantojd() : Convertit une date grégorienne en nombre de jours du calendrier Julien....203
6.3.14 jddayofweek() : Retourne le numéro du jour de la semaine....204
6.3.15 jdmonthname() : Retourne le nom du mois....204
6.3.16 jdtofrench() : Convertit le nombre de jours du calendrier Julien en date du calendrier français républicain....204
6.3.17 jdtogregorian() : Convertit le nombre de jours du calendrier Julien en date grégorienne....205
6.3.18 jdtojewish() : Convertit le nombre de jours du calendrier Julien en date du calendrier juif....205
6.3.19 jdtojulian() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien....205
6.3.20 jdtounix() : Convertit un nombre de jours Julien en timestamp UNIX....205
6.3.21 jewishtojd() : Convertit une date du calendrier juif en nombre de jours du calendrier Julien....205
6.3.22 juliantojd() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien....206
6.3.23 unixtojd() : Convertit un timestamp UNIX en nombre de jours Julien....206
6.4 Classes et Objets....207
6.4.1 Introduction....207
6.4.2 Pré−requis....207
6.4.3 Installation....207
6.4.4 Configuration à l'exécution....207
6.4.5 Types de ressources....207
6.4.6 Constantes prédefinies....207
6.4.7 Exemples....207
6.4.9 call_user_method() : Appelle une méthode utilisateur d'un objet....209
6.4.10 class_exists() : Vérifie qu'une classe a été définie....210
6.4.11 get_class_methods() : Retourne les noms des méthodes d'une classe....210
6.4.12 get_class_vars() : Retourne les valeurs par défaut des attributs d'une classe....210
6.4.13 get_class() : Retourne la classe d'un objet....211
6.4.14 get_declared_classes() : Liste toutes les classes définies....212
6.4.15 get_object_vars() : Retourne un tableau associatif des propriétés d'un objet....212
6.4.16 get_parent_class() : Retourne le nom de la classe d'un objet....213
6.4.17 is_a() : Retourne TRUE si un objet a pour parent une classe donnée....214
6.4.18 is_subclass_of() : Détermine si un objet est une sous−classe....214
6.4.19 method_exists() : Vérifie que la méthode existe pour une classe....214
6.5 CURL....215
6.5.1 Introduction....215
6.5.2 Pré−requis....215
6.5.3 Installation....215
6.5.4 Constantes prédefinies....215

# Sommaire

**6 Index des fonctions**
6.5.5 Exemples....218
6.5.7 curl_errno() : Retourne un entier représentant le dernier message d'erreur CURL....219
6.5.8 curl_error() : Retourne une chaîne contenant le dernier message d'erreur CURL....219
6.5.9 curl_exec() : Exécute une session CURL....219
6.5.10 curl_getinfo() : Lit les informations détaillant un transfert CURL....219
6.5.11 curl_init() : Initialise une session CURL....221
6.5.12 curl_multi_add_handle() : Ajoute une ressource cURL à un cURL multiple....221
6.5.13 curl_multi_close() : Termine un jeu de transferts cURL....221
6.5.14 curl_multi_exec() : Exécute les sous−requêtes d'un cURL....221
6.5.15 curl_multi_getcontent() : Retourne le contenu obtenu avec l'option CURLOPT_RETURNTRANSFER....222
6.5.16 curl_multi_info_read() : Lit les informations sur les transferts actuels....222
6.5.17 curl_multi_init() : Retourne un nouveau cURL multiple....222
6.5.18 curl_multi_remove_handle() : Retire un cURL multiple d'un jeu de cURL....222
6.5.19 curl_multi_select() : Retourne les sockets associées à cURL, pour utilisation....222
6.5.20 curl_setopt() : Modifie une option de transfert CURL....223
6.5.21 curl_version() : Retourne la version courante de CURL....225
6.6 Dates et heures....226
6.6.1 Introduction....226
6.6.2 Pré−requis....226
6.6.3 Installation....226
6.6.4 Configuration à l'exécution....226
6.6.5 Types de ressources....226
6.6.6 Constantes prédefinies....226
6.6.8 date() : Formate une date/heure locale....227
6.6.9 getdate() : Retourne la date/heure....229
6.6.10 gettimeofday() : Retourne l'heure actuelle....229
6.6.11 gmdate() : Formate une date/heure GMT/CUT....229
6.6.12 gmmktime() : Retourne le timestamp UNIX d'une date GMT....230
6.6.13 gmstrftime() : Formate une date/heure GMT/CUT en fonction des paramétrages locaux....230
6.6.14 localtime() : Lit l'heure locale....230
6.6.15 microtime() : Retourne le timestamp UNIX actuel avec microsecondes....231
6.6.16 mktime() : Retourne le timestamp UNIX d'une date....231
6.6.17 strftime() : Formate une date/heure locale avec les options locales....232
6.6.18 strtotime() : Transforme un texte anglais en timestamp....234
6.6.19 time() : Retourne le timestamp UNIX actuel....235
6.7 dBase....236
6.7.1 Introduction....236
6.7.2 Installation....236
6.7.3 Configuration à l'exécution....236
6.7.4 Types de ressources....236
6.7.5 Constantes prédefinies....236
6.7.7 dbase_close() : Ferme une base dBase....237
6.7.8 dbase_create() : Crée une base de données dBase....237
6.7.9 dbase_delete_record() : Efface un enregistrement dans une base dBase....238
6.7.10 dbase_get_header_info() : Affichage des informations d'en−tête d'une base de données dBase....238

# Sommaire

**6 Index des fonctions**
6.7.11 dbase_get_record_with_names() : Lit un enregistrement dans une base dBase, sous la forme d'un tableau associatif....239
6.7.12 dbase_get_record() : Lit un enregistrement dans une base dBase....239
6.7.13 dbase_numfields() : Compte le nombre de champs d'une base dBase....239
6.7.14 dbase_numrecords() : Compte le nombre d'enregistrements dans une base dBase....239
6.7.15 dbase_open() : Ouvre une base dBase....240
6.7.16 dbase_pack() : Compacte une base dBase....240
6.7.17 dbase_replace_record() : Remplace un enregistrement dans une base dBase....240
6.8 Accès aux dossiers....241
6.8.1 Introduction....241
6.8.2 Pré−requis....241
6.8.3 Installation....241
6.8.4 Configuration à l'exécution....241
6.8.5 Types de ressources....241
6.8.6 Constantes prédefinies....241
6.8.7 Voir aussi....241
6.8.9 chroot() : Change le dossier racine....242
6.8.10 closedir() : Ferme le pointeur sur le dossier....242
6.8.11 dir() : Classe dossier....242
6.8.12 getcwd() : Retourne le dossier de travail....242
6.8.13 opendir() : Ouvre un dossier, et récupère un pointeur dessus....243
6.8.14 readdir() : Lit une entrée du dossier....243
6.8.15 rewinddir() : Retourne à la première entrée du dossier....244
6.8.16 scandir() : Liste les fichiers et dossier dans un dossier....244
6.9 Gestion des erreurs....246
6.9.1 Introduction....246
6.9.2 Pré−requis....246
6.9.3 Installation....246
6.9.4 Configuration à l'exécution....246
6.9.5 Constantes prédefinies....248
6.9.6 Exemples....249
6.9.7 Voir aussi....250
6.9.9 debug_print_backtrace() : Affiche une backtrace....251
6.9.10 error_log() : Stocke un message d'erreur....251
6.9.11 error_reporting() : Fixe le niveau de rapport d'erreurs PHP....251
6.9.12 restore_error_handler() : Réactive l'ancienne fonction de gestion des erreurs....252
6.9.13 set_error_handler() : Choisi une fonction utilisateur comme gestionnaire d'erreurs....252
6.9.14 trigger_error() : Déclenche une erreur utilisateur....254
6.9.15 user_error() : Alias de trigger_error....255
6.10 Système de fichiers....256
6.10.1 Introduction....256
6.10.2 Pré−requis....256
6.10.3 Installation....256
6.10.4 Configuration à l'exécution....256
6.10.5 Types de ressources....257
6.10.6 Constantes prédefinies....257
6.10.7 Voir aussi....257

# Sommaire

**6 Index des fonctions**
6.10.9 chgrp() : Change le groupe propriétaire d'un fichier....259
6.10.10 chmod() : Change le mode du fichier....259
6.10.11 chown() : Change le groupe propriétaire du fichier....260
6.10.12 clearstatcache() : Efface le cache de stat....260
6.10.13 copy() : Copie un fichier....261
6.10.14 delete() : Effacer....261
6.10.15 dirname() : Renvoie le nom du dossier....261
6.10.16 disk_free_space() : Renvoie l'espace disque disponible dans le répertoire....262
6.10.17 disk_total_space() : Retourne la taille d'un dossier....262
6.10.18 diskfreespace() : Alias de disk_free_space....262
6.10.19 fclose() : Ferme un fichier....262
6.10.20 feof() : Teste la fin du fichier....263
6.10.21 fflush() : Envoi tout le contenu généré dans un fichier....263
6.10.22 fgetc() : Lit un caractère dans un fichier....263
6.10.23 fgetcsv() : Renvoie la ligne courante et cherche les champs CSV....264
6.10.24 fgets() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier....264
6.10.25 fgetss() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier et élimine les balises HTML....265
6.10.26 file_exists() : Vérifie si un fichier existe....265
6.10.27 file_get_contents() : Lit tout un fichier dans une chaîne....266
6.10.28 file_put_contents() : Ecrit une chaîne dans un fichier....266
6.10.29 file() : Lit le fichier et renvoie le résultat dans un tableau....267
6.10.30 fileatime() : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois....267
6.10.31 filectime() : Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois....268
6.10.32 filegroup() : Lire le nom du groupe....268
6.10.33 fileinode() : Renvoie le numéro d'inode du fichier....268
6.10.34 filemtime() : Renvoie la date de dernière modification du fichier....268
6.10.35 fileowner() : Renvoie le nom du propriétaire du fichier....269
6.10.36 fileperms() : Renvoie les permissions affectées au fichier....269
6.10.37 filesize() : Renvoie la taille du fichier....269
6.10.38 filetype() : Retourne le type de fichier....270
6.10.39 flock() : Verrouille le fichier....270
6.10.40 fnmatch() : Repère un fichier à partir d'un masque....271
6.10.41 fopen() : Ouverture d'un fichier ou d'une URL....271
6.10.42 fpassthru() : Affiche le reste du fichier....272
6.10.43 fputs() : Ecrit dans un fichier....273
6.10.44 fread() : Lecture du fichier en mode binaire....273
6.10.45 fscanf() : Analyse un fichier en fonction d'un format....273
6.10.46 fseek() : Modifie le pointeur de fichier....274
6.10.47 fstat() : Lit les informations sur un fichier à partir d'un pointeur de fichier....275
6.10.48 ftell() : Renvoie la position du pointeur du fichier....275
6.10.49 ftruncate() : Tronque un fichier....275
6.10.50 fwrite() : Ecriture du fichier en mode binaire....276
6.10.51 glob() : Recheche des chemins qui vérifie un masque....276
6.10.52 is_dir() : Indique si le nom de fichier est un dossier....277
6.10.53 is_executable() : Indique si le fichier est exécutable....277
6.10.54 is_file() : Indique si le fichier est un véritable fichier....277

# Sommaire

**6 Index des fonctions**
6.10.55 is_link() : Indique si le fichier est un lien symbolique....278
6.10.56 is_readable() : Indique un fichier est autorisé en lecture....278
6.10.57 is_uploaded_file() : Indique si le fichier a été téléchargé par HTTP POST....278
6.10.58 is_writable() : Indique si un fichier est autorisé en écriture....279
6.10.59 is_writeable() : Indique si un fichier est autorisé en écriture....279
6.10.60 link() : Crée un lien....279
6.10.61 linkinfo() : Renvoie les informations à propos d'un lien....280
6.10.62 lstat() : Retourne les informations à propos d'un fichier ou d'un lien symbolique....280
6.10.63 mkdir() : Crée un dossier....280
6.10.64 move_uploaded_file() : Déplace un fichier téléchargé....281
6.10.65 parse_ini_file() : Traite un fichier de configuration....281
6.10.66 pathinfo() : Retourne des informations sur un chemin système....282
6.10.67 pclose() : Ferme un processus de pointeur de fichier....283
6.10.68 popen() : Crée un processus de pointeur de fichier....283
6.10.69 readfile() : Affiche un fichier....284
6.10.70 readlink() : Renvoie le nom du fichier vers lequel pointe un lien symbolique....284
6.10.71 realpath() : Retourne le chemin canonique absolu....284
6.10.72 rename() : Renomme un fichier....284
6.10.73 rewind() : Replace le pointeur de fichier au début....285
6.10.74 rmdir() : Efface un dossier....285
6.10.75 set_file_buffer() : Alias de stream_set_write_buffer....285
6.10.76 stat() : Renvoie les informations à propos d'un fichier....285
6.10.77 symlink() : Crée un lien symbolique....286
6.10.78 tempnam() : Crée un fichier avec un nom unique....286
6.10.79 tmpfile() : Crée un fichier temporaire....287
6.10.80 touch() : Modifie la date de modification et de dernièr accès d'un fichier....287
6.10.81 umask() : Change le "umask" courant....288
6.10.82 unlink() : Efface un fichier....288
6.11 FTP....289
6.11.1 Introduction....289
6.11.2 Pré−requis....289
6.11.3 Installation....289
6.11.4 Configuration à l'exécution....289
6.11.5 Types de ressources....289
6.11.6 Constantes prédefinies....289
6.11.7 Exemples....290
6.11.9 ftp_cdup() : Change de dossier et passe au dossier parent....291
6.11.10 ftp_chdir() : Change le dossier FTP courant....291
6.11.11 ftp_chmod() : Modifie les droits d'un fichier via FTP....292
6.11.12 ftp_close() : Ferme une connexion FTP....292
6.11.13 ftp_connect() : Ouvre une connexion FTP....292
6.11.14 ftp_delete() : Efface un fichier sur un serveur FTP....292
6.11.15 ftp_exec() : Exécute une commande sur le serveur FTP....293
6.11.16 ftp_fget() : Télécharge un fichier via FTP dans un flux local....293
6.11.17 ftp_fput() : Charge un fichier sur un serveur FTP....293
6.11.18 ftp_get_option() : Lit différenctes options pour la connexion FTP courante....293
6.11.19 ftp_get() : Télécharge un fichier depuis un serveur FTP....294
6.11.20 ftp_login() : Identification avec une connexion FTP....294
6.11.21 ftp_mdtm() : Retourne la date de dernière modification d'un fichier sur un

# Sommaire

**6 Index des fonctions**
serveur FTP....295
6.11.22 ftp_mkdir() : Crée un dossier sur un serveur FTP....295
6.11.23 ftp_nb_continue() : Reprend le téléchargement d'un fichier (non bloquant)....295
6.11.24 ftp_nb_fget() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)....295
6.11.25 ftp_nb_fput() : Ecrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant)....296
6.11.26 ftp_nb_get() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)....296
6.11.27 ftp_nb_put() : Envoie un fichier sur le serveur FTP (non−bloquant)....297
6.11.28 ftp_nlist() : Retourne la liste des fichiers dans un dossier....298
6.11.29 ftp_pasv() : Active ou désactive le mode passif....298
6.11.30 ftp_put() : Charge un fichier sur un serveur FTP....298
6.11.31 ftp_pwd() : Retourne le nom du dossier courant....299
6.11.32 ftp_quit() : Ferme une connexion FTP....299
6.11.33 ftp_raw() : Envoie une commande FTP brute....299
6.11.34 ftp_rawlist() : Fait une liste détaillée de fichiers dans un dossier....299
6.11.35 ftp_rename() : Renomme un fichier sur un serveur FTP....299
6.11.36 ftp_rmdir() : Efface un dossier FTP....300
6.11.37 ftp_set_option() : Modifie les options de la connexion FTP....300
6.11.38 ftp_site() : Envoie la commande SITE au serveur....300
6.11.39 ftp_size() : Retourne la taille d'un fichier....300
6.11.40 ftp_ssl_connect() : Ouvre une connexion FTP sécurisée avec SSL....301
6.11.41 ftp_systype() : Retourne un identifiant de type de serveur FTP....301
6.12 Fonctions....302
6.12.1 Introduction....302
6.12.2 Pré−requis....302
6.12.3 Installation....302
6.12.4 Configuration à l'exécution....302
6.12.5 Constantes prédefinies....302
6.12.7 call_user_func() : Appelle une fonction utilisateur....302
6.12.8 create_function() : Crée une fonction anonyme (style lambda)....303
6.12.9 func_get_arg() : Retourne un élément de la liste des arguments....304
6.12.10 func_get_args() : Retourne les arguments d'une fonction sous forme de tableau....305
6.12.11 func_num_args() : Retourne le nombre d'arguments passé à la fonction....305
6.12.12 function_exists() : Indique si une fonction est définie....306
6.12.13 get_defined_functions() : Liste toutes les fonctions définies....306
6.12.14 register_shutdown_function() : Enregistre une fonction pour exécution à l'extinction....307
6.12.15 register_tick_function() : Enregistre une fonction exécutée à chaque tick....307
6.12.16 unregister_tick_function() : Annule la fonction exécutée à chaque tick....307
6.13 HTTP....309
6.13.1 Introduction....309
6.13.2 Pré−requis....309
6.13.3 Installation....309
6.13.4 Configuration à l'exécution....309
6.13.5 Types de ressources....309
6.13.6 Constantes prédefinies....309
6.13.8 headers_list() : Retourne la liste des entêtes de réponse du script courant....309
6.13.9 headers_sent() : Indique si les en−têtes HTTP ont déjà été envoyés....310

# Sommaire

**6 Index des fonctions**
6.13.10 setcookie() : Envoie un cookie....311
6.14 Images....313
6.14.1 Introduction....313
6.14.2 Pré−requis....313
6.14.3 Installation....314
6.14.4 Configuration à l'exécution....314
6.14.5 Types de ressources....315
6.14.6 Constantes prédefinies....315
6.14.7 Exemples....316
6.14.9 exif_read_data() : Lit les en−têtes EXIF des images JPEG et TIFF....318
6.14.10 exif_thumbnail() : Récupère la miniature d'une image TIFF ou JPEG....320
6.14.11 gd_info() : Retourne des informations à propos de la librairie GD installée....321
6.14.12 getimagesize() : Retourne la taille d'une image....322
6.14.13 image_type_to_mime_type() : Lit le Mime−Type d'un type d'image....323
6.14.14 image2wbmp() : Crée une image WBMP....324
6.14.15 imagealphablending() : Modifie le mode de blending d'une image....324
6.14.16 imageantialias() : Utiliser ou non les fonctions d'antialias....324
6.14.17 imagearc() : Dessine une ellipse partielle....325
6.14.18 imagechar() : Dessine un caractère horizontalement....325
6.14.19 imagecharup() : Dessine un caractère verticalement....325
6.14.20 imagecolorallocate() : Alloue une couleur pour une image....325
6.14.21 imagecolorallocatealpha() : Alloue une couleur à une image....326
6.14.22 imagecolorat() : Retourne l'index de la couleur d'un pixel donné....326
6.14.23 imagecolorclosest() : Retourne l'index de la couleur la plus proche d'une couleur donnée....326
6.14.24 imagecolorclosestalpha() : Retourne la couleur la plus proche, en tenant compte du canal alpha....327
6.14.25 imagecolorclosesthwb() : Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir (hue, white and blackness)....327
6.14.26 imagecolordeallocate() : Supprime une couleur d'une image....327
6.14.27 imagecolorexact() : Retourne l'index de la couleur donnée....327
6.14.28 imagecolorexactalpha() : Retourne l'index d'une couleur avec son canal alpha....328
6.14.29 imagecolormatch() : Fait correspondre un peu plus les couleurs de la version palette d'une image aux couleurs de sa version truecolor....328
6.14.30 imagecolorresolve() : Retourne l'index de la couleur donnée, ou la plus proche possible....328
6.14.31 imagecolorresolvealpha() : Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha....328
6.14.32 imagecolorset() : Change la couleur dans une palette à l'index donné....329
6.14.33 imagecolorsforindex() : Retourne la couleur associée à un index....329
6.14.34 imagecolorstotal() : Calcule le nombre de couleurs d'une palette....329
6.14.35 imagecolortransparent() : Définit la couleur transparente....329
6.14.36 imagecopy() : Copie une partie d'une image....329
6.14.37 imagecopymerge() : Copie et fusionne une partie d'une image....330
6.14.38 imagecopymergegray() : Copie et fusionne une partie d'une image en niveaux de gris....330
6.14.39 imagecopyresampled() : Copie, redimensionne, rééchantillonne une image....330
6.14.40 imagecopyresized() : Copie et redimensionne une partie d'une image....331
6.14.41 imagecreate() : Crée une nouvelle image à palette....331

# Sommaire

**6 Index des fonctions**

6.14.42 imagecreatefromgd() : Crée une nouvelle image à partir d'un fichier GD ou d'une URL....332
6.14.43 imagecreatefromgd2() : Crée une nouvelle image à partir d'un fichier GD2 ou d'une URL....332
6.14.44 imagecreatefromgd2part() : Crée une nouvelle image à partir d'une partie de fichier GD2 ou d'une URL....332
6.14.45 imagecreatefromgif() : Crée une nouvelle image à partir d'un fichier ou d'une URL....332
6.14.46 imagecreatefromjpeg() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL....333
6.14.47 imagecreatefrompng() : Crée une nouvelle image PNG à partir d'un fichier ou d'une URL....334
6.14.48 imagecreatefromstring() : Crée une image à partir d'une chaîne....334
6.14.49 imagecreatefromwbmp() : Crée une image depuis un fichier WBMP....334
6.14.50 imagecreatefromxbm() : Crée une image à partir d'un fichier XBM....335
6.14.51 imagecreatefromxpm() : Crée une image à partir d'un fichier XPM....335
6.14.52 imagecreatetruecolor() : Crée une nouvelle image en vraies couleurs....335
6.14.53 imagedashedline() : Dessine une ligne pointillée....336
6.14.54 imagedestroy() : détruit une image....336
6.14.55 imageellipse() : Dessine une ellipse....336
6.14.56 imagefill() : Remplit....337
6.14.57 imagefilledarc() : Dessine une ellipse partielle et la remplit....337
6.14.58 imagefilledellipse() : Dessine une ellipse pleine....337
6.14.59 imagefilledpolygon() : Dessine un polygone rempli....338
6.14.60 imagefilledrectangle() : Dessine un rectangle rempli....338
6.14.61 imagefilltoborder() : Remplit avec une région avec une couleur spécifique....338
6.14.62 imagefontheight() : Retourne la hauteur de la police....338
6.14.63 imagefontwidth() : Retourne la largeur de la police....338
6.14.64 imageftbbox() : Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2....339
6.14.65 imagefttext() : Ecrit du texte dans une image avec la police courante et FreeType 2....339
6.14.66 imagegammacorrect() : Applique une correction gamma à l'image....339
6.14.67 imagegd() : Génère une image au format GD, vers le navigateur ou un fichier....339
6.14.68 imagegd2() : Génère une image au format GD2, vers le navigateur ou un fichier....339
6.14.69 imagegif() : Envoie une image GIF vers un navigateur ou un fichier....340
6.14.70 imageinterlace() : Active ou désactive l'entrelacement....340
6.14.71 imageistruecolor() : Détermine si une image est une image truecolor....341
6.14.72 imagejpeg() : Envoie une image JPEG vers un navigateur ou un fichier....341
6.14.73 imageline() : Dessine une ligne....341
6.14.74 imageloadfont() : Charge une nouvelle police....342
6.14.75 imagepalettecopy() : Copie la palette d'une image à l'autre....342
6.14.76 imagepng() : Envoie une image PNG vers un navigateur ou un fichier....342
6.14.77 imagepolygon() : Dessine un polygone....342
6.14.78 imagepsbbox() : Retourne le rectangle entourant un texte et dessiné avec une police PostScript Type1....343
6.14.79 imagepscopyfont() : Fait une copie d'une police chargée pour modification ultérieure....344
6.14.80 imagepsencodefont() : Change le codage vectoriel d'un caractère dans

# Sommaire

**6 Index des fonctions**
une police....344
6.14.81 imagepsextendfont() : Etend ou condense une police de caractères....344
6.14.82 imagepsfreefont() : Libère la mémoire occupée par une police PostScript Type 1....345
6.14.83 imagepsloadfont() : Charge une police PostScript Type 1 depuis un fichier....345
6.14.84 imagepsslantfont() : Incline une police de caractères PostScript....345
6.14.85 imagepstext() : Dessine un texte sur une image avec une police PostScript Type1....346
6.14.86 imagerectangle() : Dessine un rectangle....346
6.14.87 imagerotate() : Fait tourner une image d'un angle....347
6.14.88 imagesavealpha() : Définit l'option pour enregistrer les informations complètes du canal alpha lors de la sauvegarde d'images PNG....347
6.14.89 imagesetbrush() : Modifie la brosse pour le dessin des lignes....347
6.14.90 imagesetpixel() : Dessine un pixel....347
6.14.91 imagesetstyle() : Configure le style pour le dessin des lignes....348
6.14.92 imagesetthickness() : Modifie l'épaisseur d'un trait....348
6.14.93 imagesettile() : Modifie l'image utilisée pour le carrelage....349
6.14.94 imagestring() : Dessine une chaîne horizontale....349
6.14.95 imagestringup() : Dessine une chaîne verticale....349
6.14.96 imagesx() : Retourne la largeur d'une image....350
6.14.97 imagesy() : Retourne la hauteur de l'image....350
6.14.98 imagetruecolortopalette() : Convertit une image en vraies couleurs en image à palette....350
6.14.99 imagettfbbox() : Retourne le rectangle entourant un texte et dessiné avec une police TrueType....351
6.14.100 imagettftext() : Dessine un texte avec une police TrueType....351
6.14.101 imagetypes() : Retourne les types d'images supportés par la version courante de PHP....352
6.14.102 imagewbmp() : Affiche une image WBMP....352
6.14.103 iptcembed() : Intègre des données binaires IPTC dans une image JPEG....353
6.14.104 iptcparse() : Analyse un bloc binaire IPTC http://www.iptc.org/ et recherche les balises simples....353
6.14.105 jpeg2wbmp() : Convertit une image JPEG en image WBMP....353
6.14.106 png2wbmp() : Convertit une image PNG en image WBMP....353
6.14.107 read_exif_data() : Alias de exif_read_data....354
6.15 IMAP....355
6.15.1 Introduction....355
6.15.2 Pré−requis....355
6.15.3 Installation....355
6.15.4 Configuration à l'exécution....356
6.15.5 Types de ressources....356
6.15.6 Constantes prédefinies....356
6.15.7 Voir aussi....357
6.15.9 imap_alerts() : Retourne toutes les alertes....359
6.15.10 imap_append() : Ajoute une chaîne dans une boîte aux lettres....359
6.15.11 imap_base64() : Décode un texte encodé en BASE64....360
6.15.12 imap_binary() : Convertit une chaîne à 8 bits en une chaîne à base64....360
6.15.13 imap_body() : Lit le corps d'un message....360
6.15.14 imap_bodystruct() : Lit la structure d'une section du corps d'un mail....360
6.15.15 imap_check() : Vérifie le courrier de la boîte aux lettres courante....360
6.15.16 imap_clearflag_full() : Supprime un flag sur un message....361

# Sommaire

**6 Index des fonctions**
6.15.17 imap_close() : Termine un flux IMAP....361
6.15.18 imap_createmailbox() : Crée une nouvelle boîte aux lettres....361
6.15.19 imap_delete() : Marque le fichier pour l'effacement, dans la boîte aux lettres courante....362
6.15.20 imap_deletemailbox() : Efface une boîte aux lettres....362
6.15.21 imap_errors() : Retourne toutes les erreurs....363
6.15.22 imap_expunge() : Efface tous les messages marqués pour l'effacement....363
6.15.23 imap_fetch_overview() : Lit un sommaire des en−têtes de messages....363
6.15.24 imap_fetchbody() : Retourne une section extraite du corps d'un message....364
6.15.25 imap_fetchheader() : Retourne l'en−tête d'un message....364
6.15.26 imap_fetchstructure() : Lit la structure d'un message....365
6.15.27 imap_get_quota() : Lit les quotas des boîtes aux lettres....365
6.15.28 imap_get_quotaroot() : Lit les quotas de chaque utilisateur....366
6.15.29 imap_getacl() : Retourne le ACL pour la boîte aux lettres....367
6.15.30 imap_getmailboxes() : Liste les boîtes aux lettres, et retourne le détail pour chacune....367
6.15.31 imap_getsubscribed() : Liste toutes les boîtes aux lettres souscrites....368
6.15.32 imap_header() : Lit l'en−tête d'un message....368
6.15.33 imap_headerinfo() : Lit l'en−tête du message....368
6.15.34 imap_headers() : Retourne les en−têtes de tous les messages d'une boîte aux lettres....369
6.15.35 imap_last_error() : Retourne la dernière erreur (si elle existe) qui est survenu lors de la dernière requête....370
6.15.36 imap_list() : Lit la liste des boîtes aux lettres....370
6.15.37 imap_listmailbox() : Liste les boîtes aux lettres....370
6.15.38 imap_listscan() : Lit la liste des boîtes aux lettres, et y recherche une chaîne....370
6.15.39 imap_listsubscribed() : Liste les boîtes aux lettres souscrites....370
6.15.40 imap_lsub() : Liste toutes les boîtes aux lettres enregistrées....371
6.15.41 imap_mail_compose() : Crée un message MIME....371
6.15.42 imap_mail_copy() : Copie les messages spécifiés dans une boîte aux lettres....371
6.15.43 imap_mail_move() : Déplace des messages dans une boîte aux lettres....372
6.15.44 imap_mail() : Envoie un message mail....372
6.15.45 imap_mailboxmsginfo() : Lit les informations à propos de la boîte aux lettres courante....372
6.15.46 imap_mime_header_decode() : Décode les éléments MIME d'une en−tête....373
6.15.47 imap_msgno() : Retourne le numéro de séquence de message pour un UID donné....373
6.15.48 imap_num_msg() : Retourne le nombre de message dans la boîte aux lettres courante....373
6.15.49 imap_num_recent() : Retourne le nombre de messages récents dans la boîte aux lettres courante....373
6.15.50 imap_open() : Ouvre un flux IMAP vers une boîte aux lettres....374
6.15.51 imap_ping() : Vérifie que le flux IMAP est toujours actif....375
6.15.52 imap_qprint() : Convertit une chaîne à guillemets en une chaîne à 8 bits....375
6.15.53 imap_renamemailbox() : Renomme une boîte aux lettres....375
6.15.54 imap_reopen() : Ouvre un flux IMAP vers une nouvelle boîte aux lettres....376
6.15.55 imap_rfc822_parse_adrlist() : Analyse une adresse email....376
6.15.56 imap_rfc822_parse_headers() : Analyse un en−tête mail....376
6.15.57 imap_rfc822_write_address() : Retourne une adresse email formatée

# Sommaire

**6 Index des fonctions**
correctement....377
6.15.58 imap_scanmailbox() : Lit la liste des boîtes aux lettres, et y recherche une chaîne....377
6.15.59 imap_search() : Retourne un tableau de message après recherche....377
6.15.60 imap_set_quota() : Modifie le quota d'une boîte aux lettres....378
6.15.61 imap_setacl() : Modifie le ACL de la boîte aux lettres....378
6.15.62 imap_setflag_full() : Positionne un flag sur un message....378
6.15.63 imap_sort() : Trie des messages....379
6.15.64 imap_status() : Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante....379
6.15.65 imap_subscribe() : Souscrit à une boîte aux lettres....380
6.15.66 imap_thread() : Retourne l'arbre de REFERENCE organisé par thread....380
6.15.67 imap_timeout() : Configure ou retourne le timeout....380
6.15.68 imap_uid() : Retourne l'UID d'un message....380
6.15.69 imap_undelete() : Enlève la marque d'effacement d'un message....381
6.15.70 imap_unsubscribe() : Termine la souscription à une boîte aux lettres....381
6.15.71 imap_utf7_decode() : Décode une chaîne modifiée UTF−7....381
6.15.72 imap_utf7_encode() : Convertit des données 8bit en texte UTF−7....381
6.15.73 imap_utf8() : Convertit du texte au format MIME en UTF8....382
6.16 LDAP....383
6.16.1 Introduction....383
6.16.2 Pré−requis....384
6.16.3 Installation....384
6.16.4 Configuration à l'exécution....384
6.16.5 Types de ressources....384
6.16.6 Constantes prédefinies....384
6.16.7 Exemples....385
6.16.8 Utiliser les fonctions LDAP de PHP....385
6.16.10 ldap_add() : Ajoute une entrée dans un dossier LDAP....387
6.16.11 ldap_bind() : Autentification au serveur LDAP....387
6.16.12 ldap_close() : Ferme une connexion au serveur LDAP....388
6.16.13 ldap_compare() : Compare une entrée avec des valeurs d'attributs....388
6.16.14 ldap_connect() : Se connecte à un serveur LDAP....389
6.16.15 ldap_count_entries() : Compte le nombre d'entrées après une recherche....390
6.16.16 ldap_delete() : Efface une entrée dans un dossier....390
6.16.17 ldap_dn2ufn() : Convert it un DN en format UFN (User Friendly Naming)....390
6.16.18 ldap_err2str() : Convertit un numéro d'erreur LDAP en message d'erreur....390
6.16.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée....391
6.16.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP....391
6.16.21 ldap_explode_dn() : Sépare les différents composants d'un DN....392
6.16.22 ldap_first_attribute() : Return first attribute....392
6.16.23 ldap_first_entry() : Retourne la première entrée....392
6.16.24 ldap_first_reference() : Return first reference....392
6.16.25 ldap_free_result() : Free result memory....393
6.16.26 ldap_get_attributes() : Lit les attributs d'une entrée....393
6.16.27 ldap_get_dn() : Lit le DN d'une entrée....394
6.16.28 ldap_get_entries() : Lit toutes les entrées du résultat....394
6.16.29 ldap_get_option() : Lit/écrit la valeur courante d'une option....394
6.16.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée....395

# Sommaire

**6 Index des fonctions**
6.16.31 ldap_get_values() : Lit toutes les valeurs d'une entrée LDAP....395
6.16.32 ldap_list() : Recherche dans un niveau....396
6.16.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante....396
6.16.34 ldap_mod_del() : Efface un attribut à l'entrée courante....397
6.16.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante....397
6.16.36 ldap_modify() : Modifie une entrée LDAP....397
6.16.37 ldap_next_attribute() : Lit l'attribut suivant....397
6.16.38 ldap_next_entry() : Lit la prochaine entrée....397
6.16.39 ldap_next_reference() : Lit la référence suivante....398
6.16.40 ldap_parse_reference() : Extrait les informations d'une référence d'entrée....398
6.16.41 ldap_parse_result() : Extrait des informations d'un résultat....398
6.16.42 ldap_read() : Lit une entrée....398
6.16.43 ldap_rename() : Modifie le nom d'une entrée....399
6.16.44 ldap_search() : Recherche sur le serveur LDAP....399
6.16.45 ldap_set_option() : Modifie la valeur d'une option LDAP....400
6.16.46 ldap_set_rebind_proc() : Configure une fonction de callback pour refaire des liaisons, lors de recherche de référants....401
6.16.47 ldap_sort() : Trie les entrées d'un résultat LDAP....401
6.16.48 ldap_start_tls() : Démarre TLS....402
6.16.49 ldap_t61_to_8859() : Convertit les caractères t6 en caractères 8859....402
6.16.50 ldap_unbind() : Déconnecte d'un serveur LDAP....402
6.17 Mail....403
6.17.1 Introduction....403
6.17.2 Pré−requis....403
6.17.3 Installation....403
6.17.4 Configuration à l'exécution....403
6.17.5 Types de ressources....404
6.17.6 Constantes prédefinies....404
6.17.8 mail() : Envoie de mail....404
6.18 Mathématiques....407
6.18.1 Introduction....407
6.18.2 Pré−requis....407
6.18.3 Installation....407
6.18.4 Configuration à l'exécution....407
6.18.5 Types de ressources....407
6.18.6 Constantes prédefinies....407
6.18.8 acos() : arc cosinus....409
6.18.9 acosh() : Arc cosinus hyperbolique....409
6.18.10 asin() : arc sinus....409
6.18.11 asinh() : Arc sinus hyperbolique....409
6.18.12 atan() : arc tangent....409
6.18.13 atan2() : Arc tangent de deux variables....410
6.18.14 atanh() : Arc tangeant hyperbolique....410
6.18.15 base_convert() : Convertit un nombre entre des bases arbitraires....410
6.18.16 bindec() : Convertit de binaire en décimal....410
6.18.17 ceil() : Arrondit au nombre supérieur....411
6.18.18 cos() : cosinus....411
6.18.19 cosh() : Cosinus hyperbolic....411
6.18.20 decbin() : Convertit de décimal en binaire....411
6.18.21 dechex() : Convertit de décimal en hexadécimal....412
6.18.22 decoct() : Convertit de décimal en octal....412

# Sommaire

**6 Index des fonctions**
6.18.23 deg2rad() : Convertit un nombre de degrés en radians....412
6.18.24 exp() : Exponentielle....412
6.18.25 expm1() : Retourne le résultat de exp(number) − 1, calculé de manière précise, même si val est proche de 0....412
6.18.26 floor() : Arrondit à l'entier inférieur....413
6.18.27 fmod() : Retourne le reste de la division....413
6.18.28 getrandmax() : Plus grande valeur aléatoire possible....413
6.18.29 hexdec() : Convertit de hexadécimal en décimal....413
6.18.30 hypot() : Retourne le résultat de sqrt( num1*num1 + num2*num2)....414
6.18.31 is_finite() : Indique si un nombre est fini....414
6.18.32 is_infinite() :....414
6.18.33 is_nan() : Indique si une valeur est un nombre....414
6.18.34 lcg_value() : Générateur de congruence combinée linéaire....414
6.18.35 log() : Logarithme naturel (népérien)....415
6.18.36 log10() : logarithme en base 10....415
6.18.37 log1p() : Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro....415
6.18.38 max() : La plus grande valeur....415
6.18.39 min() : La plus petite valeur....416
6.18.40 mt_getrandmax() : La plus grand valeur aléatoire possible....417
6.18.41 mt_rand() : Génère une valeur aléatoire (meilleure méthode)....417
6.18.42 mt_srand() : Initialise une valeur aléatoire (meilleure méthode)....417
6.18.43 octdec() : Convertit d'octal en décimal....418
6.18.44 pi() : Retourne la valeur de pi....418
6.18.45 pow() : Expression exponentielle....418
6.18.46 rad2deg() : Convertit de radians en degrés....419
6.18.47 rand() : Génère une valeur aléatoire....419
6.18.48 round() : Arrondi....419
6.18.49 sin() : Sinus....420
6.18.50 sinh() : Sinyus hyperbolique....420
6.18.51 sqrt() : Racine carrée....420
6.18.52 srand() : Initialise le générateur de nombres aléatoires....420
6.18.53 tan() : Tangente....421
6.18.54 tanh() : Tangente hyperbolique....421
6.19 Chaînes de caractères multi−octets....422
6.19.1 Introduction....422
6.19.2 Installation....423
6.19.3 Configuration à l'exécution....423
6.19.4 Types de ressources....425
6.19.5 Constantes prédefinies....425
6.19.6 Entrées/Sorties HTTP....425
6.19.7 Jeux de caractères supportés....426
6.19.8 Exploitation des chaînes multi−octets en PHP....427
6.19.9 Cas des caractères japonais....427
6.19.10 Références....428
6.19.12 mb_convert_encoding() : Conversion d'encodage....429
6.19.13 mb_convert_kana() : Convertit entre les différents "kana"....430
6.19.14 mb_convert_variables() : Convertit l'encodage de variables....430
6.19.15 mb_decode_mimeheader() : Décode une en−tête MIME....431
6.19.16 mb_decode_numericentity() : Décode les entités HTML en caractères....431
6.19.17 mb_detect_encoding() : Détecte un encodage....431

# Sommaire

**6 Index des fonctions**
6.19.18 mb_detect_order() : Lit/modifie l'ordre de détection des encodages.....432
6.19.19 mb_encode_mimeheader() : Encode une chaîne pour une en−tête MIME.....432
6.19.20 mb_encode_numericentity() : Encode des entités HTML.....433
6.19.21 mb_ereg_match() : Expression régulière POSIX pour les chaînes multi−octets.....433
6.19.22 mb_ereg_replace() : Remplace des segments de chaînes, avec le support des expressions régulières mutli−octet.....434
6.19.23 mb_ereg_search_getpos() : Retourne l'offset du début du prochain segment repéré par une expression régulière.....434
6.19.24 mb_ereg_search_getregs() : Lit le dernier segment de chaîne multi−octets qui correspond au masque.....435
6.19.25 mb_ereg_search_init() : Configure les chaînes et les expressions régulières pour le support des caractères multi−octets.....435
6.19.26 mb_ereg_search_pos() : Retourne la position et la longueur du segement de chaîne qui vérifie le masque de l'expression régulière.....436
6.19.27 mb_ereg_search_regs() : Retourne le segment de chaîne trouvé par une expression régulière multi−octets.....436
6.19.28 mb_ereg_search_setpos() : Choisit le point de départ de la recherche par expression régulière.....437
6.19.29 mb_ereg_search() : Recherche par expression régulière multi−octets.....437
6.19.30 mb_ereg() : Recherche par expression régulière avec support des caractères multi−octets.....437
6.19.31 mb_eregi_replace() : Expression régulière avec support des caractères multi−octets, sans tenir compte de la casse.....438
6.19.32 mb_eregi() : Regular expression match ignoring case with multibyte support.....438
6.19.33 mb_get_info() : Lit la configuration interne de l'extension mbstring.....439
6.19.34 mb_http_input() : Détecte le type d'encodage d'un caractère HTTP.....439
6.19.35 mb_http_output() : Lit/modifie l'encodage d'affichage.....440
6.19.36 mb_internal_encoding() : Lit/modifie l'encodage interne.....440
6.19.37 mb_language() : Lit/modifie le langage courant.....440
6.19.38 mb_output_handler() : Fonction de traitement des affichages web.....441
6.19.39 mb_parse_str() : Analyse les données HTTP GET/POST/COOKIE et assigne les variables globales.....441
6.19.40 mb_preferred_mime_name() : Détecte l'encodage MIME.....442
6.19.41 mb_regex_encoding() : Retourne le jeu de caractères courant pour les expressions régulières.....442
6.19.42 mb_regex_set_options() : Lit et modifie les options des fonctions d'expression régulières à support de caractères multi−octets.....442
6.19.43 mb_send_mail() : Envoie un mail encodé ISO−2022−JP (mail japonais).....443
6.19.44 mb_split() : Scinde une chaîne en tableau avec une expression régulière multi−octets.....443
6.19.45 mb_strcut() : Coupe une partie de chaîne.....444
6.19.46 mb_strimwidth() : Tronque une chaîne.....444
6.19.47 mb_strlen() : Retourne la taille d'une chaîne.....444
6.19.48 mb_strpos() : Repère la première occurence d'un caractère dans une chaîne.....444
6.19.49 mb_strrpos() : Repère la dernière occurence d'un caractère dans une chaîne.....445
6.19.50 mb_strtolower() : Met tous les caractères en minuscules.....445
6.19.51 mb_strtoupper() : Met tous les caractères en majuscules.....446

# Sommaire

**6 Index des fonctions**
6.19.52 mb_strwidth() : Retourne la largeur d'une chaîne....446
6.19.53 mb_substitute_character() : Lit/modifie les caractères de substitution....447
6.19.54 mb_substr_count() : Compte le nombre d'occurrences d'une sous−chaîne....447
6.19.55 mb_substr() : Lit une sous−chaîne....447
6.20 Chiffrage mcrypt....449
6.20.1 Introduction....449
6.20.2 Pré−requis....449
6.20.3 Installation....449
6.20.4 Configuration à l'exécution....449
6.20.5 Types de ressources....449
6.20.6 Constantes prédefinies....449
6.20.7 Mcrypt ciphers....450
6.20.8 Exemples....451
6.20.10 mcrypt_cfb() : Chiffre/déchiffre des données en mode CFB....452
6.20.11 mcrypt_create_iv() : Crée un vecteur d'initialisation à partir d'une source aléatoire....453
6.20.12 mcrypt_decrypt() : Déchiffre un texte....453
6.20.13 mcrypt_ecb() : Chiffre/déchiffre des données en mode ECB....454
6.20.14 mcrypt_enc_get_algorithms_name() : Retourne le nom de l'algorithme....454
6.20.15 mcrypt_enc_get_block_size() : Retourne la taille de blocs d'un algorithme....454
6.20.16 mcrypt_enc_get_iv_size() : Retourne la taille du VI d'un algorithme....454
6.20.17 mcrypt_enc_get_key_size() : Retourne la taille maximale de la clé pour un mode....455
6.20.18 mcrypt_enc_get_modes_name() : Retourne le nom du mode....455
6.20.19 mcrypt_enc_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés acceptées par un algorithme....455
6.20.20 mcrypt_enc_is_block_algorithm_mode() : Teste le chiffrage par blocs d'un mode....455
6.20.21 mcrypt_enc_is_block_algorithm() : Teste le chiffrage par blocs d'un algorithme....456
6.20.22 mcrypt_enc_is_block_mode() : Teste si le mode retourne les données par blocs....456
6.20.23 mcrypt_enc_self_test() : Teste un module ouvert....456
6.20.24 mcrypt_encrypt() : Chiffre un texte....456
6.20.25 mcrypt_generic_deinit() : Prépare le module pour le déchargement....457
6.20.26 mcrypt_generic_end() : Termine un chiffrage....457
6.20.27 mcrypt_generic_init() : Initialise tous les buffers nécessaires....457
6.20.28 mcrypt_generic() : Chiffre....457
6.20.29 mcrypt_get_block_size() : Retourne la taille de blocs d'un chiffrement....458
6.20.30 mcrypt_get_cipher_name() : Lit le nom du chiffrement utilisé....458
6.20.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffrement/mode....459
6.20.32 mcrypt_get_key_size() : Retourne la taille de la clé d'un chiffrement....459
6.20.33 mcrypt_list_algorithms() : Liste tous les algorithmes de chiffrement supportés....459
6.20.34 mcrypt_list_modes() : Liste tous les modes de chiffrement supportés....460
6.20.35 mcrypt_module_close() : Décharge le module de chiffrement....460
6.20.36 mcrypt_module_get_algo_block_size() : Retourne la taille de blocs d'un algorithme....460
6.20.37 mcrypt_module_get_algo_key_size() : Retourne la taille maximale de clé....460
6.20.38 mcrypt_module_get_supported_key_sizes() : Retourne un tableau

# Sommaire

**6 Index des fonctions**
contenant les tailles de clés supportée par un algorithme de chiffrement....461
6.20.39 mcrypt_module_is_block_algorithm_mode() : Indique si un mode fonctionne par blocs....461
6.20.40 mcrypt_module_is_block_algorithm() : Indique si un algorithme fonctionne par blocs....461
6.20.41 mcrypt_module_is_block_mode() : Indique si un mode travaille par blocs....461
6.20.42 mcrypt_module_open() : Ouvre le module de l'algorithme et le mode à utiliser....462
6.20.43 mcrypt_module_self_test() : Teste un mode....463
6.20.44 mcrypt_ofb() : Chiffre/déchiffre des données en mode OFB....463
6.20.45 mdecrypt_generic() : Déchiffre....463
6.21 Ming pour Flash....465
6.21.1 Introduction....465
6.21.2 Pré−requis....465
6.21.3 Installation....465
6.21.4 Configuration à l'exécution....466
6.21.5 Types de ressources....466
6.21.6 Constantes prédefinies....466
6.21.7 Classes prédefinies....467
6.21.9 ming_setscale() : Configure l'échelle (??)....469
6.21.10 ming_useswfversion() : Utilise la version de SWF (??)....469
6.21.11 swfaction() : Crée une nouvelle action....469
6.21.12 swfbitmap−>getheight() : Retourne la hauteur d'une bitmap....475
6.21.13 swfbitmap−>getwidth() : Retourne la largeur d'une bitmap....475
6.21.14 swfbitmap() : Crée un objet bitmap....475
6.21.15 swfbutton_keypress() : Retourne l'option d'actino pour keyPress (char)....476
6.21.16 swfbutton−>addaction() : Ajoute une action au bouton....476
6.21.17 swfbutton−>addshape() : Ajoute une forme à un bouton....477
6.21.18 swfbutton−>setaction() : Assigne l'action du bouton....477
6.21.19 swfbutton−>setdown() : Alias de addShape(shape, SWFBUTTON_DOWN)....477
6.21.20 swfbutton−>sethit() : Alias de addShape(shape, SWFBUTTON_HIT)....478
6.21.21 swfbutton−>setover() : Alias de addShape(shape, SWFBUTTON_OVER)....478
6.21.22 swfbutton−>setup() : Alias de SWFbutton−>addShape(shape, SWFBUTTON_UP)....478
6.21.23 swfbutton() : Crée un nouveau bouton....478
6.21.24 swfdisplayitem−>addcolor() : Ajoute une couleur à une transoformation....480
6.21.25 swfdisplayitem−>move() : Déplace un objet en coordonnées relatives....480
6.21.26 swfdisplayitem−>moveto() : Déplace un objet en coordonnées globales....481
6.21.27 swfdisplayitem−>multcolor() : Multiplie la couleur de transformation....481
6.21.28 swfdisplayitem−>remove() : Supprime un objet d'une animation....482
6.21.29 swfdisplayitem−>rotate() : Fait tourner une forme relativement....482
6.21.30 swfdisplayitem−>rotateto() : Tourne un objet en angle absolu....482
6.21.31 swfdisplayitem−>scale() : Etire un objet relativement....483
6.21.32 swfdisplayitem−>scaleto() : Etire un objet en coordonnées globales....484
6.21.33 swfdisplayitem−>setdepth() : Modifie la place en profondeur (z−order)....484
6.21.34 swfdisplayitem−>setname() : Nomme un objet....484
6.21.35 swfdisplayitem−>setratio() : Modifie le ratio de l'objet....485
6.21.36 swfdisplayitem−>skewx() : Incline suivant les X relativement....485
6.21.37 swfdisplayitem−>skewxto() : Incline suivant les X....486
6.21.38 swfdisplayitem−>skewy() : Incline suivant les Y relativement....486

# Sommaire

**6 Index des fonctions**

6.21.39 swfdisplayitem−>skewyto() : Incline suivant les Y....486
6.21.40 swfdisplayitem() : Crée un nouvel objet d'affichage displayitem....487
6.21.41 swffill−>moveto() : Déplace l'origine de l'objet SWFFill....487
6.21.42 swffill−>rotateto() : Tourne la forme....487
6.21.43 swffill−>scaleto() : Modifie l'échelle de la forme....488
6.21.44 swffill−>skewxto() : Incline (abscisses)....488
6.21.45 swffill−>skewyto() : Incline (ordonnées)....488
6.21.46 swffill() : Crée un objet de remplissage....488
6.21.47 swffont−>getwidth() : Retourne la taille de la chaîne....489
6.21.48 swffont() : Charge une police....489
6.21.49 swfgradient−>addentry() : Ajoute une couleur à la liste du gradient....489
6.21.50 swfgradient() : Crée un objet gradient....490
6.21.51 swfmorph−>getshape1() : Sélectionne la forme de départ....490
6.21.52 swfmorph−>getshape2() : Sélectionne la forme de fin....491
6.21.53 swfmorph() : Crée un morphing....491
6.21.54 swfmovie−>add() : Ajoute un objet dans une animation....492
6.21.55 swfmovie−>nextframe() : Passe à l'image suivante....492
6.21.56 swfmovie−>output() : Envoie votre animation au navigateur....492
6.21.57 swfmovie−>remove() : Supprime un objet d'une animation....493
6.21.58 swfmovie−>save() : Sauve dans un fichier....493
6.21.59 swfmovie−>setbackground() : Modifie la couleur de fond....493
6.21.60 swfmovie−>setdimension() : Modifie les dimensions de l'animation....494
6.21.61 swfmovie−>setframes() : Modifie le nombre total d'images dans l'animation....494
6.21.62 swfmovie−>setrate() : Modifie la vitesse de l'animation....494
6.21.63 swfmovie−>streammp3() : Envoie un fichier MP3 en streaming....495
6.21.64 swfmovie() : Crée un objet 'animation'....495
6.21.65 swfshape−>addfill() : Ajoute un remplissage plein à la forme Flash....495
6.21.66 swfshape−>drawcurve() : Dessine une courbe relativement....497
6.21.67 swfshape−>drawcurveto() : Dessine une courbe....497
6.21.68 swfshape−>drawline() : Dessine une ligne relativement....497
6.21.69 swfshape−>drawlineto() : Dessine une ligne....498
6.21.70 swfshape−>movepen() : Déplace le stylo relativement....498
6.21.71 swfshape−>movepento() : Déplace le stylo....498
6.21.72 swfshape−>setleftfill() : Modifie la couleur de rastérisation de gauche....499
6.21.73 swfshape−>setline() : Modifie le style de ligne de la forme....499
6.21.74 swfshape−>setrightfill() : Modifie la couleur de rastériation de droite....500
6.21.75 swfshape() : Crée une nouvelle forme....500
6.21.76 swfsprite−>add() : Ajoute un objet à un sprite....501
6.21.77 swfsprite−>nextframe() : Va à la prochaine image du sprite....501
6.21.78 swfsprite−>remove() : Supprime un objet dans un sprite....501
6.21.79 swfsprite−>setframes() : Fixe le nombre maximum d'image dans le sprite....502
6.21.80 swfsprite() : Crée un sprite....502
6.21.81 swftext−>addstring() : Ajoute du texte....503
6.21.82 swftext−>getwidth() : Calcule la longueur d'une chaîne....503
6.21.83 swftext−>moveto() : Déplace le stylo de texte....503
6.21.84 swftext−>setcolor() : Modifie la couleur de la police....503
6.21.85 swftext−>setfont() : Sélectionne la police courante....504
6.21.86 swftext−>setheight() : Modifie la hauteur de la police courante....504
6.21.87 swftext−>setspacing() : Modifie l'espacement de police....504
6.21.88 swftext() : Crée un nouvel objet texte....504

# Sommaire

**6 Index des fonctions**
6.21.89 swftextfield−>addstring() : Ajoute au texte....505
6.21.90 swftextfield−>align() : Modifie l'alignement du texte....505
6.21.91 swftextfield−>setbounds() : Sélectionne la largeur et hauteur du champs....505
6.21.92 swftextfield−>setcolor() : Modifie la couleur du champs texte....506
6.21.93 swftextfield−>setfont() : Modifie la police du champs....506
6.21.94 swftextfield−>setheight() : Modifie la hauteur de la police du champs texte....506
6.21.95 swftextfield−>setindentation() : Modifie l'indentation de la première ligne....506
6.21.96 swftextfield−>setleftmargin() : Modifie la marge de gauche....507
6.21.97 swftextfield−>setlinespacing() : Modifie l'espacement de lignes....507
6.21.98 swftextfield−>setmargins() : Modifie les marges du champs texte Flash....507
6.21.99 swftextfield−>setname() : Nomme le champs texte....507
6.21.100 swftextfield−>setrightmargin() : Modifie la marge de droite....508
6.21.101 swftextfield() : Crée un nouveau champs texte....508
6.22 Fonctions diverses....509
6.22.1 Introduction....509
6.22.2 Pré−requis....509
6.22.3 Installation....509
6.22.4 Configuration à l'exécution....509
6.22.5 Types de ressources....509
6.22.6 Constantes prédefinies....510
6.22.8 connection_status() : Retourne les bits de status de la connexion HTTP....510
6.22.9 connection_timeout() : Indique si le script a expiré....510
6.22.10 constant() : Retourne la valeur d'une constante....511
6.22.11 define() : Définit une constante....511
6.22.12 defined() : Vérifie l'existence d'une constante....512
6.22.13 die() : Alias de la fonction exit....512
6.22.14 eval() : Exécute une chaîne comme un script PHP....512
6.22.15 exit() : Termine le script courant....513
6.22.16 get_browser() : Indique de quoi est capable le navigateur client....513
6.22.17 highlight_file() : Colorisation de la syntaxe d'un fichier....514
6.22.18 highlight_string() : Applique la syntaxe colorisée à du code PHP....515
6.22.19 ignore_user_abort() : Active l'interruption de script sur déconnexion du visiteur....515
6.22.20 pack() : Compacte des données dans une chaîne binaire....516
6.22.21 show_source() : Génère la syntaxe colorisée d'un fichier PHP....517
6.22.22 sleep() : Retarde l'exécution....517
6.22.23 uniqid() : Génère un identifiant unique....517
6.22.24 unpack() : Déconditionne des données depuis une chaîne binaire....517
6.22.25 usleep() : Retarde l'exécution en micro−secondes....518
6.23 mnoGoSearch....519
6.23.1 Introduction....519
6.23.2 Pré−requis....519
6.23.3 Installation....519
6.23.4 Configuration à l'exécution....519
6.23.5 Types de ressources....520
6.23.6 Constantes prédefinies....520
6.23.8 udm_alloc_agent() : Alloue une session mnoGoSearch....522
6.23.9 udm_api_version() : Lit la version des API mnoGoSearch....523
6.23.10 udm_cat_list() : Liste toutes les catégories soeurs d'une catégorie....523
6.23.11 udm_cat_path() : Lit le chemin de la catégorie courante....524
6.23.12 udm_check_charset() : Vérifie si un jeu de caractère est supporté par

# Sommaire

**6 Index des fonctions**
mnogosearch....524
6.23.13 udm_check_stored() : Vérifie la connexion MnoGoSearch avec document stocké....525
6.23.14 udm_clear_search_limits() : Annule toutes les limitates de recherche MnoGoSearch....525
6.23.15 udm_close_stored() : Ferme la connexion MnoGoSearch avec le document enregistré....525
6.23.16 udm_crc32() : Retourne la somme de contrôle CRC32 d'une chaîne....525
6.23.17 udm_errno() : Numéro d'erreur mnoGoSearch....525
6.23.18 udm_error() : Message d'erreur mnoGoSearch....525
6.23.19 udm_find() : Effectue une recherche....526
6.23.20 udm_free_agent() : Détruit une session mnoGoSearch....526
6.23.21 udm_free_ispell_data() : Libère la mémoire allouée pour ispell par MnoGoSearch....527
6.23.22 udm_free_res() : Libère un résultat mnoGoSearch....527
6.23.23 udm_get_doc_count() : Lit le nombre total de documents dans les bases MnoGoSearch....527
6.23.24 udm_get_res_field() : Lit un champ de résultat mnoGoSearch....527
6.23.25 udm_get_res_param() : Lit les paramètres de résultats mnoGoSearch....528
6.23.26 udm_load_ispell_data() : Charge les données ispell....528
6.23.27 udm_open_stored() : Ouvre une connexion MnoGoSearch avec un document stocké....530
6.23.28 udm_set_agent_param() : Modifie les paramètres de l'agent MnoGoSearch....530
6.24 MySQL....534
6.24.1 Introduction....534
6.24.2 Pré−requis....534
6.24.3 Installation....534
6.24.4 Configuration à l'exécution....534
6.24.5 Types de ressources....535
6.24.6 Constantes prédefinies....535
6.24.7 Exemples....536
6.24.9 mysql_change_user() : Change le nom de session de l'utilisateur actif....537
6.24.10 mysql_client_encoding() : Retourne le nom du jeu de caractères utilisé par le client MySQL....538
6.24.11 mysql_close() : Ferme la connexion MySQL....538
6.24.12 mysql_connect() : Ouvre une connexion à un serveur MySQL....538
6.24.13 mysql_create_db() : Crée une base de données MySQL....539
6.24.14 mysql_data_seek() : Déplace le pointeur interne de résultat....540
6.24.15 mysql_db_name() : Lit les noms des bases de données....540
6.24.16 mysql_db_query() : Envoie une requête MySQL à un serveur MySQL....541
6.24.17 mysql_drop_db() : Efface une base de données MySQL....541
6.24.18 mysql_errno() : Retourne le numéro d'erreur de la dernière commande MySQL....541
6.24.19 mysql_error() : Retourne le texte associé avec l'erreur générée lors de la dernière requête....542
6.24.20 mysql_escape_string() : Protège une chaîne pour la passer à mysql_query()....543
6.24.21 mysql_fetch_array() : Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux....543
6.24.22 mysql_fetch_assoc() : Lit une ligne de résultats dans un tableau associatif....544

# Sommaire

**6 Index des fonctions**
6.24.23 mysql_fetch_field() : Retourne les données enregistrées dans une colonne sous forme d'objet....545
6.24.24 mysql_fetch_lengths() : Retourne la taille de chaque colonne d'une ligne de résultat....546
6.24.25 mysql_fetch_object() : Retourne une ligne de résultat MySQL sous la forme d'un objet....546
6.24.26 mysql_fetch_row() : Retourne une ligne de résultat MySQL sous la forme d'un tableau....547
6.24.27 mysql_field_flags() : Retourne des détails sur une colonne MySQL....548
6.24.28 mysql_field_len() : Retourne la taille d'un champ de résultat MySQL....548
6.24.29 mysql_field_name() : Retourne le nom d'une colonne....548
6.24.30 mysql_field_seek() : Déplace le pointeur de résultat....549
6.24.31 mysql_field_table() : Retourne le nom de la table où se trouve une colonne....549
6.24.32 mysql_field_type() : Retourne le type de la colonne courante....549
6.24.33 mysql_free_result() : Efface le résultat de la mémoire....550
6.24.34 mysql_get_client_info() : Lit les informations sur le client MySQL....550
6.24.35 mysql_get_host_info() : Lit les informations sur l'hôte MySQL....550
6.24.36 mysql_get_proto_info() : Lit les informations sur le protocole MySQL....551
6.24.37 mysql_get_server_info() : Lit les informations sur le serveur MySQL....551
6.24.38 mysql_info() : Lit des informations à propos de la dernière requête....551
6.24.39 mysql_insert_id() : Retourne l'identifiant généré par la dernière requête INSERT....552
6.24.40 mysql_list_dbs() : Liste les bases de données disponibles sur le serveur MySQL....553
6.24.41 mysql_list_fields() : Liste les champs du résultat MySQL....553
6.24.42 mysql_list_processes() : Liste les processus MySQL....554
6.24.43 mysql_list_tables() : Liste les tables d'une base de données....554
6.24.44 mysql_num_fields() : Retourne le nombre de champs d'un résultat MySQL....555
6.24.45 mysql_num_rows() : Retourne le nombre de lignes d'un résultat....555
6.24.46 mysql_pconnect() : Ouvre une connexion persistante à un serveur MySQL....556
6.24.47 mysql_ping() : Pingue le serveur, et se reconnecte au besoin....556
6.24.48 mysql_query() : Envoie une requête SQL à un serveur MySQL....557
6.24.49 mysql_real_escape_string() : Protège les caractères spéciaux d'une commande SQL....558
6.24.50 mysql_result() : Retourne un champ d'un résultat....558
6.24.51 mysql_select_db() : Sélectionne une base de données MySQL....559
6.24.52 mysql_stat() : Retourne le statut courant du serveur....559
6.24.53 mysql_tablename() : Lit le nom de la table qui contient un champ....559
6.24.54 mysql_thread_id() : Retourne l'identifiant du thread courant....560
6.24.55 mysql_unbuffered_query() : Exécute une requête SQL sans mobiliser les résultats....560
6.25 Réseau....562
6.25.1 Introduction....562
6.25.2 Pré−requis....562
6.25.3 Installation....562
6.25.4 Configuration à l'exécution....562
6.25.5 Types de ressources....562
6.25.6 Constantes prédefinies....562
6.25.8 closelog() : Ferme la connexion à l'historique système....563

# Sommaire

**6 Index des fonctions**
6.25.9 debugger_off() : Désactive le debugger interne de PHP....564
6.25.10 debugger_on() : Active le debugger interne de PHP....564
6.25.11 define_syslog_variables() : Initialise toutes les constantes liées au syslog....564
6.25.12 dns_check_record() : Synonyme de checkdnsrr....564
6.25.13 dns_get_mx() : Synonyme de getmxrr....564
6.25.14 dns_get_record() : Lit les données DNS associées à un hôte....564
6.25.15 fsockopen() : Ouvre une socket de connexion Internet ou Unix....567
6.25.16 gethostbyaddr() : Retourne le nom d'hôte correspondant à une IP....568
6.25.17 gethostbyname() : Retourne l'adresse IP correspondant à un hôte....569
6.25.18 gethostbynamel() : Retourne la liste d'IP correspondante à un hôte....569
6.25.19 getmxrr() : Retourne les enregistrements MX d'un hôte....570
6.25.20 getprotobyname() : Retourne le numéro de protocole associé à un nom de protocole....570
6.25.21 getprotobynumber() : Retourne le nom de protocole associé à un numéro de protocole....570
6.25.22 getservbyname() : Retourne le numéro de port associé à un service Internet et un protocole....570
6.25.23 getservbyport() : Retourne le service Internet qui correspond au port et protocole....571
6.25.24 ip2long() : Convertit une chaîne contenant une adresse (IPv4) IP numérique en adresse littérale....571
6.25.25 long2ip() : Convertit une adresse IP (IPv4) en adresse IP numérique....571
6.25.26 openlog() : Ouvre la connexion à l'historique système....571
6.25.27 pfsockopen() : Ouvre une socket de connexion Internet ou Unix persistante....572
6.25.28 socket_get_status() : Retourne les informations sur une socket....572
6.25.29 socket_set_blocking() : Active/désactive le mode bloquant d'une socket....572
6.25.30 socket_set_timeout() : Alias de stream_set_timeout....573
6.25.31 syslog() : Génère un message dans l'historique système....573
6.26 ODBC unifié....575
6.26.1 Introduction....575
6.26.2 Pré−requis....575
6.26.3 Installation....575
6.26.4 Configuration à l'exécution....576
6.26.5 Types de ressources....577
6.26.6 Constantes prédefinies....577
6.26.8 odbc_binmode() : Modifie la gestion des colonnes de données binaires....579
6.26.9 odbc_close_all() : Ferme toutes les connexions ODBC....580
6.26.10 odbc_close() : Ferme une connexion ODBC....580
6.26.11 odbc_columnprivileges() : Liste les colonnes et leurs droits associés....580
6.26.12 odbc_columns() : Liste les colonnes d'une table....581
6.26.13 odbc_commit() : Valide une transaction ODBC....581
6.26.14 odbc_connect() : Connexion à une source....581
6.26.15 odbc_cursor() : Lecture du pointeur de fiche courante (cursorname)....582
6.26.16 odbc_data_source() : Retourne des informations sur la connexion courante....582
6.26.17 odbc_do() : Synonyme de odbc_exec....582
6.26.18 odbc_error() : Lit le dernier code d'erreur....582
6.26.19 odbc_errormsg() : Lit le dernier message d'erreur....583
6.26.20 odbc_exec() : Prépare et exécute une requête SQL....583
6.26.21 odbc_execute() : Exécute une requête SQL préparée....583

# Sommaire

**6 Index des fonctions**
6.26.22 odbc_fetch_array() : Lit une ligne de résultat dans un tableau associatif....584
6.26.23 odbc_fetch_into() : Lit une ligne de résultat, et la place dans un tableau....584
6.26.24 odbc_fetch_object() : Lit une ligne de résultat dans un objet....585
6.26.25 odbc_fetch_row() : Lit une ligne de résultat....585
6.26.26 odbc_field_len() : Lit la longueur d'un champ....585
6.26.27 odbc_field_name() : Lit le nom de la colonne....585
6.26.28 odbc_field_num() : Numéro de colonne....585
6.26.29 odbc_field_precision() : Alias de odbc_field_len....586
6.26.30 odbc_field_scale() : Lit l'échelle d'un champ....586
6.26.31 odbc_field_type() : Type de données d'un champ....586
6.26.32 odbc_foreignkeys() : Liste les clés étrangères....586
6.26.33 odbc_free_result() : Libère les ressources associées à un résultat....587
6.26.34 odbc_gettypeinfo() : Liste les types de données supportés par une source....587
6.26.35 odbc_longreadlen() : Gestion des colonnes de type LONG....588
6.26.36 odbc_next_result() : Vérifie si plusieurs résultats sont disponibles....588
6.26.37 odbc_num_fields() : Nombre de colonnes dans un résultat....588
6.26.38 odbc_num_rows() : Nombre de lignes dans un résultat....588
6.26.39 odbc_pconnect() : Ouvre une connexion persistante à une source de données....588
6.26.40 odbc_prepare() : Prépare une commande pour l'exécution....589
6.26.41 odbc_primarykeys() : Liste les colonnes utilisées dans une clé primaire....589
6.26.42 odbc_procedurecolumns() : Liste les paramètres des procédures....589
6.26.43 odbc_procedures() : Liste les procédure stockées....590
6.26.44 odbc_result_all() : Affiche le résultat sous la forme d'une table HTML....590
6.26.45 odbc_result() : Lit un champ de résultat UODBC....590
6.26.46 odbc_rollback() : Annule une transaction....591
6.26.47 odbc_setoption() : Modifie les paramètres ODBC....591
6.26.48 odbc_specialcolumns() : Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table....592
6.26.49 odbc_statistics() : Calcule des statistiques sur une table....592
6.26.50 odbc_tableprivileges() : Liste les tables et leurs privilèges....593
6.26.51 odbc_tables() : Liste les tables d'une source....593
6.27 Oracle 8....595
6.27.1 Introduction....595
6.27.2 Pré−requis....595
6.27.3 Installation....596
6.27.4 Configuration à l'exécution....596
6.27.5 Types de ressources....596
6.27.6 Constantes prédefinies....596
6.27.7 Exemples....597
6.27.9 ocicancel() : Termine la lecture de ressources Oracle....598
6.27.10 ocicloselob() : Ferme un descripteur de lob....598
6.27.11 ocicollappend() : Ajoute un objet à une collection Oracle....599
6.27.12 ocicollassign() : Assigne une collection depuis une autre collection Oracle....599
6.27.13 ocicollassignelem() : Assigne l'élément val à une collection Oracle à l'index ndx....599
6.27.14 ocicollgetelem() : Lit la valeur d'un élément d'une collection Oracle à l'index ndx....599
6.27.15 ocicollmax() : Retourne la plus grande valeur d'une collection Oracle....599
6.27.16 ocicollsize() : Retourne la taille d'une collection Oracle....599
6.27.17 ocicolltrim() : Supprime num éléments à la fin de la collection Oracle....600

# Sommaire

**6 Index des fonctions**
6.27.18 ocicolumnisnull() : Teste si la valeur d'une colonne Oracle est NULL.....600
6.27.19 ocicolumname() : Retourne le nom d'une colonne Oracle.....600
6.27.20 ocicolumnprecision() : Lit la précision d'une colonne Oracle.....600
6.27.21 ocicolumnscale() : Lit la taille d'une colonne Oracle.....601
6.27.22 ocicolumnsize() : Retourne la taille d'une colonne Oracle.....601
6.27.23 ocicolumntype() : Retourne le type de données d'une colonne Oracle.....601
6.27.24 ocicolumntyperaw() : Lit directement le type de colonne Oracle.....602
6.27.25 ocicommit() : Valide les transactions Oracle en cours.....602
6.27.26 ocidefinebyname() : Utilise une variable PHP pour la phase de définition, dans une commande SELECT Oracle.....603
6.27.27 ocierror() : Retourne la dernière erreur de stmt|conn|global sur Oracle.....603
6.27.28 ociexecute() : Exécute une commande SQL Oracle.....603
6.27.29 ocifetch() : Lit la prochaine ligne dans le résultat Oracle.....604
6.27.30 ocifetchinto() : Retourne la ligne suivante d'un résultat Oracle dans un tableau.....604
6.27.31 ocifetchstatement() : Retourne toutes les lignes d'un résultat Oracle.....604
6.27.32 ocifreecollection() : Supprime une collection Oracle.....605
6.27.33 ocifreecursor() : Libère toutes les ressources réservées par un résultat Oracle.....605
6.27.34 ocifreedesc() : Détruit un pointeur de LOB Oracle.....605
6.27.35 ocifreestatement() : Libère toutes les ressources occupées par une commande Oracle.....606
6.27.36 ociinternaldebug() : Active ou désactive l'affichage des données de déboggage Oracle.....606
6.27.37 ociloadlob() : Charge un LOB Oracle.....606
6.27.38 ocilogoff() : Déconnecte d'un serveur Oracle.....606
6.27.39 ocilogon() : Etablit une connexion à un serveur Oracle.....606
6.27.40 ocinewcollection() : Initialise une nouvelle collection.....608
6.27.41 ocinewcursor() : Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références.....608
6.27.42 ocinewdescriptor() : Initialise un nouveau pointeur vide de LOB/FILE.....609
6.27.43 ocinlogon() : Etablit une nouvelle correction avec un serveur Oracle.....610
6.27.44 ocinumcols() : Retourne le nombre de colonnes dans un résultat.....611
6.27.45 ociparse() : Analyse une requête SQL avec Oracle.....612
6.27.46 ociplogon() : Ouvre une connexion persistante à un serveur Oracle.....612
6.27.47 ociresult() : Retourne la valeur d'une colonne dans un résultat Oracle.....612
6.27.48 ocirollback() : Annule les transactions Oracle en cours.....612
6.27.49 ocirowcount() : Retourne le nombre de lignes affectées par Oracle.....612
6.27.50 ocisavelob() : Sauve un LOB Oracle.....613
6.27.51 ocisavelobfile() : Sauve un fichier LOB Oracle.....613
6.27.52 ociserverversion() : Retourne une chaîne contenant les informations de version du serveur Oracle.....613
6.27.53 ocisetprefetch() : Indique le nombre de lignes qui doivent être pré−lues par Oracle.....613
6.27.54 ocistatementtype() : Retourne le type de commande OCI.....614
6.27.55 ociwritelobtofile() : Ecrit un LOB Oracle dans un fichier.....614
6.27.56 ociwritetemporarylob() : Ecrit un blob temporaire.....614
6.28 Bufferisation de sortie.....615
6.28.1 Introduction.....615
6.28.2 Pré−requis.....615
6.28.3 Installation.....615

# Sommaire

**6 Index des fonctions**
6.28.4 Configuration à l'exécution....615
6.28.5 Types de ressources....616
6.28.6 Constantes prédefinies....616
6.28.7 Exemples....616
6.28.8 Voir aussi....616
6.28.10 ob_clean() : Efface le buffer de sortie....617
6.28.11 ob_end_clean() : Détruit les données du buffer de sortie, et éteint la bufferisation de sortie....617
6.28.12 ob_end_flush() : Envoie les données du buffer de sortie, et éteint la bufferisation de sortie....617
6.28.13 ob_flush() : Envoie le buffer de sortie....618
6.28.14 ob_get_clean() : Lit le contenu courant du buffer de sortie puis l'efface....618
6.28.15 ob_get_contents() : Retourne le contenu du buffer de sortie....619
6.28.16 ob_get_flush() : Vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon....619
6.28.17 ob_get_length() : Retourne la longueur du contenu du buffer de sortie....619
6.28.18 ob_get_level() : Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie....620
6.28.19 ob_get_status() : Lit le statut du buffer de sortie....620
6.28.20 ob_gzhandler() : Fonction de callback pour la compression automatique des buffers....620
6.28.21 ob_implicit_flush() : Active/désactive l'envoi implicite....621
6.28.22 ob_list_handlers() : Liste les gestionnaires d'affichage utilisés....621
6.28.23 ob_start() : Enclenche la bufferisation de sortie....621
6.28.24 output_add_rewrite_var() : Ajoute une règle de réécriture d'URL....622
6.28.25 output_reset_rewrite_vars() : Annule la réécriture d'URL....623
6.29 PDF....624
6.29.1 Introduction....624
6.29.2 Pré−requis....624
6.29.3 Installation....624
6.29.4 Configuration à l'exécution....625
6.29.5 Confusion entre les vieilles versions de PDFLib....625
6.29.6 Exemples....625
6.29.7 Voir aussi....627
6.29.9 pdf_add_bookmark() : Ajoute un signet dans la page courante....630
6.29.10 pdf_add_launchlink() : Ajoute une annotation de lancement dans la page PDF courante....630
6.29.11 pdf_add_locallink() : Ajoute une annotation de lien dans la page PDF courante....630
6.29.12 pdf_add_note() : Ajoute une annotation dans la page PDF courante....631
6.29.13 pdf_add_outline() : Obsolète : Ajoute un signet dans la page courante....631
6.29.14 pdf_add_pdflink() : Ajoute une annotation sur un lien vers un fichier dans la page PDF courante....631
6.29.15 pdf_add_thumbnail() : Ajoute une miniature sur la page PDF courante....631
6.29.16 pdf_add_weblink() : Ajoute un lien web sur la page PDF courante....631
6.29.17 pdf_arc() : Dessine un arc de cercle PDF dans le sens anti−horaire....632
6.29.18 pdf_arcn() : Dessine un arc de cercle dans le sens horaire....632
6.29.19 pdf_attach_file() : Ajoute un fichier attaché à la page PDF....632
6.29.20 pdf_begin_page() : Initialise une nouvelle page de document PDF....632
6.29.21 pdf_begin_pattern() : Initialise un nouveau pattern PDF....633
6.29.22 pdf_begin_template() : Initialise un nouveau template PDF....633

# Sommaire

**6 Index des fonctions**

6.29.23 pdf_circle() : Dessine un cercle dans un document PDF....633
6.29.24 pdf_clip() : Modifie le chemin de clipping PDF....633
6.29.25 pdf_close_image() : Ferme une image dans un document PDF....633
6.29.26 pdf_close_pdi_page() : Ferme la page PDF....634
6.29.27 pdf_close_pdi() : Ferme le fichier PDF d'entrée....634
6.29.28 pdf_close() : Ferme le fichier PDF....634
6.29.29 pdf_closepath_fill_stroke() : Termine le chemin, dessine les bords et remplit la forme....634
6.29.30 pdf_closepath_stroke() : Termine le chemin et dessine les bords....634
6.29.31 pdf_closepath() : Termine le chemin PDF courant....634
6.29.32 pdf_concat() : Concatène une matrice avec le CTM....635
6.29.33 pdf_continue_text() : Affiche du texte à la prochaine ligne PDF....635
6.29.34 pdf_curveto() : Dessine une courbe de Bezier....635
6.29.35 pdf_delete() : Efface un objet PDF....635
6.29.36 pdf_end_page() : Termine la page PDF courante....635
6.29.37 pdf_end_pattern() : Termine le pattern PDF....636
6.29.38 pdf_end_template() : Termine le template PDF....636
6.29.39 pdf_endpath() : Obsolète : Termine le chemin courant....636
6.29.40 pdf_fill_stroke() : Remplit et passe le pinceau sur le chemin PDF courant....636
6.29.41 pdf_fill() : Remplit le chemin PDF courant avec la couleur courante....636
6.29.42 pdf_findfont() : Prépare une police pour utilisation ultérieure avec pdf_setfont....636
6.29.43 pdf_get_buffer() : Lit le tampon contenant le fichier PDF généré....637
6.29.44 pdf_get_font() : Obsolète: charge une police....637
6.29.45 pdf_get_fontname() : Obsolète : lit le nom de la police....637
6.29.46 pdf_get_fontsize() : Obsolète : gère les polices....637
6.29.47 pdf_get_image_height() : Retourne la hauteur d'une image....637
6.29.48 pdf_get_image_width() : Retourne la largeur d'une image....637
6.29.49 pdf_get_majorversion() : Retourne le numéro de version majeur de PDFlib....638
6.29.50 pdf_get_minorversion() : Retourne le numéro de version mineure de PDFlib....638
6.29.51 pdf_get_parameter() : Lit certains paramètres....638
6.29.52 pdf_get_pdi_parameter() : Lit des paramètres textuels dans le document PDI....638
6.29.53 pdf_get_pdi_value() : Lit des paramètres numériques dans le document PDF d'entrée....638
6.29.54 pdf_get_value() : Lit des paramètres numériques....638
6.29.55 pdf_initgraphics() : Remet à zéro l'environnement graphique PDF....639
6.29.56 pdf_lineto() : Dessine une ligne PDF....639
6.29.57 pdf_makespotcolor() : Place un point de couleur PDF....639
6.29.58 pdf_moveto() : Place le point courant PDF....639
6.29.59 pdf_new() : Crée un nouvel objet PDF....639
6.29.60 pdf_open_ccitt() : Ouvre une image contenant des données brutes CCITT....640
6.29.61 pdf_open_file() : Ouvre un nouveau fichier PDF....640
6.29.62 pdf_open_gif() : Obsolète : Ouvre une image GIF....640
6.29.63 pdf_open_image_file() : Lit une image depuis un fichier....640
6.29.64 pdf_open_image() : Ouvre une image....641
6.29.65 pdf_open_jpeg() : Obsolète : Ouvre une image JPEG....641
6.29.66 pdf_open_memory_image() : Ouvre une image créée en mémoire par PHP....641

# Sommaire

**6 Index des fonctions**

6.29.67 pdf_open_pdi_page() : Prépare une page....641
6.29.68 pdf_open_pdi() : Ouvre un fichier PDF....641
6.29.69 pdf_open_png() : Obsolète : ouvre une image PNG....642
6.29.70 pdf_open_tiff() : Obsolète : Ouvre une image TIFF....642
6.29.71 pdf_open() : Obsolète : Ouvre un nouvel objet PDF....642
6.29.72 pdf_place_image() : Place une image dans la page....642
6.29.73 pdf_place_pdi_page() : Place une page dans le document....642
6.29.74 pdf_rect() : Dessine un rectangle....642
6.29.75 pdf_restore() : Rétablit l'ancien environnement graphique PDF....642
6.29.76 pdf_rotate() : Configure la rotation....642
6.29.77 pdf_save() : Sauve l'environnement graphique courant....643
6.29.78 pdf_scale() : Configure l'échelle du document....643
6.29.79 pdf_set_border_color() : Configure la couleur des bords autour des liens et annotations....643
6.29.80 pdf_set_border_dash() : Configure le style des lignes autour des liens at annotations....643
6.29.81 pdf_set_border_style() : Choisit le style de bord autour des liens et annotations....643
6.29.82 pdf_set_char_spacing() : Obsolète : Configure l'espacement de caractères character spacing....644
6.29.83 pdf_set_duration() : Obsolète : Configure la durée entre deux pages....644
6.29.84 pdf_set_font() : Obsolète : Sélectionne la taille et le nom d'une police....644
6.29.85 pdf_set_horiz_scaling() : Obsolète : Configure l'échelle horizontale du texte....644
6.29.86 pdf_set_info_author() : Obsolète : remplit le champ d'auteur du document....644
6.29.87 pdf_set_info_creator() : Obsolète : remplit le champ de créateur du document....644
6.29.88 pdf_set_info_keywords() : Obsolète : remplit le champ de mots clés du document....644
6.29.89 pdf_set_info_subject() : Obsolète : remplit le champ de sujet du document....645
6.29.90 pdf_set_info_title() : Obsolète : remplit le champ de titre du document....645
6.29.91 pdf_set_info() : Remplit un champ de l'en−tête de document PDF....645
6.29.92 pdf_set_leading() : Obsolète : Configure la distance entre deux lignes de texte....645
6.29.93 pdf_set_parameter() : Modifie certains paramètres....645
6.29.94 pdf_set_text_matrix() : Obsolète : configure la matrice de texte....645
6.29.95 pdf_set_text_pos() : Modifie la position du texte....645
6.29.96 pdf_set_text_rendering() : Obsolète : Détermine le rendu du texte....646
6.29.97 pdf_set_text_rise() : Obsolète : configure l'élévation de texte....646
6.29.98 pdf_set_value() : Modifie les paramètres numériques....646
6.29.99 pdf_set_word_spacing() : Obsolète : configure l'espace entre deux mots....646
6.29.100 pdf_setcolor() : Configure la couleur de dessin et de remplissage....646
6.29.101 pdf_setdash() : Configure le mode de pointillé....647
6.29.102 pdf_setflat() : Configure la position à plat (flatness)....647
6.29.103 pdf_setfont() : Configure la police courante....647
6.29.104 pdf_setgray_fill() : Configure la couleur de remplissage à un niveau de gris....647
6.29.105 pdf_setgray_stroke() : Configure la couleur de dessin à un niveau de gris....648
6.29.106 pdf_setgray() : Configure la couleur de dessin et de remplissage à un niveau de gris....648
6.29.107 pdf_setlinecap() : Configure le paramètre de linecap....648

# Sommaire

**6 Index des fonctions**
6.29.108 pdf_setlinejoin() : Configure le paramètre de linejoin....648
6.29.109 pdf_setlinewidth() : Configure la largeur de ligne....648
6.29.110 pdf_setmatrix() : Configure la matrice de transformation....649
6.29.111 pdf_setmiterlimit() : Configure la "miter limit"....649
6.29.112 pdf_setpolydash() : Configure des pointillés complexes....649
6.29.113 pdf_setrgbcolor_fill() : Choisit la couleur utilisée pour le remplissage....649
6.29.114 pdf_setrgbcolor_stroke() : Choisit la couleur utilisée pour le dessin....649
6.29.115 pdf_setrgbcolor() : Choisit la couleur rgb de remplissage et de dessin....650
6.29.116 pdf_show_boxed() : Affiche le texte dans un cadre....650
6.29.117 pdf_show_xy() : Affiche un text à une position donnée....650
6.29.118 pdf_show() : Affiche le texte à la position courante....650
6.29.119 pdf_skew() : Incline le système de coordonnées....650
6.29.120 pdf_stringwidth() : Retourne la largeur d'un texte avec la police courante....651
6.29.121 pdf_stroke() : Dessine la ligne le long du chemin....651
6.29.122 pdf_translate() : Translate l'origine du système de coordonnées....651
6.30 Options PHP et informations....652
6.30.1 Introduction....652
6.30.2 Pré−requis....652
6.30.3 Installation....652
6.30.4 Configuration à l'exécution....652
6.30.5 Types de ressources....653
6.30.6 Constantes prédefinies....653
6.30.8 assert() : Vérifie si une assertion est fausse....655
6.30.9 dl() : Charge une extension PHP à la volée....656
6.30.10 extension_loaded() : Détermine si une extension est chargée ou non....657
6.30.11 get_cfg_var() : Retourne la valeur d'une option de PHP....657
6.30.12 get_current_user() : Retourne le nom du possesseur du script courant....658
6.30.13 get_defined_constants() : Retourne la liste des constantes et leur valeur....658
6.30.14 get_extension_funcs() : Liste les fonctions d'une extension....658
6.30.15 get_include_path() : Lit la valeur de la directive de configuration include_path....659
6.30.16 get_included_files() : Retourne un tableau avec les noms des fichiers qui sont inclus dans un script....659
6.30.17 get_loaded_extensions() : Retourne la liste de tous les modules compilés et chargés....660
6.30.18 get_magic_quotes_gpc() : Retourne la configuration actuelle de l'option magic_quotes_gpc....660
6.30.19 get_magic_quotes_runtime() : Retourne la configuration actuelle de l'option magic_quotes_runtime....661
6.30.20 get_required_files() : Alias de get_included_files....661
6.30.21 getenv() : Retourne la valeur d'une variable d'environnement....661
6.30.22 getlastmod() : Retourne la date de dernière modification de la page....661
6.30.23 getmygid() : Retourne le GID du propriétaire du script....662
6.30.24 getmyinode() : Retourne l'inode du script....662
6.30.25 getmypid() : Retourne le numéro de processus courant....662
6.30.26 getmyuid() : Retourne l'UID du propriétaire du script actuel....662
6.30.27 getopt() : Lit des options passés dans la ligne de commande....662
6.30.28 getrusage() : Retourne le niveau d'utilisation des ressources....663
6.30.29 ini_alter() : Alias de ini_set....663
6.30.30 ini_get_all() : Lit toutes les valeurs de configuration....663
6.30.31 ini_get() : Lit la valeur d'une option de configuration....664

# Sommaire

**6 Index des fonctions**
6.30.32 ini_restore() : Restaure la valeur de l'option de configuration....665
6.30.33 ini_set() : Modifie la valeur d'une option de configuration....665
6.30.34 main() : Fausse documentation pour main....669
6.30.35 memory_get_usage() : Indique la quantité de mémoire utilisée par PHP....669
6.30.36 php_ini_scanned_files() : Retourne la liste des fichiers .ini analysés dans les dossiers de configuration supplémentaires....669
6.30.37 php_logo_guid() : Retourne l'identifiant du logo PHP....670
6.30.38 php_sapi_name() : Retourne le type d'interface utilisé entre le serveur web et PHP....670
6.30.39 php_uname() : Retourne les informations sur le système d'exploitation....671
6.30.40 phpcredits() : Imprime les crédits de PHP....671
6.30.41 phpinfo() : Affiche de nombreuses informations sur le PHP....672
6.30.42 phpversion() : Retourne le numéro de la version courante de PHP....673
6.30.43 putenv() : Fixe la valeur d'une variable d'environnement....673
6.30.44 restore_include_path() : Restaure la valeur de la directive de configuration include_path....674
6.30.45 set_include_path() : Modifie la valeur de la directive de configuration include_path....674
6.30.46 set_magic_quotes_runtime() : Active/désactive l'option magic_quotes_runtime....674
6.30.47 set_time_limit() : Fixe le temps maximum d'exécution d'un script....675
6.30.48 version_compare() : Compare deux versions de PHP....675
6.30.49 zend_logo_guid() : Retourne le logo de Zend....675
6.30.50 zend_version() : Lit la version courante du moteur Zend....675
6.31 PostgreSQL....677
6.31.1 Introduction....677
6.31.2 Pré−requis....677
6.31.3 Installation....677
6.31.4 Configuration à l'exécution....677
6.31.5 Trucs et astuces d'utilisation....678
6.31.6 Constantes prédefinies....679
6.31.7 Exemples....680
6.31.9 pg_cancel_query() : Annule une requête asynchrone....681
6.31.10 pg_client_encoding() : Lit l'encodage du client....682
6.31.11 pg_close() : Termine une connexion PostgreSQL....682
6.31.12 pg_connect() : Ouvre une connexion PostgreSQL....682
6.31.13 pg_connection_busy() : Vérifie si la connexion PostGreSQL est occupée....683
6.31.14 pg_connection_reset() : Relance la connexion au serveur PostGresSQL....683
6.31.15 pg_connection_status() : Lit le statut de la connexion PostGreSQL....684
6.31.16 pg_convert() : Convertit un tableau associatif en une commande PostGreSQL....684
6.31.17 pg_copy_from() : Insère des lignes dans une table à partir d'un tableau....684
6.31.18 pg_copy_to() : Copie une table dans un tableau....685
6.31.19 pg_dbname() : Nom de la base de données....685
6.31.20 pg_delete() : Efface des lignes PostGreSQL....685
6.31.21 pg_end_copy() : Synchronise avec le serveur PostgreSQL....685
6.31.22 pg_escape_bytea() : Protège le caractères d'une chaîne binaire en mode bytea....686
6.31.23 pg_escape_string() : Protège une chaîne de caractères....686
6.31.24 pg_fetch_all() : Lit toutes les lignes d'un résultat....686
6.31.25 pg_fetch_array() : Lit une ligne de résultat PGSQL dans un tableau....687

# Sommaire

**6 Index des fonctions**

6.31.26 pg_fetch_assoc() : Lit une ligne de résultat PGSQL sous forme de tableau numérique....688
6.31.27 pg_fetch_object() : Lit une ligne de résultat PGSQL dans un objet....688
6.31.28 pg_fetch_result() : Retourne les valeurs d'un résultat....689
6.31.29 pg_fetch_row() : Lit une ligne dans un tableau....690
6.31.30 pg_field_is_null() : Teste si un champ PostGreSQL est à NULL....690
6.31.31 pg_field_name() : Retourne le nom d'un champ PostGreSQL....691
6.31.32 pg_field_num() : Retourne le numéro d'une colonne....691
6.31.33 pg_field_prtlen() : Retourne la taille imprimée....691
6.31.34 pg_field_size() : Retourne la taille interne de stockage d'un champ donné....692
6.31.35 pg_field_type() : Retourne le type d'un champ PostGreSQL donné par index....692
6.31.36 pg_free_result() : Libère la mémoire....692
6.31.37 pg_get_notify() : Lit le message de NOTIFY....693
6.31.38 pg_get_pid() : Lit l'identifiant de processus du serveur....693
6.31.39 pg_get_result() : Lit un résultat asynchrone....693
6.31.40 pg_host() : Retourne le nom d'hôte....694
6.31.41 pg_insert() : Insère un tableau dans une table....694
6.31.42 pg_last_error() : Lit le dernier message d'erreur sur la connexion....694
6.31.43 pg_last_notice() : Retourne la dernière note du serveur PostgreSQL....695
6.31.44 pg_last_oid() : Retourne l'identifiant du dernier objet....695
6.31.45 pg_lo_close() : Ferme un objet de grande taille PostgreSQL....695
6.31.46 pg_lo_create() : Crée un objet de grande taille PostgreSQL....696
6.31.47 pg_lo_export() : Exporte un objet de grande taille vers un fichier....696
6.31.48 pg_lo_import() : Importe un objet de grande taille depuis un fichier....696
6.31.49 pg_lo_open() : Ouvre un objet de grande taille PostgreSQL....697
6.31.50 pg_lo_read_all() : Lit un objet de grande taille en totalité....697
6.31.51 pg_lo_read() : Lit un objet de grande taille....697
6.31.52 pg_lo_seek() : Modifie la position dans un objet de grande taille....698
6.31.53 pg_lo_tell() : Retourne la position courante dans un objet de grande taille....698
6.31.54 pg_lo_unlink() : Efface un objet de grande taille....698
6.31.55 pg_lo_write() : Ecrit un objet de grande taille PGSQL....698
6.31.56 pg_meta_data() : Lit les données méta de la table PostGreSQL....699
6.31.57 pg_num_fields() : Retourne le nombre de champ....700
6.31.58 pg_num_rows() : Retourne le nombre de lignes PostgreSQL....700
6.31.59 pg_options() : Retourne les options PostgreSQL....700
6.31.60 pg_pconnect() : Etablit une connexion PostgreSQL persistante....700
6.31.61 pg_ping() : Pingue la connexion à la base....701
6.31.62 pg_port() : Retourne le numéro de port....701
6.31.63 pg_put_line() : Envoie une chaîne au serveur PostgreSQL....701
6.31.64 pg_query() : Exécute une requête....702
6.31.65 pg_result_error() : Lit le message d'erreur associé à un résultat....702
6.31.66 pg_result_seek() : Modifie la ligne courant dans un résultat....702
6.31.67 pg_result_status() : Lit le statut du résultat....703
6.31.68 pg_select() : Effectue une selection....703
6.31.69 pg_send_query() : Exécute une requête PostGreSQL asynchrone....703
6.31.70 pg_set_client_encoding() : Choisit l'encodage du client....704
6.31.71 pg_trace() : Active le suivi d'une connexion PostgreSQL....704
6.31.72 pg_tty() : Retourne le nom de tty....705
6.31.73 pg_unescape_bytea() : Supprime le protection d'une chaîne de type bytea....705
6.31.74 pg_untrace() : Termine le suivi d'une connexion PostgreSQL....705

# Sommaire

**6 Index des fonctions**
6.31.75 pg_update() : Modifie les lignes d'une table....705
6.32 Exécution de programmes externes....707
6.32.1 Introduction....707
6.32.2 Pré−requis....707
6.32.3 Installation....707
6.32.4 Configuration à l'exécution....707
6.32.5 Types de ressources....707
6.32.6 Constantes prédefinies....707
6.32.7 Voir aussi....707
6.32.9 escapeshellcmd() : Protège les caractères spéciaux du Shell....708
6.32.10 exec() : Exécute un programme externe....708
6.32.11 passthru() : Exécute un programme externe et affiche le résultat brut....709
6.32.12 proc_close() : Ferme un processus ouvert par proc_open() et retourne le code de sortie du processus....709
6.32.13 proc_get_status() : Lit les informations concernant un processus....709
6.32.14 proc_nice() : Change la priorité d'exécution du processus courant....709
6.32.15 proc_open() : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties....710
6.32.16 proc_terminate() : Termine un processus ouvert par proc_open()....711
6.32.17 shell_exec() : Exécute une commande via le shell et retourne le résultat sous forme de chaîne....711
6.32.18 system() : Exécute un programme externe et affiche le résultat....711
6.33 Expressions régulières compatibles Perl....713
6.33.1 Introduction....713
6.33.2 Pré−requis....713
6.33.3 Installation....713
6.33.4 Configuration à l'exécution....713
6.33.5 Types de ressources....713
6.33.6 Constantes prédefinies....713
6.33.7 Exemples....714
6.33.9 syntaxe des masques() : Fonctionnement des expressions régulières....714
6.33.10 preg_grep() : Retourne un tableau avec les résultat de la recherche....714
6.33.11 preg_match_all() : Expression régulière globale....715
6.33.12 preg_match() : Expression régulière standard....716
6.33.13 preg_quote() : Echappement des caractères spéciaux des expressions régulières....717
6.33.14 preg_replace_callback() : Rechercher/remplacer avec une expression régulière et fonction de callback....718
6.33.15 preg_replace() : Rechercher et remplacer par expression régulière standard....719
6.33.16 preg_split() : Eclate une chaîne par expression régulière....721
6.34 Sessions....723
6.34.1 Introduction....723
6.34.2 Sessions et sécurité....723
6.34.3 Pré−requis....724
6.34.4 Installation....724
6.34.5 Configuration à l'exécution....724
6.34.6 Types de ressources....726
6.34.7 Constantes prédefinies....727
6.34.8 Exemples....727
6.34.9 Passer l'identifiant de session (session ID)....728

# Sommaire

**6 Index des fonctions**
6.34.10 Gestion personnalisée des sessions....729
6.34.12 session_cache_limiter() : Lit et/ou modifie le limiteur de cache de session....729
6.34.13 session_decode() : Décode les données de session....730
6.34.14 session_destroy() : Détruit une session....730
6.34.15 session_encode() : Encode les données de session....731
6.34.16 session_get_cookie_params() : Lit la configuration du cookie de session....731
6.34.17 session_id() : Let et/ou modifie l'identifiant courant de session....731
6.34.18 session_is_registered() : Vérifie si une variable est enregistrée dans la session....732
6.34.19 session_module_name() : Lit et/ou modifie le module de session courant....732
6.34.20 session_name() : Lit et/ou modifie le nom de la session....732
6.34.21 session_regenerate_id() : Remplace l'identifiant de session courant par un nouveau....733
6.34.22 session_register() : Enregistre une variable dans une session....733
6.34.23 session_save_path() : Lit et/ou modifie le chemin de sauvegarde des sessions....734
6.34.24 session_set_cookie_params() : Modifie les paramètres du cookie de session....735
6.34.25 session_set_save_handler() : Configure les fonctions de stockage de sessions....735
6.34.26 session_start() : Initialise une session....736
6.34.27 session_unregister() : Supprime une variable de la session....737
6.34.28 session_unset() : Détruit toutes les variables d'une session....738
6.34.29 session_write_close() : Ecrit les données de session et ferme la session....738
6.35 Socket....739
6.35.1 Introduction....739
6.35.2 Pré−requis....739
6.35.3 Installation....739
6.35.4 Configuration à l'exécution....739
6.35.5 Types de ressources....739
6.35.6 Constantes prédefinies....740
6.35.7 Socket Errors....740
6.35.8 Exemples....741
6.35.10 socket_bind() : Lie un nom à une socket....743
6.35.11 socket_clear_error() : Efface toutes les erreurs précédemment générées par une socket....743
6.35.12 socket_close() : Ferme une socket....744
6.35.13 socket_connect() : Crée une connexion sur une socket....744
6.35.14 socket_create_listen() : Ouvre une socket sur un port pour accepter les connexions....745
6.35.15 socket_create_pair() : Crée une paire de socket identiques et les stocke dans fds....745
6.35.16 socket_create() : Crée une socket (point de communication)....745
6.35.17 socket_get_option() : Lit les options de socket....746
6.35.18 socket_getpeername() : Interroge l'autre extrémité de la communication....746
6.35.19 socket_getsockname() : Interroge la socket locale....747
6.35.20 socket_iovec_add() : Ajoute un nouveau vecteur dans le tableau....747
6.35.21 socket_iovec_alloc() : Construit un 'struct iovec'....747
6.35.22 socket_iovec_delete() : Efface un vecteur....748
6.35.23 socket_iovec_fetch() : Lit les données du vecteur specifié par iovec_id[iovec_position]....748

# Sommaire

**6 Index des fonctions**
6.35.24 socket_iovec_free() : Détruit le vecteur iovec specifié par iovec_id....748
6.35.25 socket_iovec_set() : Modifie les données du vecteur iovec_id[iovec_position]....749
6.35.26 socket_last_error() : Lit la dernière erreur générée par une socket....749
6.35.27 socket_listen() : Attend une connexion (listen) sur une socket....749
6.35.28 socket_read() : Lit des données d'une socket....750
6.35.29 socket_readv() : Lit des données dans un fd, en utilisant le tableau scatter−gather définit par iovec_id....751
6.35.30 socket_recv() : Reçoit des données d'une socket connectée....751
6.35.31 socket_recvfrom() : Reçoit des données d'une socket, connectée ou pas....751
6.35.32 socket_recvmsg() : Reçoit des messages sur uen socket, qu'elle soit orientée connexion ou pas....751
6.35.33 socket_select() : Exécute l'appel système select() un tableau de sockets avec une durée d'expiration....752
6.35.34 socket_send() : Envoie des données à une socket connectée....753
6.35.35 socket_sendmsg() : Envoie un message à une socket, qu'elle soit orientée connexion ou pas....754
6.35.36 socket_sendto() : Envoie une message à une socket, qu'elle soit connectée ou pas....754
6.35.37 socket_set_block() : Met la socket en mode bloquant....754
6.35.38 socket_set_nonblock() : Sélectionne le mode non bloquant d'un descripteur de fichier....755
6.35.39 socket_set_option() : Modifie les options de socket....755
6.35.40 socket_shutdown() : Eteind une socket en lecture et/ou écriture....755
6.35.41 socket_strerror() : Retourne une chaîne décrivant un message d'erreur....756
6.35.42 socket_write() : Ecrit dans une socket....756
6.35.43 socket_writev() : Ecrit des données dans un descripteur de fichier, fd, en utilisant le tableau scatter−gather définie par iovec_id....757
6.36 Chaînes de caractères (Strings)....758
6.36.1 Introduction....758
6.36.2 Pré−requis....758
6.36.3 Installation....758
6.36.4 Constantes prédefinies....758
6.36.5 Voir aussi....759
6.36.7 addslashes() : Ajoute des anti−slashs dans une chaîne....760
6.36.8 bin2hex() : Convertit des données binaires en représentation hexadécimale....761
6.36.9 chop() : Alias de rtrim....761
6.36.10 chr() : Retourne un caractère spécifique....761
6.36.11 chunk_split() : Scinde une chaîne....761
6.36.12 convert_cyr_string() : Convertit une chaîne d'un jeu de caractères cyrillique à l'autre....762
6.36.13 count_chars() : Retourne des statistiques sur les caractères utilisés dans une chaîne....762
6.36.14 crc32() : Calcule la somme de contrôle CRC32....763
6.36.15 crypt() : Chiffrage indéchiffrable (hashing)....763
6.36.16 echo() : Affiche une chaîne de caractères....764
6.36.17 explode() : Coupe une chaîne en segments....765
6.36.18 fprintf() : Ecrit une chaîne formatée dans un flux....766
6.36.19 get_html_translation_table() : Retourne la table de traduction des entités utilisée par htmlspecialchars et htmlentities....767

# Sommaire

**6 Index des fonctions**
6.36.20 hebrev() : Convertit un texte logique hébreux en texte visuel....767
6.36.21 hebrevc() : Convertit un texte logique hébreux en texte visuel, avec retours à la ligne....767
6.36.22 html_entity_decode() : Convertit toutes les entités HTML en caractères normaux....768
6.36.23 htmlentities() : Convertit tous les caractères éligibles en entités HTML....769
6.36.24 htmlspecialchars() : Convertit les caractères spéciaux en entités HTML....769
6.36.25 implode() : Rassemble les éléments d'un tableau en une chaîne....770
6.36.26 join() : Rassemble les éléments d'un tableau en une chaîne....771
6.36.27 levenshtein() : Calcule la distance Levenshtein entre deux chaînes....771
6.36.28 localeconv() : Lit la configuration locale....772
6.36.29 ltrim() : Supprime les caractères invisibles de début de chaîne....773
6.36.30 md5_file() : Calcule le md5 d'un fichier....774
6.36.31 md5() : Calcule le md5 d'une chaîne....774
6.36.32 metaphone() : Calcule la clé métaphone....775
6.36.33 money_format() : Met un nombre au format monétaire....775
6.36.34 nl_langinfo() : Rassemble des informations sur la langue et la configuration locale....777
6.36.35 nl2br() : Insère un retour à la ligne HTML à chaque nouvelle ligne....777
6.36.36 number_format() : Formate un nombre pour l'affichage....778
6.36.37 ord() : Retourne le code ASCII d'un caractère....778
6.36.38 parse_str() : Analyse une requête HTTP....778
6.36.39 print() : Affiche une chaîne de caractères....779
6.36.40 printf() : Affiche une chaîne de caractères formattée....780
6.36.41 quoted_printable_decode() : Convertit une chaîne quoted−printable en chaîne 8 bits....780
6.36.42 quotemeta() : Echappe les méta−caractères....780
6.36.43 rtrim() : Supprime les espaces de fin de chaîne....780
6.36.44 setlocale() : Modifie les informations de localisation....781
6.36.45 sha1_file() : Calcule le sha1 d'un fichier....782
6.36.46 sha1() : Calcule le sha1 d'une chaîne de caractères....782
6.36.47 similar_text() : Calcule la similarité de deux chaînes....783
6.36.48 soundex() : Calcule la clé soundex....783
6.36.49 sprintf() : Retourne une chaîne formattée....784
6.36.50 sscanf() : Analyse une chaîne à l'aide d'un format....785
6.36.51 str_ireplace() : Version insensible à la casse de str_replace....786
6.36.52 str_pad() : Complète une chaîne jusqu'à une taille donnée....786
6.36.53 str_repeat() : Répéte une chaîne....787
6.36.54 str_replace() : Remplace toutes les occurrences dans une chaîne....787
6.36.55 str_rot13() : Effectue une transformation rot13....788
6.36.56 str_shuffle() : Mélange les caractères d'une chaîne de caractères....788
6.36.57 str_split() : Convertit une chaîne de caractères en tableau....788
6.36.58 str_word_count() : Compte le nombre de mots utilisés dans une chaîne....789
6.36.59 strcasecmp() : Comparaison de chaînes binaires....790
6.36.60 strchr() : Trouve la première occurence d'un caractère dans une chaîne....790
6.36.61 strcmp() : Comparaison binaire de chaînes....790
6.36.62 strcoll() : Comparaison de chaînes localisée....791
6.36.63 strcspn() : Trouve un segment de chaîne ne contenant pas certains caractères....791
6.36.64 strip_tags() : Supprime les balises HTML et PHP d'une chaîne....791
6.36.65 stripcslashes() : Supprime les anti−slash d'une chaîne C....791

# Sommaire

**6 Index des fonctions**
6.36.66 stripos() : Rercherche la position d'une occurence dans une chaîne, sans tenir compte de la casse....792
6.36.67 stripslashes() : Supprimer les anti−slash d'une chaîne....792
6.36.68 stristr() : Trouve la première occurence dans une chaîne (insensible à la casse)....793
6.36.69 strlen() : Calcule la taille d'une chaîne....793
6.36.70 strnatcasecmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel" (insensible à la casse)....793
6.36.71 strnatcmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel"....794
6.36.72 strncasecmp() : Compare en binaire des chaînes de caractères....794
6.36.73 strncmp() : Comparaison binaire des n premiers caractères....795
6.36.74 strpos() : Trouve la position d'un caractère dans une chaîne....795
6.36.75 strrchr() : Retourne la fin de la chaîne....796
6.36.76 strrev() : Inverse une chaîne....796
6.36.77 strripos() : Trouve la position de la dernière occurence d'une chaîne dans une autre de façon insensible à la casse....796
6.36.78 strrpos() : Trouve la position de la dernière occurence d'un caractère dans une chaîne....797
6.36.79 strspn() : Trouve le premier segment de chaîne....798
6.36.80 strstr() : Trouve la première occurence dans une chaîne....798
6.36.81 strtok() : Coupe une chaîne en segments....798
6.36.82 strtolower() : Renvoie une chaîne en minuscules....799
6.36.83 strtoupper() : Renvoie une chaîne en majuscules....799
6.36.84 strtr() : Remplace des chaînes dans une chaîne....800
6.36.85 substr_count() : Compte de le nombre d'occurrences de segments dans une chaîne....800
6.36.86 substr_replace() : Remplace un segment dans une chaîne....801
6.36.87 substr() : Retourne un segment de chaîne....801
6.36.88 trim() : Supprime les espaces en début et fin de chaîne....802
6.36.89 ucfirst() : Met le premier caractère en majuscule....803
6.36.90 ucwords() : Met en majuscule la première lettre de tous les mots....803
6.36.91 vprintf() : Affiche une chaîne formatée....803
6.36.92 vsprintf() : Retourne une chaîne formatée....804
6.36.93 wordwrap() : Effectue la césure d'une chaîne....804
6.37 URL....806
6.37.1 Introduction....806
6.37.2 Pré−requis....806
6.37.3 Installation....806
6.37.4 Configuration à l'exécution....806
6.37.5 Types de ressources....806
6.37.6 Constantes prédefinies....806
6.37.8 base64_encode() : Encode une chaîne en MIME base64....807
6.37.9 get_meta_tags() : Extrait tous les méta tags d'un fichier....807
6.37.10 http_build_query() : Génère une chaîne de requête en encodage URL....808
6.37.11 parse_url() : Analyse une URL et retourne ses composants....809
6.37.12 rawurldecode() : Décode une chaîne URL....810
6.37.13 rawurlencode() : Encode une chaîne en URL, selon la RFC1738....810
6.37.14 urldecode() : Décode une chaîne encodée URL....810
6.37.15 urlencode() : Encode une chaîne en URL....811
6.38 Variables....812
6.38.1 Introduction....812

# Sommaire

**6 Index des fonctions**
6.38.2 Pré−requis....812
6.38.3 Installation....812
6.38.4 Configuration à l'exécution....812
6.38.5 Types de ressources....812
6.38.6 Constantes prédefinies....812
6.38.8 empty() : Détermine si une variable contient une valeur non nulle....813
6.38.9 floatval() : Convertit une chaîne en nombre à virgule flottante....814
6.38.10 get_defined_vars() : Liste toutes les variables définies....814
6.38.11 get_resource_type() : Retourne le type de ressource....815
6.38.12 gettype() : Retourne le type de la variable....815
6.38.13 import_request_variables() : Importe les variables de GET/POST/Cookie dans l'environnement global....816
6.38.14 intval() : Retourne la valeur numérique entière équivalent d'une variable....816
6.38.15 is_array() : Détermine si une variable est un tableau....817
6.38.16 is_bool() : Détermine si une variable est un booléen....817
6.38.17 is_callable() : Détermine si l'argument peut être appelé comme fonction....817
6.38.18 is_double() : Alias de is_float....818
6.38.19 is_float() : Détermine si une variable est de type float....818
6.38.20 is_int() : Détermine si une variable est de type integer....818
6.38.21 is_integer() : Détermine si une variable est de type int....818
6.38.22 is_long() : Alias de is_int....819
6.38.23 is_null() : Indique si une variable vaut NULL....819
6.38.24 is_numeric() : Détermine si une variable est un type numérique....819
6.38.25 is_object() : Détermine si une variable est de type object....819
6.38.26 is_real() : Détermine si une variable est de type real....819
6.38.27 is_resource() : Détermine si une variable est une ressource....819
6.38.28 is_scalar() : Indique si une variable est un scalaire....820
6.38.29 is_string() : Détermine si une variable est de type chaîne....820
6.38.30 isset() : Détermine si une variable est affectée....820
6.38.31 print_r() : Affiche des informations lisibles pour une variable....821
6.38.32 serialize() : Linéarise une variable....822
6.38.33 settype() : Affecte un type à une variable....823
6.38.34 strval() : Retourne la valeur de la variable, au format chaîne....823
6.38.35 unserialize() : Crée une variable PHP à partir d'une valeur linéarisée....824
6.38.36 unset() : Détruit une variable....825
6.38.37 var_dump() : Affiche les informations d'une variable....826
6.38.38 var_export() : Retourne le code PHP utilisé pour générer une variable....827
6.39 WDDX....828
6.39.1 Introduction....828
6.39.2 Pré−requis....828
6.39.3 Installation....828
6.39.4 Configuration à l'exécution....828
6.39.5 Types de ressources....828
6.39.6 Constantes prédefinies....828
6.39.7 Exemples....828
6.39.9 wddx_deserialize() : Lire un paquet WDDX....829
6.39.10 wddx_packet_end() : Clos un paquet WDDX....829
6.39.11 wddx_packet_start() : Commencer un nouveau paquet WDDX avec une structure....829
6.39.12 wddx_serialize_value() : Enregistrer une valeur dans un paquet WDDX....830
6.39.13 wddx_serialize_vars() : Enregistrer plusieurs valeurs dans un paquet

# Sommaire

**6 Index des fonctions**
WDDX....830
6.40 Analyseur syntaxique XML....831
6.40.1 Introduction....831
6.40.2 Pré−requis....831
6.40.3 Installation....831
6.40.4 Configuration à l'exécution....831
6.40.5 Types de ressources....831
6.40.6 Constantes prédefinies....832
6.40.7 gestionnaires d'évènements....832
6.40.8 Problèmes de casse....833
6.40.9 Codes d'erreurs....833
6.40.10 Codage des caractères....833
6.40.11 Exemples....834
6.40.13 utf8_encode() : Convertit une chaîne ISO−8859−1 en UTF−8....838
6.40.14 xml_error_string() : Lit le message d'erreur de l'analyseur XML....838
6.40.15 xml_get_current_byte_index() : Retourne l'index de l'octet courant d'un analyseur XML....838
6.40.16 xml_get_current_column_number() : Retourne le nombre courant de colonne d'un analyseur XML....838
6.40.17 xml_get_current_line_number() : Retourne le numéro de ligne courant d'un analyseur XML....839
6.40.18 xml_get_error_code() : Retourne le nombre courant de colonne d'un analyseur XML....839
6.40.19 xml_parse_into_struct() : Analyse une structure XML....839
6.40.20 xml_parse() : Commence l'analyse d'un fichier XML....841
6.40.21 xml_parser_create_ns() : Crée un analyseur XML....842
6.40.22 xml_parser_create() : Création d'un analyseur XML....842
6.40.23 xml_parser_free() : Détruit un analyseur XML....842
6.40.24 xml_parser_get_option() : Lit les options d'un analyseur XML....843
6.40.25 xml_parser_set_option() : Affecte les options d'un analyseur XML....843
6.40.26 xml_set_character_data_handler() : Affecte les gestionnaires de texte littéral....843
6.40.27 xml_set_default_handler() : Affecte le gestionnaire XML par défaut....844
6.40.28 xml_set_element_handler() : Affecte les gestionnaires de début et de fin de balise XML....844
6.40.29 xml_set_end_namespace_decl_handler() : Configure le gestionnaire XML de données....845
6.40.30 xml_set_external_entity_ref_handler() : Configure le gestionnaire XML de références externes....846
6.40.31 xml_set_notation_decl_handler() : Configure le gestionnaire XML de notations....846
6.40.32 xml_set_object() : Utilise un analyseur XML à l'intérieur d'un objet....847
6.40.33 xml_set_processing_instruction_handler() : Affecte les gestionnaires d'instructions exécutables....848
6.40.34 xml_set_start_namespace_decl_handler() : Configure le gestionnaire de caractères....849
6.40.35 xml_set_unparsed_entity_decl_handler() : Affecte les gestionnaires d'entités non déclarés....849
6.41 Compression Zlib....851
6.41.1 Introduction....851
6.41.2 Pré−requis....851

# Sommaire

**6 Index des fonctions**
6.41.3 Installation....851
6.41.4 Configuration à l'exécution....851
6.41.5 Types de ressources....852
6.41.6 Constantes prédefinies....852
6.41.7 Exemples....852
6.41.9 gzcompress() : Compresse une chaîne....853
6.41.10 gzdeflate() : Compresse une chaîne avec la méthode DEFLATE....854
6.41.11 gzencode() : Crée une chaîne compressée....854
6.41.12 gzeof() : Indique si la fin d'un fichier compressé est atteinte....855
6.41.13 gzfile() : Lit la totalité d'un fichier compressé....855
6.41.14 gzgetc() : Lit un caractère dans un fichier compressé....855
6.41.15 gzgets() : Lit une ligne dans un fichier compressé....855
6.41.16 gzgetss() : Lit une ligne dans un fichier compressé, et supprime les balises....856
6.41.17 gzinflate() : Décompresse une chaîne....856
6.41.18 gzopen() : Ouvre un fichier compressé avec gzip....856
6.41.19 gzpassthru() : Affiche le contenu d'un fichier en une seule fois....857
6.41.20 gzputs() : Ecrit une chaîne dans un fichier compressé Gzip....857
6.41.21 gzread() : Lecture de fichier compressé binaire....857
6.41.22 gzrewind() : Recommence la lecture du fichier....857
6.41.23 gzseek() : Déplace le pointeur de lecture....858
6.41.24 gztell() : Lit la position courante du pointeur de lecture....858
6.41.25 gzuncompress() : Décompresse une chaîne compressée....858
6.41.26 gzwrite() : Ecrit dans un fichier compressé, en gzip....858
6.41.27 readgzfile() : Lit tout le fichier compressé....859
6.41.28 zlib_get_coding_type() : Retourne la méhode de compression utilisée avec Gzip....859

**7 Appendices....861**
7.1 Histoire de PHP....861
7.1.1 Histoire de PHP....861
7.1.2 Quelques projets liés à PHP....862
7.1.3 Livres traitant de PHP....864
7.1.4 Publications sur PHP....864
7.2 Migration de PHP 3.0 à PHP 4.0....864
7.2.1 Ce qui a changé en PHP 4.0....864
7.2.2 Utiliser PHP 3 et PHP 4 simultanément....865
7.2.3 Migration des fichiers de configuration....865
7.2.4 Comportement de l'analyseur....866
7.2.5 Rapport d'erreur....867
7.2.6 Initialiseur....868
7.2.7 empty("0")....868
7.2.8 Fonctions manquantes....868
7.2.9 Extensions PHP 3.0....869
7.2.10 Substitution de variables dans les chaînes....869
7.2.11 Cookies....869
7.2.12 Gestion des variables globales....869
7.3 Migration de PHP/FI 2.0 à PHP 3.0....870
7.3.1 A propos des incompatibilités en 3.0....870
7.3.2 old_function....870
7.3.3 Balises PHP....870

# Sommaire

**7 Appendices**
7.3.4 Syntaxe if..endif....871
7.3.5 Syntaxe while....871
7.3.6 Types d'expression....872
7.3.7 Les messages d'erreur ont changé....872
7.3.8 Evaluation rapide des booléens....872
7.3.9 La valeur TRUE / FALSE comme retour de fonctions....873
7.3.10 Diverses incompatibilités....873
7.4 Débuggeur PHP....874
7.4.1 A propos du débuggeur....874
7.4.2 Utiliser le débuggeur PHP....874
7.4.3 Protocole du débuggeur....874
7.5 Développement PHP....875
7.5.1 Créer une fonction PHP 3....875
7.5.2 Appeler des fonctions utilisateurs....883
7.5.3 Rapport d'erreurs....884
7.6 Utiliser PHP en ligne de commande....885
7.7 Liste d'alias....892
7.8 Mots réservés en PHP....896
7.8.1 Liste de mots−clé....897
7.8.2 Variables prédéfinies....897
7.8.3 Classes prédéfinies....902
7.8.4 Constantes prédéfinies en PHP....903
7.9 Types des ressources PHP....908
7.10 Liste des protocoles supportés....914
7.10.1 HTTP et HTTPS....914
7.10.2 FTP and FTPS....914
7.10.3 Flôts d'entrée/sortie....915
7.10.4 Flôts compressés....916
7.11 Liste des token de l'analyseur....916
7.12 A propos du manuel....918
7.12.1 Formats....918
7.12.2 A propos des notes utilisateurs....919
7.12.3 Comment lire la définition d'une fonction (prototype)....919
7.12.4 Versions de PHP documentées dans ce manuel....920
7.12.5 Où trouver plus d'informations sur PHP ?....920
7.12.6 Comment aider à l'amélioration de la documentation....921
7.12.7 Comment sont générées les documentations....921
7.12.8 Traductions....921
7.12.9 Traduction française....922

**8 Index....923**
8.1 Index des fonctions....923
8.2 Index des concepts....950
8.3 Index des exemples....977

# Manuel PHP

## 1.1 Les auteurs

- Stig Sæther Bakken
- Alexander Aulbach
- Egon Schmid
- Jim Winstead
- Lars Torben Wilson
- Rasmus Lerdorf
- Andrei Zmievski
- Jouni Ahto

Date de génération : 16/01/2004 09:01:02

## 1.2 L'éditeur

- Damien Seguy
- Mehdi Achour

©

## 1.3 Note légale

Copyright

Ce manuel est © Copyright 1997 − 2004 par PHP Documentation Group. Les membres de ce groupe sont listés sur la première page de ce manuel.

Ce manuel peut être redistribué sous licence GNU General Public License, comme stipulé par la Free Software Foundation; soit la version 2 de la Licence, soit (à votre choix), une version ultérieure.

Les membres du groupe de documentation sont listés au début de ce manuel. Si vous souhaitez des contacts avec les membres du groupe, écrivez à phpdoc@lists.php.net .

Les membres de la traduction française sont listés au début de ce manuel. Si vous souhaitez des contacts avec les membres du groupe, écrivez à doc−fr@lists.php.net .

La section 'Développer PHP 4.0' de ce manuel est copyrighté © 2000 par Zend Technologies, Ltd. Ce document ne peut pas être redistribué sans satisfaire les termes et conditions de la license Open Publication License, v1.0 ou plus récente (la dernière version est actuellement disponible à http://www.opencontent.org/openpub/ ).

# 2 Préface

**Introduction**

Ce manuel est essentiellement une référence des fonctions , mais il contient aussi des informations de référence sur le langage , des explications sur les fonctionnalités principales de PHP et diverses informations supplémentaires .

Vous pouvez télécharger ce manuel dans divers formats, sur le site officiel de http://www.php.net/docs.php , et sur le site de http://www.nexen.net/ . Les fichiers en téléchargement sont mis à jour au fur et à mesure des modifications. Plus d'informations sur ce manuel sont disponibles dans la section A propos du manuel .

Voir aussi PHP History .

# 3 Comment commencer

## 3.1 Configuration

### 3.1.1 Le fichier de configuration

Le fichier de configuration (appelé php3.ini dans la version 3.0 du PHP, et simplement php.ini dans la version 4.0) est lu par PHP au démarrage. Si vous avez compilé PHP en module, le fichier n'est lu qu'une seule fois, au lancement du démon HTTP . Pour la version CGI le fichier est lu à chaque invocation.

La localisation par défaut du php.ini est une option de compilation (voir FAQ ), mais elle peut être changée pour les versions CGI et CLI avec l'option de ligne de commande −c : voyez le chapitre sur l'utilisation de PHP en ligne de commande . Vous pouvez aussi utiliser la variable d'environnement PHPRC pour indiquer des chemins supplémentaires pour rechercher le fichier php.ini .

Toutes les directives PHP ne sont pas documentées ci−dessous. Pour une liste de toutes les directives, lisez le php.ini , qui est très bien documenté. Vous pouvez jeter un oeil à sa toute dernière version dans le CVS .

| Note |
| --- |
| La valeur par défaut de la directive PHP register_globals a été changée de **on** à **off** en PHP 4.2.0 . |

Extrait du php.ini

```
 ; tout texte sur une ligne, situé après un point−virgule  ";" est ignoré
[php] ; les marqueur de section (texte entre crochets) sont aussi ignorés
; Les valeurs booléenne peuvent être spécifiées comme ceci :
;    true, on, yes
; ou false, off, no, none
register_globals = off
magic_quotes_gpc = yes
; vous pouvez placer les chaînes de caractères entre guillemets
include_path = ".:/usr/local/lib/php"
; Les anti−slash sont traités comme n'importe quel caractère
include_path = ".;c:\php\lib"
</programlisting>
```

### 3.1.2 Comment modfier la configuration

#### 3.1.2.1 Exécuter PHP comme module Apache

Lorsque vous utilisez le module Apache, vous pouvez aussi changer les paramètres de configuration en utilisant les directives dans les fichiers de configuration d'Apache et dans les fichiers .htaccess .

Dans la version 3.0, à chaque directive de configuration présente dans le fichier de configuration d'Apache correspond une directive de configuration dans le fichier php3.ini , à l'exception des directives préfixées par "php3_".

Dans la version 4.0, il y a de nombreuses directives Apache qui vous permettent de modifier la configuration de PHP .

*php_value name value*

Modifie la valeur de la directive spécifiée. Cette instruction n'est utilisable qu'avec les directives PHP de type PHP_INI_ALL et PHP_INI_PERDIR. Pour annuler une valeur qui aurait été modifiée au préalable, utilisez la valeur none .

| Modification d'une valeur de type php_value |
|---|
| `php_value auto_prepend_file none` |

*php_flag name on|off*

Cette instruction est utilisée pour activer ou désactiver une option. Cette instruction n'est utilisable qu'avec les directives PHP de type PHP_INI_ALL et PHP_INI_PERDIR.

*php_admin_value name value*

Cette instruction affecte une valeur à la variable spécifiée. Cette instruction NE peux PAS être utilisée dans un fichier .htaccess . Toute directive de PHP configurée avec le type php_admin_value ne peut pas être modifiée en utilisant le fichier .htaccess .

| Modification d'une valeur de type php_admin_value |
|---|
| `php_admin_value open_basedir none` |

*php_admin_flag name on|off*

Cette directive est utilisée pour activer ou désactiver l'option précédente. Cette instruction NE peux PAS être utilisée dans un fichier .htaccess . Toute directive de PHP configurée avec le type php_admin_value ne peut pas être modifiée en utilisant le fichier .htaccess .

Exemple de configuration Apache

```
 <IfModule mod_php4.c>
  php_value include_path ".:/usr/local/lib/php"
  php_admin_flag safe_mode on
</IfModule>
<IfModule mod_php3.c>
  php3_include_path ".:/usr/local/lib/php"
  php3_safe_mode on
</IfModule>
```

Note

Les constantes PHP n'existent pas en dehors de PHP . Par exemple, dans le fichier httpd.conf , vous ne pouvez pas utiliser des constantes PHP telles que E_ALL ou E_NOTICE pour fixer le niveau de rapport d'erreur , car ces constantes n'ont pas de signification pour Apache, et seront remplacées par **0** . Utilisez les valeurs numériques à la place. Les valeurs numériques peuvent aussi être utilisées dans le php.ini

### 3.1.2.2 Autres interfaces de configuration de PHP

Indépendamment de l'interface de PHP , vous pouvez modifier certaines valeurs de la configuration durant l'exécution des scripts, grâce à la fonction ini_set . La table suivante fournit une présentation des différents endroits où la directive peut être modifiée.

| Constante | Valeur | Signification |
|---|---|---|
| PHP_INI_USER | 1 | La directive peut être configurée dans les scripts. |
| PHP_INI_PERDIR | 2 | La directive peut être configurée dans le fichier php.ini , le fichier .htaccess ou encore dans le fichier httpd.conf |
| PHP_INI_SYSTEM | 4 | La directive peut être configurée dans le fichier php.ini ou dans httpd.conf |
| PHP_INI_ALL | 7 | La directive peut être modifiée n'importe où. |

Vous pouvez voir l'état de votre configuration en utilisant la fonction phpinfo . Vous pouvez aussi accéder aux valeurs de votre configuration de manière individuelle en utilisant les fonctions ini_get or get_cfg_var .

## 3.1.3 Directives de configuration

### 3.1.3.1 Options Httpd

| Nom | Valeur par défaut | Modifications |
|---|---|---|
| async_send | "0" | PHP_INI_ALL |

### 3.1.3.2 Options du langage

| Nom | Valeur par défaut | Modifications |
|---|---|---|
| short_open_tag | On | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| asp_tags | Off | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| precision | "14" | PHP_INI_ALL |
| y2k_compliance | Off | PHP_INI_ALL |
| allow_call_time_pass_reference | On | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| expose_php | On | PHP_INI_SYSTEM |

Voici un éclaircissement sur l'utilisation des directives de configuration.

*short_open_tag bool*

Active le support de la forme courte des balises PHP : <? ?> . Si vous voulez utiliser PHP avec du XML, vous pouvez désactiver cette option afin que vous puissiez utiliser <?xml ?> . Sinon vous pouvez l'afficher avec PHP comme ceci : <?php echo '<?xml version="1.0"'; ?> . Si cette directive est désactivée, vous devez utiliser la forme longue des balises PHP <?php ?> .

| Note |
|---|
| Cette directive affecte aussi les raccourcis <?= , qui est l'équivalent de <? echo . Utiliser ce raccourci impose l'activation de la directive short_open_tag . |

*asp_tags bool*

Active l'utilisation des balises de type ASP <% %>, en plus des traditionnelles balises <?php ?> . Cela inclut l'utilisation du raccourcid <%= $value %>. Pour plus d'informations, reportez−vous à inclusion dans le HTML .

| Note |
|---|
| Le support des balises ASP a été ajouté dans la version 3.0.4. |

*precision integer*

Le nombre de chiffres significatifs à afficher avec les nombre à virgule flottante .

*y2k_compliance bool*

Active la compatibilité an 2000 (Cela posera des problèmes avec les navigateurs qui ne sont pas compatibles)

*allow_call_time_pass_reference bool*

Cette directive force le passage des arguments par référence au moment de l'appel des fonctions. Cette méthode est obsolète, et il est probable qu'elle soit abandonnée dans les futures versions de PHP et Zend. La méthode qui est recommandée est d'indiquer le passage par référence dans la déclaration de la fonction. Nous vous recommandons de tenter de désactiver cette directive et d'adapter vos scripts, afin d'en garantir leur portabilité (vous verrez une alerte s'afficher à chaque fois que vous utilisez cette fonctionnalité, et les arguments seront passés par valeur plutôt quee par référence).

Voir aussi Les références .

*expose_php bool*

Cette directive active ou non le fait que PHP est marqué comme installé sur le serveur web (en ajoutant sa propre signature dans les en−têtes du serveur web). Cet affichage n'est pas une menace en terme de sécurité, mais il permet de déterminer

l'utilisation de PHP sur votre serveur.

### 3.1.3.3 Limitation des ressources

| Nom | Valeur par défaut | Modifications |
|---|---|---|
| memory_limit | "8M" | PHP_INI_ALL |

Voici un éclaircissement sur l'utilisation des directives de configuration.

*memory_limit integer*
: Cette directive fixe la quantité maximale de mémoire, exprimée en octets, qu'un script peut réserver. Cela évite que des scripts mal écrits consomment toute la mémoire sur le serveur. Afin d'utiliser cette directive, vous devez l'avoir activée au moment de la compilation. De ce fait, vous devez trouver dans la ligne de configuration l'option −−enable−memory−limit . Notez que vous pouvez lui donner la valeur de −1 si vous ne souhaitez pas de limitation de consommation de mémoire.

Voir aussi max_execution_time .

### 3.1.3.4 Gestion des données

| Nom | Par défaut | Modifiable |
|---|---|---|
| track−vars | "On" | PHP_INI_?? |
| arg_separator.output | " | PHP_INI_ALL |
| arg_separator.input | " | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| variables_order | "EGPCS" | PHP_INI_ALL |
| register_globals | "Off" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| register_argc_argv | "On" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| post_max_size | "8M" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| gpc_order | "GPC" | PHP_INI_ALL |
| auto_prepend_file | "" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| auto_append_file | "" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| default_mimetype | "text/html" | PHP_INI_ALL |
| default_charset | "iso−8859−1" | PHP_INI_ALL |
| always_populate_raw_post_data | "0" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| allow_webdav_methods | "0" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |

Voici un éclaircissement sur l'utilisation des directives de configuration.

*track_vars bool*
: Si activé, les variables de l'environnement, GET, POST, Cookie et du serveur sont disponibles respectivement dans les tableaux associatifs $_ENV , $_GET , $_POST , $_COOKIE et $_SERVER .

  Notez que depuis PHP 4.0.3, track_vars est toujours activée.

*arg_separator.output string*
: Le séparateur utilisé par PHP dans les URL générées par PHP pour séparer les arguments.

*arg_separator.input string*
: Liste de séparateur(s) utilisé par PHP pour analyser les URL d'entrées, et en déduire les variables.

| Note |
|---|
| Chacun des caractères de cette directive est considéré comme un séparateur. |

*variables_order string*
: Etablit l'ordre de priorité des méthodes GET/POST/COOKIE et des valeurs issues de l'environnement et du serveur. Par défaut, cette directive est établie à "EGPCS". En affectant "GP" à cette directive, PHP ignorera les cookies, le serveur et

l'environnement et écrasera toute variable issue de la méthode GET par les valeurs des variables de même nom, issues de la méthode POST.

Voir aussi register_globals .

*register_globals bool*

Cette option active l'enregistrement des variables EGPCS (Environnement, GET, POST, Cookie, Serveur), en tant que variables globales. Par exemple, si register_globals = on, l'URL suivante http://www.example.com/test.php?id=3 va créer la variable $id dans votre script. Ou bien, $DOCUMENT_ROOT , issue de $_SERVER['DOCUMENT_ROOT'] . Vous pouvez désactiver cette fonction si vous ne voulez pas truffer vos scripts avec des valeurs utilisateurs. Depuis PHP 4.2.0 , cette directive vaut par défaut **off** . Il est recommandé de lire la liste des Variables prédéfinies , pour découvrir les super−globales : $_ENV , $_GET , $_POST , $_COOKIE et $_SERVER . Lisez aussi le chapitre sur la sécurité Utilisation des variables super−globales pour plus de détails.

Notez que register_globals ne peut être modifiée durant l'exécution du script (avec ini_set ). Cependant, vous pouvez utiliser le fichier .htaccess si votre hôte les supporte, comme décrit ci−dessus. Par exemple, vous pouvez inscrire dans votre fichier .htaccess : php_flag register_globals on .

| Note |
| --- |
| register_globals est combiné avec la directive variables_order . |

*register_argc_argv bool*

Indique à PHP s'il doit déclarer les variables argv et argc (qui contiendront alors les mêmes informations passées par méthode GET).

Voir aussi utiliser PHP en ligne de commande . De plus, cette directive a été introduite en PHP 4.0.0 et a toujours été à "on" jusque là.

*register_long_arrays boolean*

Indique à PHP s'il doit enregistrer les anciens tableaux $HTTP_*_VARS , comme variables prédéfinies . Lorsque cette option est activée (par défaut), les anciens tableaux tels que $HTTP_GET_VARS seront définis. Si vous ne les utilisez pas, il est recommandé de les désactiver, pour des raisons de performances. A la place, utilisez les super globales, telles que $_GET .

Cette directive a été introduite avec PHP 5.0.0.

*post_max_size integer*

Configure la taille maximale de données par méthode POST. Cette configuration affecte aussi le téléchargement de fichiers. Pour autoriser l'upload de grands fichiers, il faut que cette configuration soit plus grande encore, et plus grande que upload_max_filesize .

Si la limitation de mémoire est activée dans votre script de configuration memory_limit va aussi affecter le téléchargement de fichiers. En général, memory_limit doit être plus grand que post_max_size .

*gpc_order string*

Cette directive est obsolète. Voyez plutôt variables_order .

*auto_append_file string*

Spécifie le nom d'un fichier qui sera automatiquement ajouté après le fichier principal. Le fichier est inclus comme s'il avait été appelé avec la fonction include , donc include_path est utilisé.

Le mot réservé none désactive l'auto−appending.

| Note |
| --- |
| Si le script s'arrête par la fonction exit , auto−append **ne fonctionnera pas** . |

*auto_prepend_file string*

Spécifie le nom d'un fichier qui sera automatiquement ajouté avant le fichier principal. Le fichier est inclus comme s'il avait été appelé avec la fonction include , donc include_path est utilisé.

Le mot réservé none désactive l'auto−appending.

*default_mimetype string*

*default_charset string*

Depuis la version 4.0b4, PHP fournit toujours une indication du jeu de caractères pour les fichiers générés, dans l'en−tête HTTP Content−type . Pour désactiver l'envoi de cette indication, laissez cette directive vide.

*always_populate_raw_post_data bool*

Force le remplissage de la variable $HTTP_RAW_POST_DATA .

*allow_webdav_methods bool*

Cette directive autorise la gestion de requêtes HTTP WebDAV par PHP : PROPFIND, PROPPATCH, MOVE, COPY, etc... Si vous voulez les données postées par ces requêtes, vous devez aussi activer always_populate_raw_post_data .

Voir aussi magic_quotes_gpc , magic−quotes−runtime et magic_quotes_sybase .

### 3.1.3.5 Dossiers

| Nom | Par défaut | Modifiable |
| --- | --- | --- |
| include_path | PHP_INCLUDE_PATH | PHP_INI_ALL |
| doc_root | PHP_INCLUDE_PATH | PHP_INI_SYSTEM |
| user_dir | NULL | PHP_INI_SYSTEM |
| extension_dir | PHP_EXTENSION_DIR | PHP_INI_SYSTEM |
| cgi.force_redirect | "1" | PHP_INI_SYSTEM |
| cgi.redirect_status_env | "" | PHP_INI_SYSTEM |
| fastcgi.impersonate | "0" | PHP_INI_SYSTEM |

Voici un éclaircissement sur l'utilisation des directives de configuration.

*include_path string*

Spécifie une liste de dossiers, où les fonctions require , include et fopen_with_path() iront chercher les fichiers. Le format est le même que celui de la variable d'environnement **PATH** : une liste de dossiers, séparés par des deux points (:) sous UNIX, et par des points virgules (;) sous Windows.

| UNIX include_path |
| --- |
| include_path=".:/php/includes" |

| Windows include_path |
| --- |
| include_path=".;c:\php\includes" |

Utiliser un point ( . ) dans l'include_path permet de réaliser des inclusions relatives au dossier courant.

*doc_root string*

Le dossier racine de PHP . Cette directive n'est utilisée que si elle est définie. Si PHP est configuré en safe mode , aucun fichier en dehors de ce dossier ne sera accessible. Si PHP n'a pas été compilé avec l'option FORCE_REDIRECT, vous DEVRIEZ utiliser cette option, si vous exécutez PHP en tant que CGI avec serveur web autre que IIS. La solution alternative est d'utiliser la directive cgi.force_redirect

ci−dessous.

*user_dir string*

Répertoire où sont stockés les fichiers PHP dans le répertoire d'un utilisateur. Par exemple, public_html .

*extension_dir string*

Définit le répertoire dans lequel le PHP doit chercher les extensions lors du chargement dynamique. Voir aussi enable_dl et dl .

*extension string*

Définit les extensions qui doivent être chargées lors du démarrage du PHP.

*cgi.force_redirect bool*

cgi.force_redirect est nécesssaire pour assurer la sécurité de PHP , lorsqu'il fonctionne en tant que CGI pour la plupart des serveurs web. Même si laissé vide, PHP va activer cette option par défaut. Vous pouvez la désactiver **à vos risques et périls** .

| Note |
| --- |
| Utilisateurs Windows : vous pouvez désactiver cette option sans problème avec IIS et en fait, vous DEVEZ la désactiver. Pour faire fonctionner OmniHTTPD ou Xitami, vous devrez aussi la désactiver. |

*cgi.cgi.redirect_status_env string*

Si la directive cgi.force_redirect est activée, et que vous ne fonctionnez pas sous Apache ou Netscape (iPlanet), vous POURRIEZ devoir allouer une variable d'environnement que PHP ira rechercher pour s'assurer qu'il peut continuer son exécution.

| Note |
| --- |
| Allouer cette variable PEUT causer des problèmes de sécurité, RENSEIGNEZ VOUS AVANT DE LE FAIRE. |

*cgi.rfc2616_headers int*

Indique à PHP quel type d'en−tête utiliser lorsqu'il envoie les en−têtes de réponse HTTP. Si cette option vaut 0, PHP envoie un en−tête Status: qui est supporté par Apache et d'autres serveurs web. Lorsque cette option vaut 1, PHP va envoyer des en−têtes compatibles RFC 2616 . Laissez le à 0 si vous ne savez pas de quoi il s'agit.

*fastcgi.impersonate string*

FastCGI sous IIS (sur Windows NT) supporte la possibilité de traiter les tokens de sécurité du client. Cela permet à IIS de définir le contexte de sécurité à utiliser. mod_fastcgi sous Apache ne supporte pas encore cette fonctionnalité (17/03/2002). Utilisez la valeur de 1 pour l'activer sous IIS. La valeur par défaut est zéro.

### 3.1.3.6 Chargement de fichiers

| Nom | Par défaut | Modifiable |
| --- | --- | --- |
| file_uploads | "1" | PHP_INI_SYSTEM |
| upload_tmp_dir | NULL | PHP_INI_SYSTEM |
| upload_max_filesize | "2M" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |

Voici un éclaircissement sur l'utilisation des directives de configuration.

*file_uploads bool*

Cette directive indique si les chargements de fichiers sont supportés sur le serveur. Voir aussi upload_max_filesize , upload_tmp_dir et post_max_size .

*upload_tmp_dir string*

Indique le répertoire utilisé lors du chargement d'un fichier sur le serveur. Ce répertoire doit être accessible en écriture pour l'utilisateur qui exécute le script PHP , ou par le serveur web. Si il n'est pas spécifié, PHP utilisera le répertoire par défaut

du système.

*upload_max_filesize integer*
La taille maximale d'un fichier chargé sur le serveur. La valeur est en octets. Par défaut, elle est de 2 méga−octets.

#### 3.1.3.7 Options de SQL

| Nom | Par défaut | Modifiable |
|---|---|---|
| sql.safe_mode | "0" | PHP_INI_SYSTEM |

Voici un éclaircissement sur l'utilisation des directives de configuration.

*sql.safe_mode bool*

#### 3.1.3.8 Configuration des débogeurs

*debugger.host string*
Nom DNS ou adresse IP de l'hôte du débogueur.

*debugger.port string*
Numéro de port du débogueur.

*debugger.enabled bool*
Indique si le débogeur est activé ou pas.

# 3.2 Introduction

## 3.2.1 Qu'est ce que PHP?

PHP (officiellement, ce sigle est un acronyme récursif pour "PHP: Hypertext Preprocessor") est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML.

Réponse simple et claire, mais qu'est ce que cela veut dire? Un exemple :

Exemple d'introduction

```
 <html>
    <head>
        <title>Exemple</title>
    </head>
    <body>

        <?php
         echo "Bonjour, je suis un script PHP!";
         ?>

    </body>
</html>
```

Il est à noter la différence avec les autres scripts CGI écrits dans d'autres langages tels que le Perl ou le C : Au lieu d'écrire un programme avec de nombreuses lignes de commandes afin d'afficher une page HTML, vous écrivez une page HTML avec du code inclus à l'intérieur afin de réaliser une action précise (dans ce cas là, afficher du texte). Le code PHP est inclus entre une balise de début et une balise de fin qui permettent au serveur web de passer en "mode PHP".

Ce qui distingue le PHP des langages de script comme le Javascript est que le code est exécuté sur le serveur. Si vous avez un script similaire sur votre serveur, le client ne reçoit que le résultat du

script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques.

Le grand avantage de PHP est qu'il est extrêmement simple pour les néophytes, mais offre des fonctionnalités avancées pour les experts. Ne craignez pas de lire la longue liste de fonctionnalités PHP . Vous pouvez vous plonger dans le code, et en quelques instants, écrire des scripts simples.

Bien que le développement de PHP soit orienté vers la programmation pour les sites web, vous pouvez en faire bien d'autres usages. Lisez donc la section suivante : Que peut faire PHP? .

## 3.2.2 Que peut faire PHP?

Tout. PHP est principalement conçu pour servir de langage de script coté serveur, ce qui fait qu'il est capable de réaliser tout ce qu'un script CGI quelconque peut faire, comme collecter des données de formulaire, générer du contenu dynamique, ou gérer des cookies. Mais PHP peut en faire bien plus.

Il y a trois domaines différents où PHP peut s'illustrer.

- Langage de script coté serveur. C'est l'utilisation la plus traditionnelle, et aussi le principal objet de PHP . Vous aurez besoin de trois composants pour l'exploiter : un analyseur PHP (CGI ou module serveur), un serveur web et un navigateur web. Vous devez exécuter le serveur web en corrélation avec PHP . Vous pouvez accéder au programme PHP avec l'aide du navigateur web. Voyez la section d'installation pour plus d'informations.
- Langage de programmation en ligne de commande. Vous pouvez écrire des scripts PHP et l'exécuter en ligne de commande, sans l'aide du serveur web et d'un navigateur. Il vous suffit de disposer de l'exécutable PHP . Cette utilisation est idéale pour les scripts qui sont exécutés régulièrement (avec un cron sous Unix ou Linux), ou un Task Scheduler (sous Windows). Ces scripts peuvent aussi être utilisés pour réaliser des opérations sur des fichiers texte. Voyez la section sur l'utilisation de PHP en ligne de commande pour plus d'informations.
- Ecrire des applications graphiques. PHP n'est probablement pas le meilleur langage pour écrire des applications graphiques, mais si vous connaissez bien PHP et que vous souhaitez exploiter des fonctionnalités avancées dans vos applications clientes, vous pouvez utiliser PHP−GTK pour écrire de tels programmes. Vous avez aussi la possibilité d'écrire des applications très portables avec ce langage. PHP−GTK est une extension de PHP , qui n'est pas fournie dans la distribution de base. Si vous êtes interessé par PHP−GTK, visitez son site web .

PHP est utilisable sur la majorité des systèmes d'exploitation, comme Linux, de nombreuses variantes Unix (incluant HP−UX, Solaris et OpenBSD), Microsoft Windows, Mac OS X, RISC OS et d'autres encore. PHP supporte aussi la plupart des serveurs web actuels : Apache, Microsoft Internet Information Server, Personal Web Server, Netscape et iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd et beaucoup d'autres encore. Pour la majorité des serveurs web, PHP fonctionne comme module, et pour d'autres, il fonctionne comme exécutable CGI.

Avec PHP vous avez le choix de votre système d'exploitation et de votre serveur web. De plus, vous avez aussi le choix d'utiliser la programmation procédurale ou objet, ou encore un mélange des deux. Bien que le support de la couche objet ne soit pas très standard, de nombreuses librairies et applications d'envergure (y compris la bibliothèque PEAR) sont écrites en langage objet.

Avec PHP , vous n'êtes pas limités à la production de code HTML. Les capacités de PHP lui

permettent de générer aussi bien des images, des fichiers PDF, des animations Flash (avec l'aide des librairies libswf et Ming), générés à la volée. Vous pouvez aussi générer facilement du texte, du code XML ou XHTML. PHP génère tous ces fichiers, et les sauve dans le système de fichier, ou bien les envoie directement au navigateur web.

Une des grandes forces de PHP est le support de nombreuses bases de données. Ecrire une page web exploitant une base de données est extrêmement simple. Les bases de données suivantes sont toutes supportées par PHP :

- Adabas D
- dBase
- Empress
- FilePro (lecture seule)
- Hyperwave
- IBM DB2
- Informix
- Ingres
- InterBase
- FrontBase
- mSQL
- Direct MS−SQL
- MySQL
- ODBC
- Oracle (OCI7 et OCI8)
- Ovrimos
- PostgreSQL
- Solid
- Sybase
- Velocis
- Unix dbm

Il existe aussi des couches d'abstraction de base de données comme DBX qui vous permettent de vous connecter de manière transparente à toute base de données supportée par cette extension. De plus, PHP supporte ODBC, ce qui vous permet de vous connecter à toute autre base de données qui supporte ce standard.

PHP supporte de nombreux protocoles comme par exemple LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (sous Windows) et encore d'autres. Vous pouvez ouvrir des sockets réseau, et intéragir avec n'importe quel autre protocole. PHP supporte le format complexe WDDX, qui permet de communiquer entre tous les langages web. En terme d'interconnexion, PHP supporte aussi les instanciations d'objets Java, et les utilise de manière transparente comme objets intégrés. Vous pouvez aussi exploiter les objets distants avec CORBA.

PHP dispose de fonctionnalités extrêmement utiles pour le traitement de texte, allant des expressions régulières POSIX étendues ou Perl aux traitements des fichiers XML, avec les standards SAX et DOM. Vous pouvez utiliser les transformations XSLT.

En commerce électronique, vous trouverez des outils de paiement intégrés comme Cybercash, CyberMut, VeriSign Payflow Pro et CCVS, pour réaliser des paiements en ligne.

Enfin, PHP dispose d'extensions très pratiques comme le moteur de recherche mnoGoSearch, la passerelle avec IRC, des outils de compression (gzip, bz2) et de conversion calendaire, de traduction...

Comme vous le voyez, cette page n'est pas assez grande pour lister toutes les puissantes fonctionnalités de PHP . Lisez la section sur l'installation de PHP et étudiez la liste de fonctions pour avoir plus de détails sur toutes ces technologies.

# 3.3 Une introduction à PHP

Dans cette section, nous voulons illustrer les principes de base de PHP dans une courte introduction. Ce chapitre traite uniquement de créations de pages web dynamiques avec PHP, laissant de coté temporairement les autres possibilités de PHP. Voyez la section Ce que peut faire PHP pour plus d'informations.

Les pages web qui exploitent PHP sont traitées comme des pages HTML standards, et vous pouvez les créer, éditer et effacer tout comme vous le faites normalement avec des pages HTML classiques.

## 3.3.1 Le nécessaire

Dans ce tutoriel, nous présumons que vous avez un serveur web avec le support PHP activé, et que les fichiers terminés par l'extension .php sont traités par PHP. Sur la plupart des serveurs, c'est la configuration par défaut, mais n'hésitez pas à interroger votre administrateur système en cas de doute. Si votre serveur web supporte PHP, vous n'avez rien à faire. Simplement, créez un dossier, puis créez un fichier texte, avec l'extension .php : le serveur va automatiquement l'exécuter avec PHP. Il n'y a pas de compilation, ou d'installation compliquée. Gardez en tête que les fichiers sont comparables à des fichiers HTML, dans lequels vous allez utiliser des balises magiques, qui feront beaucoup de choses pour vous.

Supposons que vous souhaitiez économiser du temps en ligne et travailler localement. Dans ce cas, vous devez installer un serveur web comme Apache , et bien sur PHP . Vous souhaiterez aussi installer une base de données comme par exemple MySQL . Vous pouvez installer ces logiciels individuellement, ou bien avec des installeurs qui prennent en charge l'ensemble de l'installation en quelques clics. Il est facile de configurer un serveur web avec le support de PHP sur n'importe quel système d'exploitation, y compris Linux et Windows. Sous Linux, vous pouvez aussi trouver des commandes comme rpmfind , très pratique pour rechercher les paquets pré compilés.

## 3.3.2 Votre première page PHP

Créez un fichier appelé bonjour.php dans votre dossier web avec le contenu suivant :

Notre premier script PHP : bonjour.php

```
 <html>
 <head>
  <title>Test PHP</title>
 </head>
 <body>
 <?php echo "Bonjour le monde<p>"; ?>
 </body>
</html>
```

Utilisez votre navigateur pour accéder au fichier via votre serveur web, en ajoutant le nom de fichier "/bonjour.php". Si vous développez localement, votre URL ressemblera à http://localhost/bonjour.php ou encore http://127.0.0.1/bonjour.php mais cela dépend de la configuration de votre serveur web. Bien que cela soit hors du contexte de ce tutoriel, étudiez DocumentRoot et ServerName dans le fichier de configuration de votre serveur web (sous Apache, c'est httpd.conf ). Si ceci est configuré correctement, le fichier sera analysé par PHP et le résultat

suivant sera affiché :

```
<html>
<head>
<title>Test PHP</title>
</head>
<body>
Bonjour le monde<p>
</body>
</html>
```

Notez que cela n'est pas un script CGI. Ce fichier n'a nul besoin d'exécutable ou de configuration particulière. C'est vraiment comme un fichier HTML classique, qui comporte des balises spéciales, capables de réaliser des tâches particulières.

Ce programme est extrêmement simple et vous n'avez pas besoin de PHP pour créer une page web comme ceci. Elle ne fait qu'afficher Bonjour le monde , grâce à la fonction echo de PHP.

Si vous avez essayé cet exemple, et qu'il n'a rien affiché de spécial, ou même qu'un dialogue a surgi pour vous proposer de le télécharger, ou encore vous avez vu le code tel que nous l'avons écrit dans le fichier, alors votre serveur web ne supporte probablement pas le PHP. Demandez à votre administrateur de l'activer pour vous, en utilisant le chapitre Installation . Si vous voulez développer des scripts PHP localement, voyez la section de téléchargements . Vous pouvez développer en PHP, localement, sur tous les systèmes d'exploitation. N'oubliez pas d'installer aussi un serveur web approprié. Si le problème persiste, n'hésitez pas à utiliser l'une des options de support .

Le point important de cet exemple était de montrer le format des balises spéciales PHP. Nous avons utilisé ici <?php pour indiquer le début de la balise PHP. Puis, nous avons introduit les commandes PHP et refermé les balises PHP avec ?> . Vous pouvez passer du mode PHP au mode HTML et vice−versa, de cette manière, et à votre guise.

Note

Une note sur les éditeurs de texte

Il existe de nombreux éditeurs de texte et environnements de développement (IDE) que vous pouvez utiliser pour créer, éditer et gérer vos applications PHP. Une liste partielle de ces outils est entretenue à l'adresse PHP Editor's List . Si vous voulez recommander un éditeur, rendez donc une visite à cette page, et demandez au webmestre d'ajouter votre éditeur.

Note

Une note sur les traitements de texte

Les traitements de texte tels que StarOffice Writer, Microsoft Word et Abiword sont de très mauvais choix pour éditer des scripts PHP.

Si vous voulez utiliser l'un d'entre eux, malgré tout, pour tester vos scripts, vous devez vous assurer que vous sauvez les fichiers au format texte seul (PLAIN TEXT) : sinon, PHP ne sera pas capable de lire et d'exécuter ces scripts.

Note

Une note sur le Notepad de Windows

Si vous écrivez vos scripts PHP avec Windows Notepad, vous devez vous assurer que vos fichiers sont sauvés avec l'extension .php (Notepad ajoute automatiquement une extension .txt à vos

fichiers, à moins que vous ne preniez l'une des mesures suivantes :

Lorsque vous sauvez un fichier, et que vous êtes invité à lui donner un nom, placez le nom du fichier entre guillemets (i.e. "hello.php").

Ou bien, vous pouvez cliquer dans le menu 'Documents texte' du dialogue de sauvegarde, et choisir l'option "Tous les fichiers". Vous pourrez alors saisir le nom de votre fichier sans les guillemets.

## 3.3.3 Trucs pratiques

Réalisons maintenant quelque chose de plus puissant. Nous allons vérifier le type de navigateur que le visiteur de notre site utilise. Pour cela, nous allons accéder aux informations que le navigateur du visiteur nous envoie, lors de sa requête HTTP. Cette information est stockée dans une variable . Les variables sont faciles à repérer, car elles commencent toutes par un signe dollar. La variable qui nous interesse ici est is $_SERVER["HTTP_USER_AGENT"] .

Note

Attention pour les auto−globales de PHP

$_SERVER est une variable spéciale de PHP, qui contient toutes les informations relatives au serveur web. C'est une variable réservée de PHP, et une auto−globale. Reportez−vous aux pages du manuel traitant des Auto−globales (aussi connues sous le nom de super−globales). Ces variables spéciales ont été introduites en PHP 4.1.0 . Auparavant, il fallait utiliser les variables $HTTP_*_VARS , comme par exemple $HTTP_SERVER_VARS . Bien qu'obsolètes, ces variables existent toujours.

Pour afficher cette variable, nous pouvons simplement faire :

Afficher le contenu d'une variable (élément de tableau)

```
<?php echo $_SERVER["HTTP_USER_AGENT"]; ?>
```

Un résultat possible du script pourra alors être :
Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)

Il y a de nombreux types de variables disponibles en PHP. Dans l'exemple ci−dessus, nous avons affiché un élément de Tableau (Array) . Les tableaux peuvent être très utiles.

$_SERVER est juste une variable qui est automatiquement disponible dans votre script. Une liste de toutes les variables qui sont rendues disponibles est fournie dans la section Variables réservées . Vous pouvez aussi en obtenir une liste complète en créant un fichier comme ceci :

Afficher toutes les variables prédéfinies phpinfo

```
<?php phpinfo(); ?>
```

Lorsque vous chargez cette page dans votre navigateur, vous allez découvrir une page pleine d'informations sur PHP, ainsi qu'une liste des variables qui sont disponibles.

Vous pouvez ajouter plusieurs commandes PHP dans une balise PHP, et créer de petits blocs de code qui réalisent des opérations plus complexes qu'un simple affichage. Par exemple, si nous voulons vérifier que le navigateur est bien de la famille des Internet Explorer, nous pouvons faire cela :

Exemple utilisant les structures de contrôle et les fonctions

```
<?php
if (strstr($_SERVER["HTTP_USER_AGENT"], "MSIE")) {
    echo "Vous utilisez Internet Explorer<br />";
}
?>
```

Le résultat de ce script, si vous utilisez Internet Explorer, sera :
Vous utilisez Internet Explorer<br />

Ici, nous introduisons plusieurs nouveaux concepts. Nous avons une structure if . Si vous êtes familier avec les syntaxes de base du langage C, cela ne vous surprendra pas. Si vous ne connaissez pas assez le langage C ou un autre langage où la syntaxe est similaire à celle ci−dessus, il vaudrait mieux que vous lisiez une introduction à PHP, et assimiliez les premiers chapitres, ou bien lisez le chapitre consacré à la référence du langage . Vous pouvez trouver une liste de livres PHP à http://www.php.net/books.php .

Le second concept que nous avons introduit est la fonction strstr . strstr est une fonction intégrée à PHP, qui recherche la présence d'une chaîne dans une autre. Dans notre cas, nous avons recherché la chaîne "MSIE" dans la chaîne $_SERVER["HTTP_USER_AGENT"] . Si cette chaîne est trouvée, la fonction retourne TRUE et sinon, FALSE . Si elle retourne TRUE , la commande if reçoit TRUE et le code entre accolades {} est exécuté. N'hésitez pas à expérimenter d'autres exemples, à l'aide de if , else , et d'autres fonctions comme strtoupper et strlen . Chaque page de la documentation contient aussi des exemples.

Nous pouvons aussi progresser et vous montrer comment utiliser le mode PHP, au milieu du code HTML :

Passer du mode PHP au mode HTML et vice−versa

```
<?php
if (strstr($_SERVER["HTTP_USER_AGENT"], "MSIE")) {
?>
<h3>strstr a retourné TRUE</h3>
<center><b>Vous utilisez Internet Explorer</b></center>
<?php
} else {
?>
<h3>strstr a retourné FALSE</h3>
<center><b>Vous n'utilisez pas Internet Explorer</b></center>
<?php
}
?>
```

Un exemple de résultat obtenu dans ce script est :
<h3>strstr a retourné TRUE</h3>
Vous utilisez Internet Explorer

Au lieu d'utiliser une commande echo , pour afficher du texte, vous pouvez utiliser du code HTML pur. Le point important a noter ici et que la logique de programmation est conservée. Seul un des deux blocs HTML sera affiché, suivant que le navigateur utilise ou pas Internet Explorer.

## 3.3.4 Utiliser un formulaire

L'un des points forts de PHP est sa capacité à gérer les formulaires. Le concept de base qui est important à comprendre est que tous les champs d'un formulaire seront automatiquement disponibles dans le script PHP d'action. Lisez le chapitre du manuel concernant les variables d'origine externe à PHP pour plus d'informations et d'exemples sur comment utiliser les formulaires. Voici un exemple de formulaire HTML :

U*n simple formulaire HTML

```
<form action="action.php" method="POST">
Votre nom : <input type="text" name="nom" />
Votre age : <input type="text" name="age" />
```

```
 <input type="submit" value="OK">
</form>
```

Il n'y rien de particulier dans ce formulaire. Il est en HTML pur, sans aucune configuration particulière. Lorsque le visiteur remplit le formulaire, et clique sur le bouton d'envoi, le fichier action.php est appelé. Dans ce fichier, vous pouvez écrire le script suivant :

```
Afficher des données issues d'un formulaire
 Bonjour, <?php echo $_POST["nom"]; ?>.
Tu as <?php echo $_POST["age"]; ?> ans.
```

Voici le résultat que vous pourriez obtenir, selon les valeurs que vous avez saisies :
Bonjour Jean.
Tu as 29 ans.

Le fonctionnement du script devrait être évident. Il n'y a rien de plus simple. Les variables $_POST["nom"] et $_POST["age"] sont automatiquement créés par PHP. Un peu plus tôt dans ce tutorial, nous avons utilisé la variable $_SERVER , une auto−globale. Maintenant, nous avons introduit une autre auto−globale $_POST qui contient toutes les données envoyées par la méthode **POST** . Notez que dans notre formulaire, nous avons choisi la méthode **POST** . Si vous avions utilisé la méthode **GET** alors notre formulaire aurait placé ses informations dans la variable $_GET , une autre auto−globale. Vous pouvez aussi utiliser la variable $_REQUEST , si vous ne souhaitez pas vous embarrasser de la méthode utilisée. Elle contient un mélange des données de GET, POST, COOKIE et FILE. Voyez aussi la fonction import_request_variables .

## 3.3.5 Utiliser des codes anciens avec les nouvelles versions de PHP

Maintenant que PHP est devenu un langage de script populaire, il existe de nombreuses ressources qui vous proposent des portions de code que vous pouvez réutiliser dans vos codes. Pour la plupart, les dévelopeurs de PHP ont tâché d'assurer la compatibilité ascendante, ce qui fait que de nombreux scripts écrits pour les anciennes versions sont aussi valables pour les nouvelles versions de PHP, idéalement sans modifications. En pratique, certaines modifications doivent être apportées.

Les deux modifications récentes les plus importantes qui affectent les anciens codes sont :

- Les anciennes variables $HTTP_*_VARS (qui devaient être indiquées comme globales pour être utilisées dans une fonction ou une méthode) sont obsolètes. Les nouveaux tableaux auto−globaux ont été introduits en PHP 4.1.0 . Ce sont les variables suivantes : $_GET , $_POST , $_FILES , $_COOKIE , $_SERVER , $_ENV , $_REQUEST et $_SESSION . Les vieux tableaux $HTTP_*_VARS , tels que $HTTP_POST_VARS existent toujours depuis PHP 3. Depuis PHP 5.0.0, les tableaux prédéfinis PHP peuvent être désactivés avec l'option de configuration register_long_arrays .
- Les variables externes ne sont plus enregistrées dans l'environnement variable par défaut. En d'autres termes, depuis PHP 4.2.0 , la directive PHP register_globals vaut **off** par défaut dans le php.ini . La méthode recommandée pour accéder à ces valeurs, est via les tableaux auto−globaux mentionnés ci−dessus. Les anciens scripts, livres et tutoriaux continuent de considérer que cette directive devrait être à on . Lorsque cette directive est à on , vous pouvez utiliser la variable $id , si l'URL http://www.exemple.com/foo.php?id=42 a été appelée. Quelle que soit la valeur de la directive, $_GET['id'] est toujours disponible.

Pour plus de détails sur ces modifications, reportez−vous à variables prédéfinies .

### 3.3.6 Et après?

Avec ce que vous savez, vous êtes maintenant capable de comprendre l'essentiel de la documentation PHP, et les différents scripts d'exemple disponibles dans les archives. Vous pouvez aussi trouver d'autres exemples dans la section liens ("links", en anglais) du site http://www.php.net/links.php .

Différentes présentations des capacités de PHP sont disponibles sur le site des conférences PHP : http://conf.php.net/ ou http://talks.php.net/ .

## 3.4 Installation

### 3.4.1 Considérations générales sur l'installation

Avant d'installer PHP, vous devez savoir ce que vous voulez faire de PHP. Il y a trois cas d'utilisation que vous a décrit la section Que peut faire? section:

- Scripts web
- Scripts de ligne de commande
- Application à interface graphique

Pour la première tâche, qui est de loin la plus répandue, vous avez besoin de trois choses : PHP lui−même, un serveur web et un navigateur. Vous avez probablement un navigateur, et en fonction de votre système d'exploitation, vous pouvez aussi disposer d'un serveur Web (i.e. Apache sous Linux ou IIS sous Windows). Vous pouvez aussi louer un espace à une société. De cette façon, vous n'aurez pas à mettre en place PHP, mais uniquement à écrire vos scripts, les charger sur le serveur et voir le résultat sur votre navigateur.

Si vous installez PHP et le serveur par vous−même, vous avez deux choix. Soit sous la forme d'un module du serveur web (via une interface directe appelée SAPI). Les serveurs qui supportent cette solution comptent notamment Apache, Microsoft Internet Information Server, Netscape et iPlanet. D'autres serveurs ont aussi le support ISAPI, l'interface de module Microsoft (OmniHTTPd par exemple). Si PHP ne supporte pas l'interface de module de votre serveur web, vous pouvez toujours l'utiliser comme processeur CGI. Cela signifie que vous devez configurer votre serveur pour qu'il utilise l'exécutable PHP ( php.exe sous Windows), pour qu'il traite les fichiers PHP sur le serveur.

Si vous souhaitez aussi utiliser PHP en ligne de commande (écrire des scripts de génération d'image hors ligne, par exemple, ou bien traiter des textes en fonctions d'information que vous leur passeriez), vous aurez besoin d'un exécutable PHP. Pour plus de détails, lisez la section écrire des applications PHP en ligne de commande . Dans ce cas, vous n'aurez pas besoin de serveur web, ni de navigateur.

Avec PHP, vous pouvez aussi écrire des interfaces graphiques clientes, en utilisant l'extension PHP−GTK. C'est une approche complètement différente de l'écriture des pages web, car vous ne générerez pas de fichiers HTML, mais vous aurez à gérer des fenêtres et des objets. Pour plus de détails sur PHP−GTK, voyez le site officiel . PHP−GTK n'est pas inclus dans la distribution officielle de PHP.

A partir de maintenant, cette section décrit l'installation de PHP avec un serveur web sous Unix et Windows, sous forme de module ou d'exécutables CGI.

Les codes source et les exécutables compilés de certains OS (y compris Windows), sont disponibles à http://www.php.net/downloads.php . Nous recommandons l'utilisation du miroir le plus proche pour accélérer les chargements.

## 3.4.2 Installation sous UNIX

Cette section va vous guider lors du processus d'installation et de configuration de PHP sous Unix. Commencez par étudier les sections spécifiques à votre plate−forme ou à votre serveur web avant de passer à l'installation.

Pré−requis :

- Connaissance de base d'UNIX (savoir faire un "make" et compiler en C, si besoin).
- Un compilateur ANSI C (pour les codes sources)
- flex (pour compiler)
- bison (pour compiler)
- Un serveur web
- Tous les composants nécessaires aux extensions (librairie GD , PDF , etc...)

Il y a plusieurs façons d'installer PHP sur une plate−forme UNIX : soit un processus de compilation−configuration, ou bien avec des packages déjà tout prêts. Cette documentation se concentre sur la première solution.

La première partie du processus est faite en ligne de commande, grâce aux options du script configure . Cette section présente l'utilisation des options les plus courantes, mais il y en a beaucoup d'autres à essayer. Reportez−vous à la liste complète des options de configuration pour une liste exhaustive. Voici les différentes méthodes d'installation de PHP :

- Comme module Apache
- Comme module fhttpd
- Pour l'utiliser avec AOLServer, NSAPI, phttpd, Pi3Web, Roxen, thttpd, ou Zeus.
- Comme exécutable CGI

### 3.4.2.1 Installer PHP comme module Apache

PHP peut être compilé de nombreuses manières différentes, mais la plus populaire est le module Apache. La liste suivante est un récapitulatif de l'installation.

Installer PHP comme module Apache

```
 1.  gunzip apache_1.3.x.tar.gz
2.  tar xvf apache_1.3.x.tar
3.  gunzip php−x.x.x.tar.gz
4.  tar xvf php−x.x.x.tar
5.  cd apache_1.3.x
6.  ./configure −−prefix=/www
7.  cd ../php−x.x.x
8.  ./configure −−with−mysql −−with−apache=../apache_1.3.x −−enable−track−vars
9.  make
10. make install
11. cd ../apache_1.3.x
12. ./configure −−activate−module=src/modules/php4/libphp4.a
13. make
14. make install
15. cd ../php−x.x.x
16. cp php.ini−dist /usr/local/lib/php.ini
17. "Editez votre fichier httpd.conf ou srm.conf et ajoutez : "
      AddType application/x−httpd−php .php
18. "Utilisez votre procédure habituelle pour redémarrer le serveur Apache.
    (vous devez arrêter puis redémarrer le serveur, et pas seulement forcer
    le serveur à relire la configuration initiale).
```

#### 3.4.2.2 Compilation

Lorsque PHP est configuré, vous êtes prêts à compiler l'exécutable CGI. La commande make doit prendre tout en charge. Si ce n'est pas le cas et que vous restez bloqués, reportez−vous aux problèmes courants .

## 3.4.3 Installation sous Linux/Unix

Cette section contient les notes et conseils d'installation de PHP sur les distributions Linux.

#### 3.4.3.1 Utilisation des packages

De nombreuses distributions Linux disposent d'un système d'installation par package, comme le fameux RPM. Ils vous permettent de faire des installations standard, mais si vous avez des configurations spécifiques (comme par exemple un serveur sécurisé, ou un pilote de bases de données exotique), vous aurez probablement à compiler vous−même votre PHP et votre serveur web. Si vous n'êtes pas familier avec la compilation de vos propres logiciels, il vaut mieux rechercher le package qui pourra répondre à vos besoins.

## 3.4.4 Installation sous HP−UX

Cette section contient les notes et conseils d'installation de PHP sur les distributions HP−UX. (Fournis par paul_mckay at clearwater−it dot co dot uk).

| Note |
| --- |
| Ces conseils sont destinés à PHP 4.0.4 et Apache v1.3.9. |

1. Vous aurez besoin de gzip. Téléchargez la distribution compilée à http://hpux.connect.org.uk/ftp/hpux/Gnu/gzip−1.2.4a/gzip−1.2.4a−sd−10.20.depot.Z , puis décompressez la, et utilisez swinstall pour installer.
2. Vous aurez besoin de gcc. Téléchargez une distribution compilée à http://gatekeep.cs.utah.edu/ftp/hpux/Gnu/gcc−2.95.2/gcc−2.95.2−sd−10.20.depot.gz , puis décompressez la, et utilisez swinstall pour installer.
3. Vous aurez besoin de GNU binutils. Téléchargez une distribution compilée à http://hpux.connect.org.uk/ftp/hpux/Gnu/binutils−2.9.1/binutils−2.9.1−sd−10.20.depot.gz , puis décompressez la, et utilisez swinstall pour installer.
4. Vous aurez besoin de bison. Téléchargez une distribution compilée à http://hpux.connect.org.uk/ftp/hpux/Gnu/bison−1.28/bison−1.28−sd−10.20.depot.gz , puis décompressez la, et utilisez swinstall pour installer.
5. Vous aurez besoin de flex. Téléchargez une distribution source sur l'un des miroirs http://www.gnu.org. Il se trouve dans le dossier non−gnu du site FTP. Téléchargez le fichier, gunzip , décompressez le, puis utilisez tar −xvf avec. Allez dans le nouveau dossier flex ainsi créé, et exécutez la commande ./configure , puis faites un make , puis un make install ..

   Si vous avez des erreurs à cette étape, c'est probablement parce que gcc et les autres ne sont pas inscrites dans votre PATH. Ajoutez les. Maintenant, la partie délicate.
6. Téléchargez les sources d'Apache et de PHP.
7. Décompressez les avec gunzip et tar −xvf . Nous devons modifier quelques fichiers avant de les compiler.

8. Premièrement, le fichier de configuration doit être modifié car il semble oublier qu'il est sur une machine HP−UX. Il y a des méthodes plus rusées, mais le plus simple et le plus efficace est d'ajouter " lt_target=hpux10.20 " à la ligne 47286 du script de configuration.
9. Le fichier d'Apache GuessOS doit être modifié. Sous apache_1.3.9/src/helpers , modifier la ligne 89, en remplaçant " echo "hp${HPUXMACH}−hpux${HPUXVER}"; exit 0 " par : " echo "hp${HPUXMACH}−hp−hpux${HPUXVER}"; exit 0 "
10. Il n'est pas possible d'installer PHP sous forme de shared object sous HP−UX, ce qui fait que vous devez le compiler en statique. Suivez simplement les instructions de la section Apache.
11. PHP et Apache sont maintenant compilés correctement, mais Apache ne démarre pas. Vous devez créer un nouvel utilisateur Apache, par exemple www, ou apache. Puis, modifiez les lignes 252 et 253 de conf/httpd.conf pour remplacer
    User nobody
    Group nogroup

    par vos valeurs, par exemple :

    User www
    Group sys

    Il n'est pas possible d'exécuter Apache avec l'utilisateur nobody sous HP−UX. A partir de ce moment là, PHP et Apache doivent fonctionner.

## 3.4.5 Installation sous Solaris

Cette section contient les notes et conseils d'installation de PHP sur les distributions Solaris.

### 3.4.5.1 Logiciels nécessaires

L'installation Solaris oublie généralement les compilateurs C, et leurs utilitaires. Voici la liste des outils nécessaires :

- gcc (recommandé, mais d'autres compilateurs C peuvent fonctionner)
- make
- flex
- bison
- m4
- autoconf
- automake
- perl
- gzip
- tar

De plus, vous devrez aussi installer (et peut être aussi compiler) toutes les librairies nécessaires aux extensions (MySQL, ORACLE..).

### 3.4.5.2 Utilisation des packages

Vous pouvez simplifier l'installation Solaris en utilisant pkgadd pour installer la plupart des composants.

## 3.4.6 Installations Unix/OpenBSD

Cette section contient les notes spécifiques à l'installation de PHP sous OpenBSD 3.4 .

### 3.4.6.1 Utilisation des paquets binaires

Cette méthode est la méthode recommandée pour installer PHP sur OpenBSD. C'est aussi la méthode la plus simple. Le paquet core a été séparé des modules et chacun d'entre eux peut être installé et supprimé indépendamment des autres. Les fichiers dont vous avez besoin sont sur le CD OpenBSD ou sur le site FTP.

Le package principal que vous devez installer est php4−core−4.3.3.tgz , qui contient le moteur de base, plus gettext et iconv). Puis, jetez un oeil aux packages de module, comme php4−mysql−4.3.3.tgz ou php4−imap−4.3.3.tgz . Vous devez utiliser la commande phpxs pour activer et désactiver ces modules dans votre php.ini .

A cause d'une erreur dans les packages OpenBSD 3.4, le package php4−gd−4.3.3−no_x11 n'a pas été généré. Vous pouvez encore utiliser le package php4−gd−4.3.3 , mais il requière l'installation de xbase34.tgz . La branche −stable de l'arbre de ports a corrigé cette erreur et un package de remplacement est en cours de génération.

Exemple d'installation de PHP sous OpenBSD avec Ports

```
 # pkg_add php4−core−4.3.3.tgz
# /usr/local/sbin/phpxs −s
# cp /usr/local/share/doc/php4/php.ini−recommended /var/www/conf/php.ini
  (add in mysql)
# pkg_add php4−mysql−4.3.3.tgz
# /usr/local/sbin/phpxs −a mysql
  (add in imap)
# pkg_add php4−imap−4.3.3.tgz
# /usr/local/sbin/phpxs −a imap
  (remove mysql as a test)
# pkg_delete php4−mysql−4.3.3
# /usr/local/sbin/phpxs −r mysql
  (install the PEAR libraries)
# pkg_add php4−pear−4.3.3.tgz
```

Lisez la page de manuel Unix packages(7) pour plus de détails sur les packages binaires d'OpenBSD.

### 3.4.6.2 Utilisation des ports

Vous pouvez aussi compiler PHP en utilisant l'arbre des ports . Cette méthode est recommandée aux utilisateurs expérimentés de OpenBSD. Le port PHP4 est scindé en deux sous−dossiers : core et extensions. Le dossier extensions génère les sous paquets de tous les modules PHP . Si vous souhaitez ne pas créer ces modules, vous pouvez utiliser la commande en ligne no_* FLAVOR. Par exemple, pour ne pas compiler le module imap, utilisez FLAVOR avec la valeur no_imap .

### 3.4.6.3 Versions plus anciennes

Les anciennes versions de OpenBSD utilisaient le système des FLAVORS pour compiler statiquement PHP. Comme il est trop difficile de générer des packages binaires avec cette méthode, elle est considérée comme obsolète. Vous pouvez toujours utiliser les anciennes versions stables, mais sachez qu'elles ne sont plus supportées par l'équipe d'OpenBSD. Si vous avez des commentaires sur le sujet, le responsable actuel est Anil Madhavapeddy .

## 3.4.7 Installation sous Unix et Mac OS X

Cette section contient les notes et conseils pour installer PHP sur un serveur Mac OS X Server.

### 3.4.7.1 Utilisation des paquets

Il existe quelques versions pré−packagées et pré−compilées de PHP pour MacOSX. Cela peut être utile pour mettre en place une configuration standard, mais si vous avez besoin d'accéder à des fonctionnalités spécifiques (comme un serveur sécurisé, ou un pilote de bases de données exotiques), vous aurez à compiler PHP et/ou votre serveur web vous−même. Si vous n'êtes pas familiers avec la compilation et la mise en place d'applications, il est bon de vérifier si personne d'autre n'a réalisé un package.

### 3.4.7.2 Compilation pour OS X server

Il existe deux versions légèrement différentes de Mac OS X, le client et le serveur. Les lignes ci−dessous sont pour OS X Server.

```
Installation sur Mac OS X server
```

```
1. Get the latest distributions of Apache and PHP
2. Untar them, and run the configure program on Apache like so.
./configure −−exec−prefix=/usr \
−−localstatedir=/var \
−−mandir=/usr/share/man \
−−libexecdir=/System/Library/Apache/Modules \
−−iconsdir=/System/Library/Apache/Icons \
−−includedir=/System/Library/Frameworks/Apache.framework/Versions/1.3/Headers \
−−enable−shared=max \
−−enable−module=most \
−−target=apache

4. You may also want to add this line:
setenv OPTIM=−O2
If you want the compiler to do some optimization.

5. Next, go to the PHP 4 source directory and configure it.
./configure −−prefix=/usr \
−−sysconfdir=/etc \
−−localstatedir=/var \
−−mandir=/usr/share/man \
−−with−xml \
−−with−apache=/src/apache_1.3.12

If you have any other additions (MySQL, GD, etc.), be sure to add
them here. For the −−with−apache string, put in the path to your
apache source directory, for example "/src/apache_1.3.12".
6. make
7. make install
This will add a directory to your Apache source directory under
src/modules/php4.

8. Now, reconfigure Apache to build in PHP 4.
./configure −−exec−prefix=/usr \
−−localstatedir=/var \
−−mandir=/usr/share/man \
−−libexecdir=/System/Library/Apache/Modules \
−−iconsdir=/System/Library/Apache/Icons \
−−includedir=/System/Library/Frameworks/Apache.framework/Versions/1.3/Headers \
−−enable−shared=max \
−−enable−module=most \
−−target=apache \
−−activate−module=src/modules/php4/libphp4.a

You may get a message telling you that libmodphp4.a is out of date.
If so, go to the src/modules/php4 directory inside your apache
source directory and run this command:

ranlib libmodphp4.a

Then go back to the root of the apache source directory and run the
above configure command again. That'll bring the link table up to
date.

9. make

10. make install

11. copy and rename the php.ini−dist file to your "bin" directory from your
PHP 4 source directory:
```

```
cp php.ini-dist /usr/local/bin/php.ini

or (if your don't have a local directory)

cp php.ini-dist /usr/bin/php.ini
```

#### 3.4.7.3 Compilation pour MacOS X client

Ces conseils sont fournis gracieusement par Marc Liyanage .

Le module PHP d'Apache est inclus dans l'installation de Mac OS X. Cette version inclut le support des bases de données MySQL et PostgreSQL.

NOTE: Attention lors des manipulations ci−dessous : vous pourriez détruire la configuration de votre serveur.

Faites ceci :

- 1. Ouvrez une fenêtre de terminal
- 2. Tapez "wget http://www.diax.ch/users/liyanage/software/macosx/libphp4.so.gz", et attendez le téléchargement.
- 3. Tapez "gunzip libphp4.so.gz"
- 4. Tapez "sudo apxs −i −a −n php4 libphp4.so"

Maintenant, tapez " sudo open −a TextEdit /etc/httpd/httpd.conf " TextEdit va ouvrir le fichier de configuration de votre serveur web. Recherchez les deux lignes suivantes, à la fin du fichier (utilisez la commande de recherche) :

```
#AddType application/x-httpd-php .php
   #AddType application/x-httpd-php-source .phps
```

Supprimez les deux signes dièse ( # ), puis sauvez le fichier, et quittez TextEdit.

Finalement, tapez " sudo apachectl graceful " pour redémarrer le serveur Apache.

PHP devrait être fonctionnel. Vous pouvez le tester en créant un fichier dans votre dossier "Sites", appelé "test.php". Dans ce fichier, écrivez la ligne ci−contre : " <?php phpinfo() ?> ".

Maintenant, ouvrez votre navigateur à l'adresse suivante : 127.0.0.1/~your_username/test.php . Vous devriez obtenir la page d'information de phpinfo .

## 3.4.8 Liste des options de configuration diverses

Vous trouverez ci−dessus une liste partielle des options de configuration utilisées par le script de configuration PHP configure , lors de la compilation sur les environnements de type Unix. La plupart des options de configuration sont décrites dans leur chapitre et ne sont pas présentées ici. Pour une liste complète des options de configuration, utilisez la commande en ligne ./configure −−help dans votre dossier PHP après avoir exécuté la commande autoconf (voyez aussi le chapitre d'installation ). Vous serez aussi interessés par la lecture de la documentation de GNU configure pour plus de détails sur les options du script configure , telles que −−prefix=PREFIX .

| Note |
| --- |
| Ces options sont utilisées uniquement au moment de la compilation. Si vous voulez modifier le comportement de PHP lors de l'exécution, reportez vous au chapitre configuration . |

Ci−dessous, vous trouverez une liste complète des options supportées par le script de configuration de PHP 4, configure (depuis PHP 4.1.0), utilisé lors de la compilation dans les environnements Unix. Certaines options sont disponibles en PHP 3, d'autres en PHP 4, d'autres encore dans les deux.

Il y a des options de configuration générales pour le script configure , consultez les pages appropriées du manuel de GNU autoconf ou bien utilisez la commande configure −−help pour avoir une liste complète et à jour des options.

- Bases de données
- Graphisme
- Divers
- PHP
- Serveur

### 3.4.8.1 Options de configuration en PHP 4

| Note |
| --- |
| Ces options ne sont utilisées qu'en PHP 4, depuis la version PHP 4.1.0. Certaines sont disponibles dans les anciennes versions PHP 4, certaines même en PHP 3, certaines uniquement en PHP 4.1.0. Si vous essayez de compiler une version plus ancienne, certaines options peuvent simplement ne pas exister. |

#### 3.4.8.1.1 Options graphiques

*−−with−imagick*
: L'extension imagick a été placée dans PECL de PEAR et peut être trouvée ici . Les instructions d'installation pour PHP 4 sont disponibles sur le site de PEAR.

  En bref, −−with−imagick est uniquement supporté en PHP 3, à moins que vous ne suiviez les instructions disponibles sur le site de PEAR.

#### 3.4.8.1.2 Options diverses

*−−enable−debug*
: Compile PHP avec les symboles de débogage.

*−−with−layout=type*
: Indique comment les fichiers installés doivent être rangés. Les valeurs possibles sont Type (valeur par défaut) et GNU.

*−−with−pear=dir*
: Installe PEAR dans le dossier DIR. Par défaut, c'est PREFIX/lib/php.

*−−without−pear*
: Exclut l'installation de PEAR.

*−−enable−sigchild*
: Active le gestionnaire de SIGCHLD propre à PHP.

*−−disable−rpath*
: Désactive le passage de chemins supplémentaires pour chercher les librairies additionnelles.

*−−enable−libgcc*
: Active explicitement le linkage avec libgcc.

*−−enable−php−streams*

Inclut le support des flux de PHP (expérimental). Ne pas utiliser à moins que vous ne testiez le code.

*−−with−zlib−dir=<dir>*
: Définit le chemin d'installation de zlib.

*−−enable−trans−sid*
: Active la propagation transparente des sessions.

*−−with−tsrm−pthreads*
: Utilise les threads POSIX (configuré par défaut).

*−−enable−shared[=pkgs]*
: Compile les librairies partagées. Par défaut, c'est yes.

*−−enable−static[=pkgs]*
: Compile les librairies statiques. Par défaut, c'est yes.

*−−enable−fast−install[=pkgs]*
: Optimise pour l'installation rapide. Par défaut, c'est yes.

*−−with−gnu−ld*
: Suppose que le compilateur utilise GNU ld. Par défaut, c'est no.

*−−disable−libtool−lock*
: Evite les verrouillages (mais cela risque de corrompre des compilations parallèles).

*−−with−pic*
: Essaie d'utiliser uniquement des objets PIC/non−PIC. Par défaut, c'est both (les deux).

*−−enable−memory−limit*
: Compile PHP avec le support de la limitation de mémoire.

*−−disable−url−fopen−wrapper*
: Désactive les gestionnaires HTTP et FTP avec les fonctions d'accès aux fichiers.

*−−enable−versioning*
: Exporte uniquement les symboles nécessaires. Voir le fichier INSTALL pour plus de détails.

*−−with−imsp[=dir]*
: Inclut le support de IMSP. DIR est le dossier d'IMSP qui contient les fichiers d'en−têtes et le fichier libimsp.a. Valable uniquement pour PHP 3!

*−−with−mck[=dir]*
: Inclut le support de Cybercash MCK. DIR est le dossier d'installation de cybercash mck. Par défaut, c'est /usr/src/mck−3.2.0.3−linux. Pour de l'aide, regardez dans extra/cyberlib. Valable uniquement pour PHP 3!

*−−with−mod−dav=dir*
: Inclut le support de DAV, grâce à l'interface Apache mod_dav. DIR est le dossier d'installation de mod_dav. Fonctionne uniquement pour la version module d'Apache. Valable uniquement pour PHP 3!

*−−enable−debugger*
: Compile PHP avec les fonctionnalités de débogage à distance. Valable uniquement pour PHP 3!

*−−enable−versioning*
: Profite du système de versionnage et de gestion d'environnement de Solaris 2.x et Linux. Valable uniquement pour PHP 3!

#### 3.4.8.1.3 Options PHP

*−−enable−maintainer−mode*
: Active des règles de compilation (make) et des dépendances pas toujours utiles aux utilisateurs occasionnels.

*−−with−config−file−path=path*
: Indique le chemin dans lequel réside le fichier php.ini . Par défaut, c'est PREFIX/lib.

*−−enable−safe−mode*

Active le safe mode par défaut.

*−−with−exec−dir[=dir]*
: Autorise uniquement les exécutables situés dans le dossier DIR lorsque le safe mode est activ. Par défaut, c'est DIR vaut /usr/local/php/bin.

*−−enable−magic−quotes*
: Active les guillemets magiques.

*−−disable−short−tags*
: Désactive les balises courtes <? .

#### 3.4.8.1.4 Options Serveur

*−−with−aolserver=dir*
: Spécifie le chemin d'installation du serveur AOLserver.

*−−with−apxs[=file]*
: Compile un module Apache partagé. FILE est un chemin d'accès optionnel vers les outils apxs d'Apache. Par défaut, c'est apxs. Assurez vous de spécifier la version d'apxs qui est réellement installée sur votre système, et non pas celle qui est fournie avec Apache.

*−−with−apache[=dir]*
: Compile le module Apache. DIR est le chemin du dossier d'installation d'Apache. Par défaut, c'est /usr/local/apache.

*−−with−mod_charset*
: Active les tables de transfert pour mod_charset (version russe de Apache).

*−−with−apxs2[=file]*
: Compile le module partagé Apache 2.0. FILE est un chemin d'accès optionnel vers les outils apxs d'Apache. Par défaut, c'est apxs.

*−−with−caudium=dir*
: Compile PHP comme module Pike pour utilisation avec Caudium. DIR est le serveur Caudium. Par défaut, c'est /usr/local/caudium/server .

*−−disable−cli*
: Empêche la compilation de la version CLI de PHP (cela force aussi l'option −−without−pear). Disponible depuis PHP 4.3.0.

*−−enable−embed[=type]*
: Active la compilation de la librairie intégrée SAPI. TYPE vaut soit shared , soit static . Par défaut, c'est shared . Disponible depuis PHP 4.3.0.

*−−with−fhttpd[=dir]*
: Compile le module fhttpd. DIR est le dossier d'installation de fhttpd. Par défaut, c'est /usr/local/src/fhttpd.

*−−with−isapi=dir*
: Compile PHP sous forme de module ISAPI pour utiliser avec le serveur Zeus.

*−−with−nsapi=dir*
: Spécifie le chemin vers le serveur Netscape Server.

*−−with−phttpd=dir*
: Aucune information fournie.

*−−with−pi3web=dir*
: Compile PHP sous forme de module pour utiliser avec le serveur Pi3Web.

*−−with−roxen=dir*
: Compile PHP sous forme de module Pike. DIR est le chemin d'installation de Roxen. Par défaut, c'est /usr/local/roxen/server.

*−−enable−roxen−zts*
: Compile PHP sous forme de module Roxen, en utilisant Zend Thread Safety.

*−−with−servlet[=dir]*
: Inclut le support des servlets. DIR est le chemin d'installation du JSDK. Cette SAPI impose que l'extension java soit compilée comme une librairie partagée.

*−−with−thttpd=srcdir*
Compile PHP comme module thttpd.
*−−with−tux=moduledir*
Compile PHP comme module TUX (Linux uniquement).
*−−with−webjames=srcdir*
Build PHP as a WebJames module (RISC OS only)
*−−disable−cgi*
Empêche la compilation de la version CGI de PHP . Disponible depuis PHP 4.3.0.
*−−enable−force−cgi−redirect*
Active la sécurité lors des redirections internes du serveur. Il est recommandé d'utiliser cette option lorsque vous utilisez PHP en CGI avec Apache.
*−−enable−discard−path*
Si cette option est activée, l'exécutable CGI PHP peut être placé hors de l'arborescence web, en toute sécurité. Il ne sera pas possible de contourner les fichiers .htaccess .
*−−with−fastcgi*
Compile PHP comme application FastCGI. Cette option n'est plus disponible depuis PHP 4.3.0.
*−−enable−fastcgi*
Si cette option est activée, le module CGI sera compilé avec le support FastCGI. Cette option est disponible depuis PHP 4.3.0.
*−−disable−path−info−check*
Si cette option est désactivée, des chemins tels que /info.php/test?a=b ne pourront pas fonctionner. Cette option est disponible depuis PHP 4.3.0. Pour plus d'informations, voir le Manuel Apache .

## 3.4.9 Installation sous Windows

Cette section s'applique aux systèmes Windows 98/Me et Windows NT/2000/XP. Ne vous attendez pas à ce que PHP fonctionne sur les plateformes 16 bits, comme Windows 3.1. Parfois, on décrira les plateformes supportées sous le nom de Win32. Windows 95 n'est plus supporté à partir de la version 4.3.0 de PHP .

Il y a deux méthodes principales pour installer PHP sous Windows : soit manuellement , soit avec InstallShield .

Si vous avez Microsoft Visual Studio, vous pouvez aussi compiler PHP à partir des sources.

Une fois que PHP est installé sur votre Windows, vous pouvez aussi ajouter diverses extensions .

### 3.4.9.1 InstallShield sous Windows

L'installeur Windows de PHP disponible depuis les pages de http://www.php.net/downloads.php , installe la version CGI de PHP , et configure les serveurs web IIS, PWS, et Xitami.

Notez bien que bien que InstallShield soit une méthode simple pour installer PHP , il est limité dans son fonctionnement, puisque l'installation automatique des extensions n'est pas prise en compte. Le jeu complet des extensions PHP est disponible en téléchargeant et décompressant manuellement l'archive.

Installez votre serveur HTTP favori sur votre système et assurez−vous qu'il fonctionne.

Exécutez l'installeur et suivez les instructions fournies par l'assistant. Deux types d'installation sont fournis : standard, qui utilise toutes les configurations par défaut les plus pratiques, et avancée, qui pose un maximum de questions pour paramétrer le plus finement.

L'assistant d'installation rassemble suffisamment d'informations pour configurer php.ini et le serveur web qui utilisera PHP . Pour IIS, mais aussi PWS sous NT Workstation, une liste de l'arboresence web est affichée, et vous pouvez sélectionner les dossiers qui utiliseront PHP .

Une fois l'installation terminée, l'installeur vous informera que vous devez redémarrer. Suivez ce conseil, ou commencez à utiliser PHP immédiatement.

| Attention |
| --- |
| Gardez bien à l'esprit que cette installation de PHP n'est pas sécurisée. Si vous voulez avoir une installation sécurisée de PHP , vous devriez commencer par lire la documentation, et choisir toutes vos options avec soin. Cet installeur automatique vous permet de réaliser l'installation en un tour de main, mais n'est pas destiné à l'utilisation sur des serveurs de production. |

### 3.4.9.2 Installation manuelle sous Windows

Ce guide d'installation vous aide à installer manuellement et configurer PHP sur votre serveur web Windows. La version initiale de ce guide a été compilée par Bob Silva , et est disponible en anglais, à http://www.umesd.k12.or.us/php/win32install.html . Vous devrez télécharger la distribution binaire Zip sur la page http://www.php.net/downloads.php .

PHP 4 pour Windows est distribué en trois versions : exécutable CGI , un exécutable CLI (sapi/php.exe) et d'autres modules SAPI :

- php4apache.dll − module Apache 1.3.x.
- php4apache2.dll − module Apache 2.0.x.
- php4isapi.dll − module ISAPI compatible avec les serveurs comme IIS 4.0/PWS 4.0 ou plus récents.
- php4nsapi.dll − module Netscape/iPlanet.

Cette dernière forme est nouvelle en PHP 4, et fournit des performances améliorées et de nouvelles fonctionnalités. La version CLI est conçue pour utiliser PHP en ligne de commande. Plus d'informations sur la forme CLI est disponible dans le chapitre " utiliser PHP en ligne de commande ".

| Attention |
| --- |
| Les modules SAPI ont été significativement améliorés dans la version 4.1, mais vous pourrez rencontrer des erreurs avec le serveur ou d'autres modules (tels ASP), dans les autres systèmes. |
| Note |
| Pré−requis DCOM et MDAC<br><br>Si vous choisissez d'installer le module SAPI sur **Windows 95** , assurez vous de télécharger et installer la mise à jour DCOM, disponible à Microsoft DCOM pages . Si vous utilisez Microsoft **Windows 9x/NT4** , téléchargez la dernière version de Microsoft Data Access Components (MDAC) pour votre plate−forme. MDAC est disponible à http://www.microsoft.com/data/ . |

Voici les différentes étapes d'installation avant les étapes spécifiques au serveur.

- Extrayez la distribution dans un dossier de votre choix, comme par exemple c:\ . L'archive zip devient un dossier appelé php−4.3.1−Win32 qui est supposé être renommé en php . Par sourci de clarté, nous supposerons alors que vous avez placé votre distribution dans le dossier c:\php . Vous pouvez choisir un autre dossier, mais vous serez prudents d'éviter les

espaces dans le nom du chemin au dossier (comme par exemple c:\program files\php n'est pas une bonne idée). car certains serveurs web vont crasher sinon. La structure du dossier que vous avez extrait ressemble à ceci :

```
c:\php
  |
  +--cli
  |  |
  |  |-php.exe           -- Executable CLI - UNIQUEMENT pour la ligne de commande
  |
  |
  +--dlls                -- DLL de support des extensions  --> dossier systeme Windows
  |  |
  |  |-expat.dll
  |  |
  |  |-fdftk.dll
  |  |
  |  |-...
  |
  +--extensions          -- extensions DLL pour PHP
  |  |
  |  |-php_bz2.dll
  |  |
  |  |-php_cpdf.dll
  |  |
  |  |-..
  |
  +--mibs                -- fichiers de support de SNMP
  |
  |
  +--openssl             -- fichiers de support de Openssl
  |
  |
  +--pdf-related         -- fichiers de support de PDF
  |
  |
  +--sapi                -- DLL SAPI
  |  |
  |  |-php4apache.dll
  |  |
  |  |-php4apache2.dll
  |  |
  |  |-php4isapi.dll
  |  |
  |  |-..
  |
  |-install.txt
  |
  |-..
  |
  |-php.exe              -- executable CGI
  |
  |-..
  |
  |-php.ini-dist
  |
  |-php.ini-recommended
  |
   |-php4ts.dll           -- DLL principale --> dossier systeme Windows
  |
   |-...
```

Le binaire CGI ( C:\php\php.exe ), le binaire CLI ( c:\php\cli\php.exe ), et les modules SAPI ( c:\php\sapi*.dll ) dépendent de la DLL principale c:\php\php4ts.dll . Vous devez vous assurer que cette DLL est disponible pour votre installation PHP . L'ordre des dossiers lors de la recherche est le suivant :

- Le dossier d'où php.exe est appelé. Lorsque vous utilisez le module SAPI, c'est le dossier d'où le serveur web charge la DLL (i.e. php4apache.dll ).
- Tout dossier dans la variable d'environnement Windows PATH .

- La meilleure chose à faire est de rendre php4ts.dll accessible, indépendemmant de l'interface utilisée (CGI ou module SAPI). Pour cela, vous devez copier la DLL dans un dossier de votre PATH Windows. La meilleure place est alors le dossier système Windows :
    - c:\windows\system pour Windows 9x/ME
    - c:\winnt\system32 pour Windows NT/2000 ou c:\winnt40\system32 pour NT/2000 server
    - c:\windows\system32 pour Windows XP

Si vous voulez utiliser le module SAPI depuis c:\php\sapi et que vous ne voulez pas copier vos DLL dans le dossier système Windows, vous avez la possibilité de copier simplement php4ts.dll dans le dossier SAPI de votre archive décompressée c:\php\sapi .

- L'étape suivante est de faire un fichier de configuration valide pour PHP , c'est à dire le fichier php.ini . Il y a deux versions de ce fichier disponibles dans l'archive : php.ini−dist et php.ini−recommended . Nous vous recommandons d'utiliser php.ini−recommended , car nous avons optimisé les configurations par défaut dans ce fichier pour les performances et la sécurité. Lisez bien ce fichier documenté, et étudiez toutes les configurations , puis faites vos propres choix. Si vous voulez atteindre la meilleure sécurité, c'est la meilleure méthode. Copiez votre fichier de configuration de choix dans un dossier où PHP est capable de le lire, et renommez le php.ini . Par défaut, PHP recherche php.ini dans votre dossier Windows :
    - Sur Windows 9x/ME/XP, copiez le fichier php.ini dans votre dossier %WINDIR% , qui est généralement c:\windows .
    - Sur Windows NT/2000, copiez le fichier php.ini dans votre dossier %WINDIR% ou %SYSTEMROOT% , qui est typiquement c:\winnt ou c:\winnt40 pour les serveurs NT/2000.
- Si vous utilisez NTFS sur Windows NT, 2000 ou XP, assurez vous que l'utilisateur qui fait tourner le serveur web a les droits de lecture de votre fichier php.ini (e.g. rendez le lisible par tout le monde).

Les étapes suivantes sont optionnelles :

- Editez votre nouveau fichier php.ini . Si vous envisagez d'utiliser OmniHTTPd , ignorez l'étape suivante. Configurez doc_root pour qu'il pointe sur la racine de votre serveur web. Par exemple :

```
doc_root = c:\inetpub        // pour IIS/PWS

doc_root = c:\apache\htdocs // pour Apache
```

- Choisissez les extensions que vous voulez que PHP démarre. Voyez la section sur les extensions Windows , pour les configurer, si elles ne sont pas intégrées. Notez que sur une nouvelle installation, il est recommandé de commencer par faire fonctionner PHP avant d'ajouter ou retirer des extensions via le php.ini .
- Sur PWS et IIS, vous pouvez utiliser la directive browscap pour qu'elle pointe sur : c:\windows\system\inetsrv\browscap.ini sur Windows 9x/Me, c:\winnt\system32\inetsrv\browscap.ini sur NT/2000, et c:\windows\system32\inetsrv\browscap.ini sur XP.

Copiez le fichier php.ini−dist dans votre dossier %WINDOWS sous Windows 95/98, ou vers votre dossier %SYSTEMROOT% sous Windows NT ou Windows 2000 et renommez−le en php.ini . Votre dossier %WINDOWS% ou %SYSTEMROOT% est généralement :

- c:\windows pour Windows 9x/Me
- c:\winnt ou c:\winnt40 pour les serveurs NT/2000

Il y a deux fichiers ini, distribués dans le fichier zip, php.ini−dist et php.ini−optimized . Nous vous recommandons d'utiliser php.ini−optimized , car nous l'avons optimisé d'un point de vue sécurité et vitesse d'exécution. Le mieux est tout de même de lire le fichier de configuration php.ini et de choisir vos options manuellement. Si vous voulez atteindre un meilleur niveau de sécurité, c'est par ce moyen que vous pourrez le faire, même si PHP fonctionne parfaitement avec les configurations par défaut.

Editez votre fichier php.ini :

- la famille des serveurs Windows , Personal Web server (PWS) 3 et 4 ou plus récents; Internet Information Server (IIS) 3 et 4 ou plus récents.
- les serveurs Apache Apache 1.3.x , et Apache 2.x .

- les serveurs Netscape/iPlanet .
- le serveur OmniHTTPd .
- le serveur Oreilly Website Pro .
- le serveur Sambar .
- le serveur Xitami .

### 3.4.9.3 Compilation des sources

Avant de commencer, il est bon de se poser la question suivante : "Pourquoi la compilation de PHP sous Windows est si difficile?". Deux raisons viennent immédiatement à l'esprit :

1. Windows ne dispose pas (encore) d'une grande communauté de développeurs qui partagent librement leurs sources. La conséquence directe est que les investissements nécessaires en infrastructure pour supporter ce type de développement n'ont pas été faits. Ce qui fait que le portage des utilitaires Unix a été la solution pour pallier ce manque. Ne soyez donc pas surpris de rencontrer cette parenté de temps en temps.
2. La majorité des instructions que vous allez rencontrer sont du type : "faire et oublier". Alors, asseyez−vous confortablement et suivez aussi scrupuleusement que possible les instructions.

#### 3.4.9.3.1 Pré−requis

Pour compiler l'exécutable PHP , vous avez besoin d'un Microsoft Development Environment. Microsoft Visual C++ 6.0 est recommendé. Pour extraire le fichier téléchargé, vous avez besoin d'un outil de décompression (e.g.: Winzip). Si vous n'avez pas encore d'utilitaire de décompression, vous pouvez en télécharger un chez InfoZip .

Avant de commencer, vous devez télécharger...

- les win32 buildtools de PHP , sur http://www.php.net/extra/win32build.zip .
- le code source du résolveur de noms DNS utilisé par PHP sur http://www.php.net/extra/bindlib_w32.zip . C'est un remplacement pour la librairie resolv.lib inclue dans win32build.zip .
- Si vous prévoyez de compiler PHP comme module pour Apache, vous aurez aussi besion des sources d'Apache .

Finalement, vous aurez besoin des sources PHP 4 elles−mêmes. Les dernières versions sont accessibles sur le serveur CVS anonyme . Vous pouvez aussi télécharger une version intermédiaire ou l'archive des sources la plus récente.

#### 3.4.9.3.2 Compiler tout ensemble

Après téléchargement des paquets nécessaires, vous devez les extraire au bon endroit.

- Créez un dossier où tous les fichiers seront placés après l'extraction, comme par exemple : c:\work .
- Créez le dossier win32build dans votre dossier de travail (ici c:\work ) et décompressez win32build.zip dedans.
- Créez le dossier bindlib_w32 sous votre dossier de travail (ici c:\work ) et décompressez bindlib_w32.zip dedans.
- Décompressez le code source de PHP dans votre dossier de travail ( c:\work ).

Après avoir suivi ces instructions, vous devriez obtenir un dossier qui ressemble à ceci :

```
+--c:\work
|  |
|  +--bindlib_w32
|  |  |
|  |  +--arpa
|  |  |
|  |  +--conf
|  |  |
|  |  +--...
|  |
|  +--php-4.x.x
|  |  |
|  |  +--build
|  |  |
|  |  +--...
|  |  |
|  |  +--win32
|  |  |
|  |  +--...
|  |
|  +--win32build
|  |  |
|  |  +--bin
|  |  |
|  |  +--include
|  |  |
|  |  +--lib
```

Créez les dossiers c:\usr\local\lib . Copiez le fichier bison.simple depuis c:\work\win32build\bin vers c:\usr\local\lib .

| Note |
| --- |
| Les utilisateurs de Cygwin peuvent omettre la dernière étable. Un environnement Cygwin correctement installé fournit les fichiers obligatoires de bison.simple et bison.exe . |

#### 3.4.9.3.3 Configuration de MVC ++

Lancez Microsoft Visual C++, et allez dans le menu "select Tools => Options". Dans le dialogue, sélectionnez l'onglet "directories". Modifiez les valeurs pour les items du menu Executables, Includes et Library files. Votre configuration doit alors ressembler à

- Fichiers exécutables : c:\work\win32build\bin , Utilisateurs Cygwin : cygwin\bin
- Fichiers include: c:\work\win32build\include
- Fichiers library : c:\work\win32build\lib

#### 3.4.9.3.4 Compilation de resolv.lib

Vous devez compiler la librairie resolv.lib . Decidez si vous avez besoin des symboles de débogage (bindlib − Win32 Debug) ou non (bindlib − Win32 Release). Compilez la configuration ad hoc :

- Pour les utilisateurs de l'interface graphique, lancez VC++, puis sélectionnez le menu "File => Open Workspace" et "bindlib". Puis sélectionnez "Build=>Set Active Configuration" et sélectionnez la configuration voulue. Enfin, sélectionnez "Build=>Rebuild All".
- Pour les utilisateurs en ligne de commande, assurez−vous que vous avez enregistré les variables d'environnement C++, ou que vous avez exécuté vcvars.bat . Exécutez maintenant l'une des commandes suivantes :
    - ♦ msdev bindlib.dsp /MAKE "bindlib − Win32 Debug"
    - ♦ msdev bindlib.dsp /MAKE "bindlib − Win32 Release"

A ce stade, vous avez une librairie resolv.lib utilisable, soit dans votre dossier Debug , soit sans le dossier Release . Copiez ce fichier dans votre dossier win32build\lib , en remplaçant le fichier du même nom.

#### 3.4.9.3.5 Compilation

La meilleure façon de compiler est de commander par la version CGI/exécutable.

- Pour les utilisateurs GUI, lancez VC++, puis sélectionnez le menu "File => Open Workspace" et sélectionnez "php4ts". Ensuite, sélectionnez le menu "Build=>Set Active Configuration", et sélectionnez la configuration voulue. Finalement, sélectionnez le menu "Build=>Rebuild All".
- Pour les utilisateurs en ligne de commande, assurez−vous que vous avez enregistré les variables d'environnement C++, ou que vous avez exécuté vcvars.bat . Exécutez maintenant l'une des commandes suivantes :
    - msdev php4ts.dsp /MAKE "php4ts − Win32 Debug_TS"
    - msdev php4ts.dsp /MAKE "php4ts − Win32 Release_TS"
    - A ce stade, vous avez un exécutable php.exe utilisable, soit dans votre dossier Debug_TS soit sans le dossier Release_TS .

Il est possible de faire des modifications particulières mineures en éditant le fichier main/config.win32.h.in .

Ensuite, vous pouvez compiler la version CLI qui est conçue pour un usage en ligne de commande . Les étapes sont les mêmes que pour la compilation en mode CGI, hormis le fait que vous devez sélectionner php4ts_cli − Win32 Debug_TS ou php4ts_cli − Win32 Release_TS dans le fichier de projet. Après une compilation réussie, vous pouvez exécuter le fichier php.exe dans l'un des dossiers Release_TS\cli\ ou Debug_TS\cli\ .

| Note |
| --- |
| Si vous voulez utiliser PEAR et l'installeur en ligne de commande, la version CLI−SAPI est obligatoire. Pour plus d'informations sur le projet PEAR et l'installeur, lisez la documentation sur le site de PEAR . |

Afin de compiler le module SAPI ( php4isapi.dll pour l'intégrer dans Microsoft IIS, vous devez utiliser la configuration php4isapi−whatever−config et compiler la DLL nécessaire.

### 3.4.9.4 Installation des extensions sous Windows

Après avoir installé PHP et votre serveur web sous Windows, vous voudrez sûrement ajouter quelques extensions bien pratiques. La table suivante liste une partie des extensions disponibles. Comme indiqué dans le manuel, vous pouvez choisir quelles extensions vous voulez charger en décommentant la ligne 'extension=php_*.dll' dans le fichier php.ini . Vous pouvez aussi charger dynamiquement un module avec la fonction dl .

Les fichiers DLLs des extensions PHP sont préfixés par php_ en PHP 4, et php3_ en PHP 3. Cela évite la confusion des extensions PHP et de leurs librairies.

| Note |
| --- |
| En PHP 4.3.1, les extensions BCMath, Calendar, COM, Ctype, FTP, MySQL, ODBC, Overload, PCRE, Session, Tokenizer, WDDX, XML et Zlib sont **intégrées** . Vous n'avez pas besoin d'ajouter d'autres extensions pour pouvoir utiliser ces fonctions. Voyez le fichier README.txt ou install.txt de votre distribution, ou bien cette table pour une liste des modules intégrés. |

Editez votre fichier php.ini :

- Vous devez modifier l'option de extension_dir pour qu'elle pointe sur le dossier qui contient les extensions, là où vous avez placé vos fichiers php_*.dll . N'oubliez surtout pas

l'anti−slash final. Par exemple :

```
extension_dir = c:/php/extensions/
```

- Pour activer les extensions dans le php.ini , il suffit de décommenter les lignes extension=php_*.dll dans le php.ini . Ceci est fait en effaçant le point−virgule initial de la ligne :

```
Activer l'extension Bzip2 sous Windows
 // changez la ligne  ...
;extension=php_bz2.dll

// ... en
extension=php_bz2.dll
```

- Certaines de ces extensions requièrent des librairies DLL supplémentaires pour fonctionner correctement. Certaines d'entre elles sont disponibles dans la distribution. dans le dossier dlls mais certaines (comme Oracle (php_oci8.dll)), demandent des dlls qui ne sont pas dans la distribution.

  Copiez les dlls fournies depuis le dossier dlls dans votre PATH Windows. Les bons emplacements sont typiquement

  - c:\windows\system pour Windows 9x/Me
  - c:\winnt\system32 pour Windows NT/2000
  - c:\windows\system32 pour Windows XP

  Si vous les avez déjà d'installées sur votre système, ne les écrasez que si PHP ne fonctionne pas correctement (et de toutes manières, faites une sauvegarde de ces DLL, en cas de problème).

| Extension | Description | Notes |
|---|---|---|
| php_bz2.dll | Fonctions de compression bzip2 | Aucune |
| php_calendar.dll | Fonctions de conversions calendaires Calendar | Disponible depuis PHP 4.0.3 |
| php_cpdf.dll | Fonctions ClibPDF | Aucune |
| php_crack.dll | Fonctions de CrackLib | Aucune |
| php3_crypt.dll | Fonctions de Crypt | Inconnu |
| php_ctype.dll | Fonctions ctype | Aucune |
| php_curl.dll | CURL, Bibliothèque de fonctions "Client URL" | Requiert : libeay32.dll , ssleay32.dll (fournies) |
| php_cybercash.dll | Fonctions de paiement Cybercash | Aucune |
| php_db.dll | Fonctions DBM | Obsolètes. Utilisez DBA à la place ( php_dba.dll ) |
| php_dba.dll | DBA: DataBase Abstraction, fonctions d'abstraction de base de données (de type dbm) | Aucune |
| php_dbase.dll | Fonctions dBase | Aucune |
| php3_dbm.dll | Fonctions Berkeley DB2 | Inconnu |
| php_dbx.dll | Fonctions dbx | |
| php_domxml.dll | Fonctions de DOM XML | Requiert : libxml2.dll (fournie) |
| php_dotnet.dll | Fonctions .NET | Aucune |
| php_exif.dll | Fonctions de lecture des en−têtes JPEG Read EXIF | Aucune |
| php_fbsql.dll | Fonctions de FrontBase | Aucune |
| php_fdf.dll | Fonctions FDF: Forms Data Format. | Requiert : fdftk.dll (fournies) |
| php_filepro.dll | Fonctions filePro | Accès en lecture seule |
| php_ftp.dll | Fonctions FTP | Disponible depuis PHP 4.0.3 |
| php_gd.dll | Fonctions de traitement des images GD | Aucune |
| php_gd2.dll | Fonctions de traitement des images GD2 | Aucune |
| php_gettext.dll | Fonctions Gettext | PHP <= 4.2.0 requiert : gnu_gettext.dll (fournie), PHP >= 4.2.3 requiert libintl−1.dll (fournie). |
| php_hyperwave.dll | Fonctions HyperWave | Aucune |
| php_iconv.dll | Fonctions de conversions de jeux de caractères ICONV | Requiert : iconv−1.3.dll (fournies) |
| php_ifx.dll | Fonctions Informix | Requiert : les bibliothèques Informix |
| php_iisfunc.dll | Fonctions de gestion de IIS | Aucune |
| php_imap.dll | Fonctions IMAP, POP3 et NNTP | PHP 3 : php3_imap4r1.dll |
| php_ingres.dll | Fonctions Ingres II | Requiert : librairies Ingres II |
| php_interbase.dll | Fonctions InterBase | Requiert : gds32.dll (fournies) |
| php_java.dll | Extension Java | Requiert : jvm.dll (fournies) |
| php_ldap.dll | Fonctions LDAP | PHP <= 4.2.0 requiert libsasl.dll (fournie), PHP >= 4.3.0 requiert libeay32.dll , ssleay32.dll (fournies). |
| php_mbstring.dll | Fonctions chaînes multi−octets | Aucune |
| php_mcrypt.dll | Fonctions de chiffrement Mcrypt | Requiert : libmcrypt.dll |
| php_mhash.dll | Fonctions Mhash | Aucune |
| php_mime_magic.dll | Fonctions Mimetype | Requiert : magic.mime (fournie) |

| | | |
|---|---|---|
| php_ming.dll | Fonctions de génération de fichiers Flash avec Ming | Aucune |
| php_msql.dll | Fonctions de mSQL | Requiert : msql.dll (fournies) |
| php3_msql1.dll | mSQL 1 client | Inconnu |
| php3_msql2.dll | mSQL 2 client | Inconnu |
| php_mssql.dll | Fonctions MSSQL | Requiert : ntwdblib.dll (fournies) |
| php3_mysql.dll | Fonctions MySQL | Intégré à PHP 4 |
| php3_nsmail.dll | Fonctions de email Netscape | Inconnu |
| php3_oci73.dll | Fonctions Oracle | Inconnu |
| php_oci8.dll | Fonctions Oracle 8 | Requiert : librairies clientes Oracle 8 |
| php_openssl.dll | Fonctions OpenSSL | Requiert : libeay32.dll (fournies) |
| php_oracle.dll | Fonctions de Oracle 7 | Requiert : librairies clientes Oracle 7 |
| php_overload.dll | Fonctions relatives à la surcharge d'objets | Intégrée depuis PHP 4.3.0 |
| php_pdf.dll | Fonctions de PDF | Aucune |
| php_pgsql.dll | Fonctions de PostgreSQL | Aucune |
| php_printer.dll | Fonctions d'impressions | Aucune |
| php_shmop.dll | Fonctions de mémoire partagée | Aucune |
| php_snmp.dll | Fonctions SNMP | NT uniquement! |
| php_sockets.dll | Fonctions de socket | Aucune |
| php_sybase_ct.dll | Fonctions Sybase | Requiert : librairies clientes Sybase |
| php_tokenizer.dll | Tokenizer | Intégrée depuis PHP 4.3.0 |
| php_w32api.dll | Fonctions W32api | Aucune |
| php_xmlrpc.dll | Fonctions XML-RPC | PHP >= 4.2.1 requiert iconv.dll (fournie) |
| php_xslt.dll | Fonctions XSLT | Requiert : sablot.dll , expat.dll (fournies) |
| php_yaz.dll | Fonctions YAZ | Aucune |
| php_zib.dll | Fonctions de compression Zip | Accès en lecture seule |
| php_zlib.dll | Fonctions de compression ZLib | Aucune |

## 3.4.10 Installation avec le serveur Apache

Cette section contient des notes spécifiques pour l'installation de PHP avec Apache, aussi bien pour la version Unix que Windows .

### 3.4.10.1 Détails pour l'installation de PHP sous Apache sous Unix.

Il y a deux méthodes d'installation de PHP avec Apache 1.3.x sous Windows. L'une passe par les CGI (php.exe), l'autre est d'utiliser la DLL de module Apache. Dans les deux cas, vous devrez arrêter le serveur Apache, et éditer le fichier srm.conf ou httpd.conf , pour configurer Apache avec PHP.

Vous pouvez sélectionner des options à ajouter au fichier configure à la ligne 8 depuis la liste complète des options de configuration .

```
Instructions d'installation avec Apache (version module)
 1.  gunzip apache_1.3.x.tar.gz
2.  tar xvf apache_1.3.x.tar
3.  gunzip php-x.x.x.tar.gz
4.  tar xvf php-x.x.x.tar
5.  cd apache_1.3.x
6.  ./configure --prefix=/www
7.  cd ../php-x.x.x
8.  ./configure --with-mysql --with-apache=../apache_1.3.x --enable-track-vars
9.  make
10. make install
11. cd ../apache_1.3.x
12. for PHP 3: ./configure --activate-module=src/modules/php3/libphp3.a
    for PHP 4: ./configure --activate-module=src/modules/php4/libphp4.a
13. make
14. make install
  Au lieu de cette etape, vous pouvez aussi copier le binaire
  httpd et remplacer votre executable actuel. Assurez-vous tout
  de meme que le serveur est bien eteint.
15. cd ../php-x.x.x
16. for PHP 3: cp php3.ini-dist /usr/local/lib/php3.ini
    for PHP 4: cp php.ini-dist /usr/local/lib/php.ini
  Vous pouvez editer votre fichier php.ini pour modifier
  certaines options PHP. Si vous preferez placer ce fichier ailleurs,
  utilisez --with-config-file-path=/path lors de l'etape 8.
17. Editez votre fichier httpd.conf ou srm.conf file et ajoutez :
     Pour PHP 3:   AddType application/x-httpd-php3 .php3
```

```
    Pour PHP 4:   AddType application/x-httpd-php  .php
  Vous pouvez choisir n'importe quelle extension que vous voulez ici. .php
  est uniquement une suggestion. Vous pouvez aussi inclure .html.
18. Utilisez votre procedure habituelle pour demarrer votre serveur Apache.
(vous devez l'eteindre et le redemarrer, pas seulement lui envoyer
un signal HUP ou USR1.)
```

Suivant votre installation d'Apache et votre variante d'Unix, il existe de nombreuses façons d'arrêter et redémarrer Apache. Voici une liste des commandes typiques, pour différentes installations. Remplacez /path/to/ par le chemin d'accès à vos applications sur votre système.

```
Redémarrage d'apache
 [1. Nombreuses variantes Linux SysV :
/etc/rc.d/init.d/httpd restart
2. Avec les scripts apachectl :
/path/to/apachectl stop
/path/to/apachectl start
3. httpdctl et httpsdctl (utilisant OpenSSL), similaire a apachectl:
/path/to/httpsdctl stop
/path/to/httpsdctl start
4. En utilisant mod_ssl, ou un autre serveur SSL, manuellement :
/path/to/apachectl stop
/path/to/apachectl startssl
```

Les exécutables apachectl et http(s)dctl peuvent être situés dans différents dossiers. Si votre système est pourvu de locate , de whereis ou de which , utilisez−les pour retrouver vos programmes.

Différents exemples de compilation PHP pour Apache suivent :

```
Compilation sur Apache avec l'extension PostGreSQL
 ./configure --with-apxs --with-pgsql
```

Cette commande va créer une librairie partagée libphp4.so qui sera chargée par Apache avec une ligne LoadModule dans le fichier httpd.conf . Le support PostgreSQL est aussi inclut dans libphp4.so .

```
Compilation sur Apache avec l'extension PostGreSQL (librairie partagée)
 ./configure --with-apxs --with-pgsql=shared
```

Cette commande va créer une librairie partagée libphp4.so pour Apache, mais va aussi créer la librairie partagée pgsql.so qui sera chargée dans PHP avec une directive du fichier php.ini file ou en la chargeant explicitement dans le script avec la fonction dl .

```
Compilation sur Apache avec l'extension PostGreSQL (2)
 ./configure --with-apache=/path/to/apache_source --with-pgsql
```

Cette commande va créer une autre librairie partagée libmodphp4.a , un fichier mod_php4.c et quelques fichiers compagnons dans le dossier src/modules/php4 du dossier Apache. Puis, vous devez compiler Apache avec −−activate−module=src/modules/php4/libphp4.a et le système de compilation d'Apache va créer un fichier libphp4.a et le lier statiquement avec httpd . Le support PostgreSQL est alors inclus directement dans l'exécutable httpd , ce qui fait que le résultat final est un fichier unique httpd , qui inclut Apache et PHP.

```
Compilation sur Apache avec l'extension PostGreSQL (2) (librairie partagée)
 ./configure --with-apache=/path/to/apache_source --with-pgsql=shared
```

Identique à la version précédente, mais au lieu d'inclure le support PostgreSQL directement dans l'exécutable final httpd , vous allez obtenir une librairie partagée pgsql.so que vous pouvez charger dans PHP soit grâce au fichier de configuration php.ini ou dynamiquement avec dl .

Lorsque vous faites le choix entre les différents modes de compilation de PHP, vous devez prendre en compte leurs avantages et inconvénients respectifs. Les objets partagés permettent de compiler

PHP et Apache de manière séparée, et vous n'aurez pas à compiler l'ensemble pour faire évoluer PHP. La compilation statique permet de charger et d'exécuter plus rapidement PHP. Pour plus d'informations, voyez la page web sur le support des DSO .

Note

Le fichier de configuation d'Apache actuel est fourni avec une section qui ressemble à ceci :

```
Configuration de l'utilisateur d'exécution d'Apache
 User nobody
Group "#-1"
```

A moins que vous ne changiez cette valeur par Group nogroup ou quelques chose comme ça ( Group daemon est aussi classique), PHP ne sera pas capable d'ouvrir des fichiers.

Note

Assurez−vous que vous spécifiez la version installée de apxs avec l'option −−with−apxs=/path/to/apxs. Vous ne devez PAS utiliser la version d'apxs qui est dans les sources d'Apache, mais celle qui est réellement installée sur votre système.

### 3.4.10.2 Installer PHP sous Windows avec Apache 1.3.x

Il y a deux méthodes pour faire fonctionner PHP avec Apache 1.3.x sous Windows. La première est d'utiliser l'exécutable CGI (php.exe), l'autre est d'utiliser les modules Apache DLL. Dans les deux cas, vous devez arrêter le serveur Apache, éditer votre fichier srm.conf ou httpd.conf pour configurer Apache.

Bien qu'il puisse y avoir quelques différences de configurations de PHP sous Apache, le processus reste simple et à la portée du néophyte. Reportez−vous aux documentations Apache pour plus de détails sur ces directives.

Si vous avez dézippé le package dans C:\PHP\ comme indiqué dans le chapitre " Installation manuelle sous Windows" , vous devez insérer les lignes suivantes dans votre fichier srm.conf ou httpd.conf pour qu'il fonctionne en CGI :

- ScriptAlias /php/ "c:/php/"
- AddType application/x−httpd−php .php .phtml
- Action application/x−httpd−php "/php/php.exe"

N'oubliez pas de redémarrer le serveur, avec la commande NET STOP APACHE suivie de NET START APACHE .

Attention

En utilisant la configuration en CGI, votre serveur sera ouvert à plusieurs attaques potentielles. Lisez donc la section sécurité des CGI pour savoir comment vous en défendre.

Si vous voulez utiliser PHP comme module Apache, vous devez déplacer le fichier php4ts.dll dans le dossier windows/system (pour Windows 9x/Me) ou winnt/system32 (pour Windows NT/2000), en écrasant les anciennes versions. Puis, vous devez ajouter les deux lignes suivantes dans le fichier de configuration Apache :

- LoadModule php4_module c:/php/sapi/php4apache.dll
- AddType application/x−httpd−php .php .phtml

Après avoir modifié le fichier de configuration, pensez à redémarrer le serveur web, par exemple avec NET STOP APACHE suivi de NET START APACHE , si vous utilisez Apache comme service Windows, ou bien utilisez les alias classiques.

| Note |
| --- |
| Il se peut qu'après avoir utilisé l'installeur Apache pour Windows, vous deviez définir la directive AddModule pour mod_php4.c dans le fichier de configuration ( httpd.conf ). Cela peut être fait en ajoutant AddModule mod_php4.c à la liste des AddModule , juste après le début du fichier de configuration. C'est particulièrement important si la directive ClearModuleList est utilisée. Ne pas le faire reviendrait à ne pas déclarer PHP comme module Apache. |

Il y a deux méthodes pour utiliser la fonctionnalité de coloration syntaxique, elles dépendent de votre installation. Si vous avez configuré Apache comme module ISAPI, alors il suffit d'ajouter la ligne suivante dans votre fichier de configuration : AddType application/x−httpd−php−source .phps

Si vous choisissez de configurer Apache pour utiliser PHP comme exécutable CGI, vous aurez besoin de la fonction show_source . Pour cela, créez simplement un script PHP et ajoutez−y le code suivant : <?php show_source ("original_php_script.php"); ?> . Substitutez original_php_script.php par le nom du fichier que vous souhaitez afficher.

| Note |
| --- |
| Avec le couple Windows−Apache tous les antislashes des noms de chemins tels que " c:\directory\file.ext ", doivent être convertis en slash c'est à dire "c:/directory/file.ext". |

## 3.4.11 Installation avec les serveurs Apache 2.0

Cette section contient les notes et conseils d'installation de PHP avec le serveur Apache 2.0, aussi bien pour Unix que pour Windows versions.

| Attention |
| --- |
| N'utilisez pas Apache 2.0 et PHP sur un serveur de production, ni sur Windows, ni sur Unix. |

Il est très recommandé de lire la documentation Apache pour avoir une meilleure connaissance du serveur web Apache 2.0.

### 3.4.11.1 Notes sur la compatibilité de PHP avec Apache 2.0

Les versions de PHP suivantes sont reconnues pour fonctionner avec la plus récente version d'Apache 2.0 :

- PHP 4.3.0 ou plus récent, disponible à http://www.php.net/downloads.php .
- Les dernières versions de développement. Le code source est disponbile à http://snaps.php.net/php4−latest.tar.gz , et les exécutables compilés de Windows à http://snaps.php.net/win32/php4−win32−latest.zip .
- Une version non stable, téléchargeable sur http://qa.php.net/ .
- Vous pouvez obtenir PHP sur le serveur CVS anonyme .

Ces versions de PHP sont compatibles avec Apache 2.0.40 et plus récent.

| Note |
| --- |
| Le support des SAPI d'Apache 2.0 a commencé avec PHP 4.2.0. PHP 4.2.3 est connu pour fonctionner avec Apache 2.0.39. N'essayez pas d'utiliser cette version de PHP avec une autre version d'Apache 2.0. Nous ne recommandons pas l'utilisation de PHP 4.2.3 avec Apache 2.0.39.<br><br>Toutes les versions de PHP mentionnées ici fonctionnent avec Apache 1.3.x. |

### 3.4.11.2 Installer PHP et Apache 2 sur Linux

Téléchargez la version la plus récente de Apache 2.0 et une version appropriée de PHP , tel que décrit ci−dessus. Ce guide rapide couvre les manipulations de base, nécessaires à l'installation de Apache 2.0 et PHP . Pour plus d'informations, lisez la documentation Apache . Les numéros de version sont omis ici, pour s'assurer que les instructions ne soient pas incorrectes. Il faudra donc remplacer les séquences 'NN' avec les valeurs correctes que vous utilisez.

```
Instruction d'installation (Module partagé Apache 2)

1. gzip −d httpd−2_0_NN.tar.gz
2. tar xvf httpd−2_0_NN.tar
3. gunzip php−NN.tar.gz
4. tar −xvf php−NN.tar
5. cd httpd−2_0_NN
6. ./configure −−enable−so
7. make
8. make install

Maintenant, vous avez un dossier Apache 2.0.NN installe dans /usr/local/apache2,
configure avec le support des modules dynamiques, et le prefork standard MPM.
Pour tester l'installation, utilisez votre procedure standard de demarrage d'Apache :
/usr/local/apache2/bin/apachectl start
et pour stopper le serveur, utilisez
/usr/local/apache2/bin/apachectl stop.

9. cd ../php4−NN
10. ./configure −−with−apxs2=/usr/local/apache2/bin/apxs
11. make
12. make install
13. cp php.ini−dist /usr/local/lib/php.ini

Editez votre fichier php.ini pour configurer les options PHP.
Si vous preferez placer ce fichier a un autre endroit, utilisez l'option
−−with−config−file−path=/path en étape 10.

14. Editez votre fichier httpd.conf et verifiez la presence des lignes suivantes :

LoadModule php4_module modules/libphp4.so
AddType application/x−httpd−php .php

Vous pouvez installer toutes les extensions que vous souhaitez. .php est
simplement celle que nous suggerons.

Le chemin à droite de la directive LoadModule doit indiquer le chemin
du module PHP de votre systeme. La directive ci−dessus est correcte si vous
avez suivi les autres etapes a la lettre.

15. Utilisez votre procedure standard pour redemarrer Apache :
/usr/local/apache2/bin/apachectl start
```

Suivez les étapes ci−dessus, et vous disposerez d'un serveur Apache 2.0 avec le support de PHP module comme module SAPI. Bien sur, il y a bien d'autres options de configuration disponibles pour les deux logiciels, Apache et PHP . Pour plus de détails, utilisez la commande ./configure −−help dans le dossier de souces approprié. Si vous souhaitez compiler une version multi−threadé de Apache 2.0 vous devrez remplacer le module standard MPM prefork avec worker ou perchild . Pour ce faire, ajoutez à la ligne de configuration de l'étape 6, l'option −−with−mpm=worker ou −−with−mpm=perchild . Soyez conscient des conséquences, et comprenez bien ce que vous faites. Pour plus de détails sur ce sujet, lisez la documentation Apache sur le module MPM .

| Note |
| --- |
| Pour compiler une version multi−threadée d'Apache, votre système doit supporter les threads. Cela implique aussi de compiler PHP avec le module expérimental de Zend Thread Safety (ZTS). Par conséquent, toutes les extensions ne seront pas disponibles. La configuration recommandée actuellement est celle avec le module standard MPM prefork . |

### 3.4.11.3 Installation de PHP et Apache 2.0 sur Windows

Lisez donc les notes spécifiques à Windows pour Apache 2.0.

| Attention |
| --- |
| |

Apache 2.0 est conçu pour fonctionner sur Windows NT 4.0 et Windows 2000. Actuellement, le support des versions Windows 9x est incomplet. Apache 2.0 n'est pas prévu pour fonctionner sur ces plate−formes pour l'instant.

Téléchargez la version la plus récente de Apache 2.0 et une version de PHP appropriée, tel que mentionné au début de ce chapitre. Suivez les instructions d'installation manuelle puis revenez ici pour réaliser l'intégration de PHP et Apache.

Il y a deux méthodes pour que PHP fonctionne avec Apache 2.0 sous Windows. La première est l'interface CGI, et l'autre est le module DLL Apache. Dans les deux cas, commencez par stopper le serveur Apache, et éditez le fichier httpd.conf pour configurer Apache avec le support PHP .

Vous devez insérer les lignes suivantes dans le fichier httpd.conf pour réaliser la configuration en mode **CGI** :

Configurer la version de développement de PHP avec Apache 2.0 en CGI

```
 ScriptAlias /php/ "c:/php/"
AddType application/x−httpd−php .php
Action application/x−httpd−php "/php/php.exe"
```

Note

A partir de PHP 4.3.0, le nom de l'exécutable CGI a changé, et il est devenu php−cgi.exe . Prenez−le en compte, et utilisez la bonne directive pour Apache.

Si vous souhaitez utiliser PHP comme module dans Apache 2.0, placez bien le fichier php4ts.dll dans votre dossier winnt/system32 , en y écrasant tout autre fichier qui s'y trouverait. Puis vous devrez insérer les deux lignes suivantes dans votre fichier de configuration d'Apache httpd.conf pour y ajouter le module PHP d'Apache 2.0 :

Configurer PHP avec Apache 2.0 en module

```
 LoadModule php4_module c:/php/sapi/php4apache2.dll
AddType application/x−httpd−php .php
```

Note

N'oubliez pas de remplacer c:/php/ par votre chemin réel de PHP dans les exemples ci−dessus. Ne mélangez pas votre installation en module (DLL) avec **différentes versions de PHP** . Vous devez utiliser uniquement les DLL et extensions qui sont fournies avec la version de PHP que vous téléchargez.

## 3.4.12 CGI/ Installation pour exécution en ligne de commande

Par défaut, PHP est compilé en version CGI. Cela crée un interpréteur de commande qui peut être utilisé soit pour le traitement CGI, soit pour les scripts non relatifs au web. Si PHP peut être incorporé au serveur web que vous utilisez en tant que module, de manière générale c'est cette solution que vous devriez adopter pour des raisons de performances. Cependant, la version CGI permet aux utilisateurs sous Apache de lancer des scripts PHP sous leurs UID respectives. Lisez attentivement le chapitre consacré à la sécurité si vous souhaitez utiliser cette solution.

Avec PHP 4.3.0, d'importants ajouts ont été faits à PHP. une nouvelle SAPI, appelée CLI, existe aussi et porte le même nom que la version CGI. Ce qui est installé en tant que {PREFIX}/bin/php dépend de votre ligne de configuration. Tout ceci est décrit en détails dans la partie du manuel intitulée Utiliser PHP en ligne de commande . Veuillez vous y référer pour de plus amples informations.

#### 3.4.12.1 Tests

Si vous avez compilé PHP comme programme CGI, vous pouvez tester votre produit en tapant : make test . C'est toujours une bonne chose de tester le résultat d'une compilation. Cela vous permet de repérer des problèmes entre PHP et votre plate−forme, plutôt que d'attendre qu'ils surviennent.

#### 3.4.12.2 Performances

Si vous avez compilé PHP comme programme CGI, vous pouvez évaluer les performances de PHP 3 avec la commande make bench . Notez que si le safe mode est activé (par défaut), vous risquez de voir l'évaluation s'arrêter une fois les 30 secondes réglementaires écoulées. En effet, la fonction set_time_limit ne peut pas être utilisée si le safe mode fonctionne. Utilisez l'option max_execution_time pour contrôler le temps d'exécution de vos scripts. make bench ignore le fichier de configuration file .

| Note |
| --- |
| make bench n'est disponible qu'en PHP 3. |

#### 3.4.12.3 Utiliser les variables prédéfinies

Certaines variables d'environnement fournies par les serveurs web ne sont pas disponibles dans les spécifications CGI/1.1 actuelles. Seules les variables suivantes sont définies, et les autres doivent être considérées comme spécifiques aux serveurs web : AUTH_TYPE, CONTENT_LENGTH, CONTENT_TYPE, GATEWAY_INTERFACE, PATH_INFO, PATH_TRANSLATED, QUERY_STRING, REMOTE_ADDR, REMOTE_HOST, REMOTE_IDENT, REMOTE_USER, REQUEST_METHOD, SCRIPT_NAME, SERVER_NAME, SERVER_PORT, SERVER_PROTOCOL et SERVER_SOFTWARE

### 3.4.13 Installation avec les serveurs fhttpd

Pour compiler PHP comme un module fhttpd, répondre "yes" à la question "Build as an fhttpd module ?" (cela correspond à l'option de configuration −−with−fhttpd = DIR et spécifier la racine de la distribution fhttpd. Le répertoire par défaut est: /usr/local/src/fhttpd . Si vous utilisez fhttpd, compiler PHP en module vous permettra d'obtenir des performances supérieures, plus de contrôle et la possibilité d'exécution à distance.

### 3.4.14 Installation sur le serveur Caudium

PHP 4 peut être compilé comme module Pike pour le serveur web Caudium . Notez que ce mode n'est pas supporté en PHP 3. Suivez simplement les instructions suivantes pour installer PHP 4 sur un serveur Caudium.

| Instructions d'installation Caudium |
| --- |
| 1. Assurez−vous que vous avez un serveur Caudium installé avant de tenter l'installation PHP 4. Pour que PHP 4 fonctionne correctement, vous devez installer Pike 7.0.268 ou plus récent. Pour cet exemple, nous supposerons que vous avez installé Caudium dans le dossier /opt/caudium/server/.<br>2. Renommez le dossier en php−x.y.z (où x.y.z est le numéro de version).<br>3. ./configure −−with−caudium=/opt/caudium/server<br>4. make<br>5. make install<br>6. Redémarrez Caudium s'il était en fonctionnement<br>7. Connectez−vous à l'interface de configuration graphique et allez |

```
dans le serveur virtuel auquel vous voulez ajouter le support PHP.
8. Cliquez sur "Add Module" et recherchez puis ajoutez le module
"PHP 4 Script Support".
9. Si la documentation dit que 'PHP 4 interpreter isn't
available', assurez-vous que vous avez bien redémarré le serveur.
Si vous l'avez fait, vérifiez le fichier
/opt/caudium/logs/debug/default.1 : il contient peut-être des
erreurs liées à PHP4.so. De même, assurez-vous
que caudium/server/lib/[pike-version]/PHP4.so
est présent.
10. Configurez le module "PHP Script Support" si nécessaire.
```

Vous pouvez bien sûr compiler votre module Caudium avec les diverses extensions disponibles. Voyez la liste complète des options de configuration pour une liste exhaustive.

| Note |
| --- |
| Lorsque vous ajoutez le support MySQL à PHP 4, vous devez vous assurer que le client MySQL normal est utilisé. Sinon, il peut y avoir des conflits avec Pike, qui dispose déjà du support MySQL. Vous pouvez le faire en spécifiant le dossier d'installation de MySQL grâce à l'option --with-mysql . |

## 3.4.15 Installation avec les serveurs IIS/PWS

Cette section contient des notes sur l'installation de PHP avec IIS ( Microsoft Internet Information Server) : PWS/IIS 3 , PWS 4 ou plus récent et IIS 4 ou plus récent .

### 3.4.15.1 Windows et PWS/IIS 3

La méthode recommandée pour configurer ces serveurs est d'utiliser le fichier INF inclus dans la distribution ( php_iis_reg.inf ). Vous pouvez éditer ce fichier, pour vous assurer que les extensions et les dossiers d'installation de PHP sont bien ceux de votre configuration. Ou alors, vous pouvez suivre les instructions suivantes :

| Attention |
| --- |
| ATTENTION: Ces instructions requièrent la manipulation du fichier de registre de Windows. Une erreur peut laisser votre système dans un état instable. Nous vous recommandons vivement de sauvegarder ce fichier en lieu sûr. L'équipe de développement et les traducteurs de cette documentation ne pourront pas être tenus responsable d'un quelconque dommage qui pourrait survenir dans votre registre. |

- Lancez Regedit.
- Naviguez jusqu'à : HKEY_LOCAL_MACHINE /System /CurrentControlSet /Services /W3Svc /Parameters /ScriptMap .
- Dans le menu "edit", sélectionnez : New->String Value .
- Entrez l'extension que vous voulez utiliser pour les scripts PHP. Par exemple : .php
- Double-cliquez sur la chaîne, et entrez le chemin jusqu'à php.exe dans le champ "value data". ex: c:\php\php.exe .

Les étapes suivantes n'affectent pas la configuration du serveur web, et ne s'appliquent que si vous voulez que vos scripts PHP soient exécutés lorsqu'il sont exécutés en ligne de commande (par exemple, run c:\messcripts\test.php ) ou en double-cliquant sur l'icone. Vous pouvez ignorer ces étapes si vous préférez que vos scripts PHP s'ouvrent dans un éditeur de texte, plutôt que de les voir s'exécuter lorsque vous double-cliquez dessus.

- Répétez ces instructions pour toutes les extensions que vous voulez associer aux scripts PHP.
- Naviguez jusqu'à : HKEY_CLASSES_ROOT

- Dans le menu edit, sélectionnez: New−>Key .
- Donnez le nom de votre extension à la clé : ex: .php
- Sélectionnez le nom de la nouvelle clé dans le panneau de droite, et double−cliquez dans "default value", puis entrez phpfile .
- Répétez ces instructions pour toutes les extensions que vous avez associé aux scripts PHP.
- Créez une autre New−>Key sous HKEY_CLASSES_ROOT et nommez−la phpfile .
- Sélectionnez la nouvelle clé phpfile et dans le panneau de droite, double−cliquez dans "default value" et entrez PHP Script .
- Faites un clic droit dans phpfile et sélectionnez New−>Key , appelez−le Shell .
- Faites un clic droit dans Shell et sélectionnez New−>Key , appelez−le open .
- Faites un clic droit dans open et sélectionnez New−>Key , appelez−le command .
- Sélectionnez la nouvelle clé command et dans le panneau de droite, faites un double−clic dans "default value", puis entrez le chemin jusqu'à php.exe . ex: c:\php\php.exe −q %1 . (n'oubliez pas le %1 ).
- Quittez Regedit.
- Si vous utilisez PWS sous Windows, redémarrez pour prendre en compte le nouveau registre.

Les utilisateurs de PWS et IIS 3 sont prêts à utiliser leur serveur. Avec IIS 3, vous pouvez utiliser un outil bien pratique de Steven Genusa pour configurer votre carte des scripts.

#### 3.4.15.2 Windows et PWS 4 ou plus récent

Pour installer PHP sous Windows avec PWS 4 ou plus récent, vous avez deux options : l'une est d'avoir PHP sous forme de CGI, l'autre est d'utiliser les modules ISAPI, sous forme de DLL.

Si vous optez pour le CGI, faites ceci :

- Editez le fichier pws−php4cgi.reg (dans le dossier sapi ) pour indiquer la localisation de votre fichier php.exe . Les slash doivent être échappés. Par exemple : [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="C:\\PHP\\php.exe"
- Dans le gestionnaire PWS Manager, faites un clic droit sur les dossiers qui supporteront PHP, et sélectionnez "Properties". Cochez l'option "Execute" et confirmez.

Si vous optez pour les modules ISAPI, faites ceci :

- Editez le fichier pws−php4isapi.reg (dans le dossier sapi ) pour indiquer la localisation de votre fichier php4isapi.dll . Les slash doivent être échappés. Par exemple : [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\w3svc\parameters\Script Map] ".php"="C:\\PHP\\sapi\\php4isapi.dll"
- Dans le gestionnaire PWS Manager, faites un clic droit sur les dossiers qui supporteront PHP, et sélectionnez "Properties". Cochez l'option "Execute" et confirmez.

#### 3.4.15.3 Windows NT/2000 et IIS 4 ou plus récent

Pour installer PHP sous Windows NT/2000 server avec IIS 4 ou plus récent, vous avez deux options : l'une est d'avoir PHP sous forme de CGI, l'autre est d'utiliser les modules SAPI, sous forme de DLL.

Dans les deux cas, vous devez lancer la console "Microsoft Management" (elle peut aussi s'appeler "Internet Services Manager". Elle est située soit dans "Windows NT 4.0 Option Pack" ou dans

"Control Panel=>Administrative Tools" sous Windows 2000). Puis, faites un clic droit sur votre dossier web (qui apparaitra probablement comme Default Web Server ), et sélectionnez "Properties".

Si vous optez pour le CGI, faites ceci :

- Sous "Home Directory", "Virtual Directory", ou "Directory", cliquez sur le bouton "Configuration", et sélectionnez l'onglet "App Mappings".
- Cliquez sur "Add", puis dans la boîte "Executable", tapez : c:\php\php.exe (si vous avez installé PHP dans le dossier c:\php\ .)
- Dans la boîte "Extension", tapez le nom de l'extension que vous voulez associer aux scripts PHP. Laissez "Method exclusions" vide, et cochez "Script engine". Vous pouvez aussi préférer cocher la boîte 'check that file exists' − pour un coût faible, IIS (ou PWS) s'assurera que le script existe, et résoudra les problèmes d'autentification avant de démarrer PHP. Cela signifie que vous obtiendrez des messages d'erreur 404 intelligents, plutôt que d'avoir simplement une plainte du serveur comme quoi les CGI n'ont rien retourné.

  Répétez les étapes 3 et 4 pour chaque extension que vous souhaitez associer aux scripts PHP. ( .php et .phtml sont les plus répandues.)
- Configurer la sécurité nécessaire (dans "Internet Service Manager"), et si votre serveur NT utilise NTFS, ajoutez les droits adéquats pour I_USR_, au dossier qui contient php.exe .

Si vous optez pour les modules ISAPI, faites ceci :

- Si vous ne voulez pas effectuer des authentifications HTTP avec PHP, vous pouvez (et devez) sauter cette étape. Avec ISAPI Filters, ajoutez un nouveau filtre ISAPI. Utilisez PHP comme nom de filtre, et ajoutez simplement le chemin jusqu'à php4isapi.dll .
- Sous "Home Directory", cliquez sur le bouton "Configuration". Ajoutez une nouvelle entrée dans "Application Mappings". Utilisez le chemin jusqu'à php4isapi.dll comme "Executable", indiquez ".php" comme extension, laissez "Method exclusions" vide, et cochez "Script engine".
- Arrêtez totalement IIS (NET STOP iisadmin)
- Démarrez IIS (NET START w3svc)

## 3.4.16 Installation sous Netscape et iPlanet Enterprise Serveur

Cette section contient les notes et détails spécifiques à l'installation Netscape et iPlanet, aussi bien pour Sun Solaris que Windows .

Vous pouvez trouvez plus d'informations sur la configuration de PHP avec Netscape Enterprise Server : http://benoit.noss.free.fr/php/install−php4.html

### 3.4.16.1 Installer PHP avec Netscape sous Sun Solaris

Pour compiler PHP avec NES ou iPlanet, indiquez le bon dossier d'installation pour l'option de configuration −−with−nsapi = DIR option. Le dossier par défaut est généralement /opt/netscape/suitespot/ . Lisez aussi /php−xxx−version/sapi/nsapi/nsapi−readme.txt .

| Note |
| --- |
| Ces instructions, valables pour Sun Solaris 2.6 avec Netscape Enterprise Server 3.6, sont l'oeuvre de bhager at invacare dot com. |

1. Installez les packages suivants depuis le serveur http://www.sunfreeware.com/ ou un miroir ad hoc :
    - autoconf−2.13
    - automake−1.4
    - bison−1_25−sol26−sparc−local
    - flex−2_5_4a−sol26−sparc−local
    - gcc−2_95_2−sol26−sparc−local
    - gzip−1.2.4−sol26−sparc−local
    - m4−1_4−sol26−sparc−local
    - make−3_76_1−sol26−sparc−local
    - mysql−3.23.24−beta (si vous voulez le support MySQL)
    - perl−5_005_03−sol26−sparc−local
    - tar−1.13 (GNU tar)
2. Assurez−vous que le path inclut bien les dossiers nécessaires : PATH=.:/usr/local/bin:/usr/sbin:/usr/bin:/usr/ccs/bin et rendez le accessible à vortre système avec export PATH .
3. gunzip php−x.x.x.tar.gz (si vous avez une distribution .gz, ou bien allez en 4).
4. tar xvf php−x.x.x.tar
5. Passez dans votre dossier PHP : cd ../php−x.x.x
6. Pour les étapes suivantes, assurez−vous que /opt/netscape/suitespot/ correspond bien à votre installation du serveur netscape. Sinon, indiquez le chemin correct :

```
 ./configure −−with−mysql=/usr/local/mysql \
−−with−nsapi=/opt/netscape/suitespot/ \
−−enable−libgcc
```

7. Faites un make puis un make install .

Après avoir fait l'installation de base et lu les fichiers readme.txt, vous pouvez avoir besoin de faire des configurations supplémentaires.

D'abord, vous devez ajouter des chemins dans la variable LD_LIBRARY_PATH pour que PHP trouve toutes les librairies partagées. Le mieux est de le faire dans le script de démarrage de votre serveur Netscape. Les utilisateurs Windows peuvent probablement ignorer cette étape. Le script de démarrage est situé dans : /path/to/server/https−servername/start

Vous pouvez aussi avoir besoin d'éditer les fichiers de configuration qui sont situés dans : /path/to/server/https−servername/config/ .

Instructions de configuration pour Netscape Entreprise Server

Tout d'abord, vous aurez besoin d'ajouter des chemins dans la variable LD_LIBRARY_PATH pour que Netscape trouve son bonheur. Il est préférable de le faire dans le script de démarrage du serveur Netscape. Les utilisateurs Windows peuvent ignorer cette étape. Le script de démarrage est souvent situé dans : /path/to/server/https−servername/start . Vous aurez peut être à éditer le fichier de configuration situé dans /path/to/server/https−servername/config/ .

1. Ajoutez les lignes suivantes dans mime.types : type=magnus−internal/x−httpd−php exts=php
2. Ajoutez les lignes suivantes dans obj.conf . shlib peut varier en fonction de votre OS. Pour Unix, c'est quelquechose comme /opt/netscape/suitespot/bin/libphp4.so . Il est conseillé de placer les lignes suivantes après les lignes de mime types init .

```
 Init fn="load−modules" funcs="php4_init,php4_close,php4_execute,php4_auth_trans" shlib="/php4/nsapiPHP4.dll"
Init fn=php4_init errorString="Failed to initialize PHP!"
```

3. Configurez les valeurs par défaut :

```
 <object name="default">
```

```
.
.
.
.#NOTE this next line should happen after all 'ObjectType' and before all 'AddLog' lines
 Service fn="php4_execute" type="magnus-internal/x-httpd-php"
     .
     .
     </Object>
```

4. Créez un nouvel objet pour PHP :

```
 <Object name="x-httpd-php">
 ObjectType fn="force-type" type="magnus-internal/x-httpd-php"
 Service fn=php4_execute
 </Object>
```

5. Configuration d'authentification L'authentification PHP ne peut pas être utilisée avec d'autres authentifications. TOUTES LES FORMES D'AUTHENTIFICATION SONT PASSEES AU SCRIPT PHP. Pour configurer l'authentification PHP pour le serveur entier, ajoutez la ligne suivante :

```
 <Object name="default">
AuthTrans fn=php4_auth_trans
.
.
.
</Object>
```

6. Pour utiliser l'identification PHP sur un seul dossier, utilisez la ligne suivante :

```
 <Object ppath="d:\path\to\authenticated\dir\*">
 AuthTrans fn=php4_auth_trans
 </Object>
```

Si vous utilisez Netscape Enterprise 4.x, alors, il faut utiliser ceci : placez ces lignes dans les informations d'initialisation des types Mime, et tout ce qui y est rattaché dans la configuration ci-dessus : (fourni par Graeme Hoose, GraemeHoose at BrightStation dot com).

```
 Init fn="load-modules" shlib="/path/to/server4/bin/libphp4.so" funcs="php4_init,php4_close,php4_execute,php4_auth_trans"
Init fn="php4_init" LateInit="yes"
```

### 3.4.16.2 Installer PHP avec Netscape Enterprise sous Windows

Pour installer PHP sous forme de CGI (pour Netscape Enterprise Server, iPlanet, peut être Fastrack), suivez ces instructions :

- Copiez le fichier php4ts.dll dans votre dossier systemroot (le dossier où vous avez installé windows)
- Faîtes un fichier d'association depuis la ligne de commande. Tapez les lignes suivantes :

```
 assoc .php=PHPScript
ftype PHPScript=c:\php\php.exe %1 %*
```

- Dans le serveur Netscape Enterprise Administration Server, créez un dossier shellcgi et supprimez-le aussitôt (cette opération crée 5 lignes importantes dans le fichier obj.conf et permet au serveur de gérer les scripts CGI).
- Dans le serveur Netscape Enterprise Administration Server, créez un nouveau type MIME (Category: type, Content-Type: magnus-internal/shellcgi, File Suffix:php).
- Recommencez pour chaque instance de serveur web qui devra exécuter PHP.

Plus de détails sur la configuration de PHP comme CGI sont disponibles à http://benoit.noss.free.fr/php/install-php.html

Pour installer PHP avec l'interface NSAPI (pour Netscape Enterprise Server, iPlanet, peut-être Fastrack), faîtes ceci :

- Copiez le fichier php4ts.dll dans votre dossier systemroot (le dossier où vous avez installé windows)

- Faîtes un fichier d'association depuis la ligne de commande. Tapez les lignes suivantes :

```
 assoc .php=PHPScript
ftype PHPScript=c:\php\php.exe %1 %*
```

- Dans le serveur Netscape Enterprise Administration Server, créez un nouveau type MIME (Category: type, Content−Type: magnus−internal/shellcgi, File Suffix:php).
- Stoppez votre serveur web et éditez le fichier obj.conf . A la fin de la section Init, placez ces deux lignes (obligatoirement après l'init de type MIME) :

```
 Init fn="load−modules" funcs="php4_init,php4_close,php4_execute,php4_auth_trans" shlib="c:/php/sapi/php4nsapi.dll"
Init fn="php4_init" errorString="Failed to initialise PHP!"
```

- Dans la section < Object name="default" > , placez cette ligne, obligatoirement après toutes les lignes 'ObjectType' et avant les lignes 'AddLog' :

```
 Service fn="php4_execute" type="magnus−internal/x−httpd−php"
```

- A la fin du fichier, créez un nouvel objet appelé x−httpd−php , en insérant ces lignes :

```
 <Object name="x−httpd−php">
ObjectType fn="force−type" type="magnus−internal/x−httpd−php"
Service fn=php4_execute
</Object>
```

- Redémarrez votre serveur web, et validez les modifications
- Recommencez pour chaque serveur qui doit utiliser PHP.

Plus de détails sur la configiration de PHP en filtre NSAPI sont disponibles : http://benoit.noss.free.fr/php/install−php4.html

### 3.4.16.3 Utilisation particulière pour les pages d'erreurs ou les listages spécifiques de dossier (PHP >= 4.3.3)

Vous pouvez utilise PHP pour générer des pages d'erreurs de type "404 Not Found" ou apparantée. Ajoutez la ligne suivante dans le fichier obj.conf pour chaque page d'erreur que vous souhaitez remplacer :

```
 Error fn="php4_execute" code=XXX script="/path/to/script.php" [inikey=value inikey=value...]
```

où XXX est le code d'erreur HTTP. Effacez tout autre directive Error qui pourrait interférer avec la votre. Si vous voulez utiliser une page pour toutes les erreurs qui existent, laissez le paramètre code vide. Votre script peut obtenir le code de statut HTTP dans la variable $_SERVER['ERROR_TYPE'] .

Une autre possibilité est de générer une liste de dossier personnalisée. Créez simplement un script PHP qui affiche la liste du dossier, et remplacez la ligne Service par défaut par type="magnus−internal/directory" dans obj.conf avec ceci :

```
 Service fn="php4_execute" type="magnus−internal/directory" script="/path/to/script.php" [inikey=value inikey=value...]
```

Pour ces deux points, l'URI originale et l'URI traduite sont dans les variables $_SERVER['PATH_INFO'] et $_SERVER['PATH_TRANSLATED'] .

### 3.4.16.4 Note au sujet de nsapi_virtual et des requetes (PHP >= 4.3.3)

Le module NSAPI supporte désormais la fonction nsapi_virtual (alias : virtual ), pour réaliser des sous requêtes au serveur web, et inclure le résultat dans une page. Le problème est que cette fonction utilise une fonctionnalité non documentée de la librairie NSAPI.

Sous Unix, ce n'est pas un problème, car le module va automatiquement rechercher les fonctions nécessaires, et les utiliser si elles sont disponibles. Sinon, nsapi_virtual sera désactivée.

Sous Windows, des limitations dans la gestion des DLL impose l'utilisation de la plus récente librairie ns−httpdXX.dll . Cela a été testé pour les serveurs jusqu'à la version 6.0. Si une nouvelle version de SunONE server est utilisée, la détection échoue, et nsapi_virtual est désactivée.

Dans ce cas, essayez ceci : ajoutez le paramètre suivant à php4_init dans magnus.conf / obj.conf :

```
Init fn=php4_init ... server_lib="ns−httpdXX.dll"
```

où XX est le numéro correct de la version de la DLL. Pour la connaître, regardez dans le server−root pour connaitre le nom correct de la DLL. La DLL la plus grande en taille est la bonne.

Vous pouvez vérifier le status en utilisant la fonction phpinfo .

| Note |
| --- |
| Soyez prévenu : le support de nsapi_virtual est expérimental. |

## 3.4.17 Installation pour les serveurs OmniHTTPd

Cette section contient les notes et conseils pour installer PHP sur un serveur OmniHTTPd .

### 3.4.17.1 OmniHTTPd 2.0b1 et plus récent pour Windows

Vous devez réaliser les étapes suivantes pour configurer PHP sur votre serveur OmniHTTPd. C'est une configuration en exécutable CGI. SAPI est supporté par OmniHTTPd, mais des tests ont prouvé que ce n'est pas une solution stable.

- Step 1: Installez le serveur OmniHTTPd.
- Step 2: Faites un clic droit sur l'icone bleue d'OmniHTTPd sur le système, et sélectionnez Properties
- Step 3: Cliquez sur Web Server Global Settings
- Step 4: Dans l'onglet 'External', entrez : virtual = .php | actual = c:\path−to−php−dir\php.exe , et utilisez le bouton "Add" (ajout).
- Step 5: Dans l'onglet Mime , entrez : virtual = wwwserver/stdcgi | actual = .php , et utilisez le bouton "Add" (ajout).
- Step 6: Cliquez sur OK

Répetez les étapes de 2 à 6 pour chaque extension que vous voulez associer à PHP .

| Note |
| --- |
| Certains packages OmniHTTPd sont fournis avec le support de PHP . Vous pouvez le choisir au moment de l'installation, et décocher le composant PHP . Nous recommandons d'utiliser les dernières versions de PHP . Certains serveurs OmniHTTPd sont livrés avec des versions béta de PHP , et il vaut donc mieux éviter de les installer, et choisir une autre version, stable. Si le serveur est déjà sur votre machine, utilisez le bouton "Replace" (remplacer) dans les étapes 4 et 5 pour configurer un nouveau PHP . |

## 3.4.18 Installation Oreilly Website Pro Server

Cette section contient les conseils d'installation spécifiques à Oreilly Website Pro.

### 3.4.18.1 Oreilly Website Pro 2.5 et plus récent pour Windows

Cette liste décrit comment installer PHP comme CGI exécutable ou module ISAPI avec Oreilly Website Pro sous Windows.

- Editez les "Server Properties" et sélectionnez l'onglet "Mapping".
- Dans la "List" sélectionnez "Associations" et entrez le nom de l'extension voulue (".php") et le chemin jusqu'à l'exécutable (ex. c:\php\php.exe) ou la DLL ISAPI (ex. c:\php\sapi\php4isapi.dll).
- Sélectionnez "Content Types", ajoutez la même extension " .php " et entrez le "content type". Si vous choisissez la forme CGI, entrez "wwwserver/shellcgi"; si vous choisissez la forme module ISAPI, entrez "wwwserver/isapi" (sans les guillemets).

## 3.4.19 Serveurs Sambar

Cette section contient les notes et conseils d'installation de PHP sur les serveurs Sambar , sur Windows.

### 3.4.19.1 Installation pour Sambar sur Windows

Cette liste décrit comment configurer le module ISAPI avec le serveur Sambar sous Windows.

- Trouvez le fichier appelé mappings.ini (dans le dossier de configuration), dans le dossier d'installation de Sambar.
- Ouvrez mappings.ini et ajoutez la ligne suivante, sous la section [ISAPI] :

```
Configuration ISAPI de Sambar
 *.php = c:\php\php4isapi.dll
```

  (Cette ligne suppose que PHP a été installé dans le dossier c:\php .)
- Maintenant, redémarrez le serveur Sambar pour que les modifications prennent effet.

## 3.4.20 Installation Xitami

Cette section contient les conseils d'installation spécifiques à Xitami .

### 3.4.20.1 Xitami pour Windows

Cette liste décrit comment installer PHP comme CGI exécutable ou module ISAPI avec Xitami sous Windows.

- Assurez−vous que le serveur web fonctionne, et allez dans la console d'administration du serveur (généralement http://127.0.0.1/admin ), puis cliquez sur "Configuration".
- Naviguez dans les "Filters", et ajoutez l'extension que vous souhaitez (souvent ".php") dans le champ "File extensions".
- Dans la commande "Filter", ajoutez le nom et le chemin de votre exécutable PHP (souvent c:\php\php.exe ).

- Cliquez sur le bouton "Save".
- Redémarrez le serveur pour prendre en compte les modifications.

## 3.4.21 Autres serveurs web

PHP peut être compilé pour fonctionner avec de nombreux autres serveurs web. Reportez−vous à la partie Options des serveurs web pour une liste complète des options de configuration. Les exécutables PHP CGI sont compatibles avec la majorité des serveurs supportant les interfaces CGI.

## 3.4.22 Des problèmes?

Lisez la FAQ

Certains problèmes sont récurrents : les plus communs sont listés dans la FAQ PHP, disponible dans ce manuel .

### 3.4.22.1 Autres problèmes

Si vous êtes complètement bloqués, quelqu'un sur la liste de diffusion PHP pourra probablement vous aider. Essayez de consulter les archives, au cas où quelqu'un aurait déjà rencontré votre problème. Les archives sont toujours accessibles à : http://www.php.net/support.php . Pour souscrire à la liste de diffusion, envoyez un mail vide à php−install−subscribe@lists.php.net . L'adresse de la liste de diffusion est php−install@lists.php.net .

Si vous voulez obtenir de l'aide sur la liste de diffusion PHP, essayez d'être concis et clair, et pensez à donner tous les détails sur votre environnement (OS, version de PHP, serveur web, CGI ou module, safe mode , etc...), et n'hésitez pas à envoyer suffisamment de code pour que nous puissions reproduire l'erreur.

### 3.4.22.2 Rapports de Bug

Si vous pensez avoir trouvé un bug dans PHP, n'oubliez pas de le signaler. L'équipe de développement PHP ne le connaît peut être pas, et si vous ne le signaler pas, vos chances de voir le bug corrigé sont nulles. Vous pouvez rapporter des bugs grâce au système de suivi, accessible à http://bugs.php.net/ .

Lisez les guides (Les bugs : ce qu'il faut faire, et ce qu'il ne faut pas faire) avant d'envoyer n'importe quel rapport!

# 4 Référence

## 4.1 La syntaxe de base

### 4.1.1 Passer du HTML au PHP

Lorsque PHP commence à traiter un fichier, il ne fait qu'afficher le texte qu'il rencontre, jusqu'à ce qu'il rencontre une balise spéciale qui lui indiquera qu'il peut interpréter le code qui suit. L'analyseur PHP va alors exécuter ce code, jusqu'à ce qu'il rencontre une balise de fin de code PHP . A ce moment là, il retourne en mode texte, et il affiche simplement le contenu. C'est ce mécanisme qui vous permet d'inclure du code PHP dans des pages HTML : tout ce qui est placé hors des balises PHP est affiché sans modification, tandis que le contenu est exécuté.

Il y a quatre jeux de balises qui servent à délimiter des blocs de code PHP . Seules deux d'entre elles sont toujours actives : c'est <?php. . .?> et <script language="php">. . .</script> ). Tandis que les autres peuvent être activées et désactivées depuis le fichier de configuration php.ini . Les balises courtes et les balises ASP peuvent être pratiques, mais elles ne sont pas aussi portables que les versions longues. De plus, si vous envisagez d'inclure du PHP dans des fichiers XML ou XHTML, vous devrez utiliser la forme <?php. . .?> pour vous conformer au XML.

Le passage du HTML à PHP

```
 1.  <?php echo("Si vous voulez afficher du XML ou du XHTML, faites comme ceci.\n"); ?>

2.  <? echo ("Ceci est un exemple d'affichage à l'écran en PHP, sous forme d'expression SGML.\n"); ?>
    <?= expression ?>  Raccourci de "<? echo expression ?>"

3.  <script language="php">
        echo ("Certains éditeurs HTML (comme FrontPage)
         n'acceptent pas les expressions telles que celle-ci.");
    </script>

4.  <% echo ("Vous pouvez aussi utiliser le style ASP comme délimiteur."); %>
    <%= $variable; # ceci est un raccourci pour  "<%%echo .." %>
```

La première méthode est généralement utilisée, car elle permet une implémentation aisée de PHP avec la prochaine génération de XHTML.

La deuxième forme n'est valable que si vous l'avez activée. Soit en faisant appel à la fonction short_tags() (uniquement en PHP 3), soit en utilisant l'option d'exécution short_open_tag dans le fichier de configuration php.ini , soit encore en utilisant l'option de compilation −−enable−short−tags . Même si les balises courtes sont activées dans le fichier de configuration php.ini−dist , leur utilisation est déconseillée.

La quatrième forme est seulement disponible si vous l'avez activée en utilisant l'option d'exécution asp_tags .

Note

Le support des balises ASP a été ajouté en PHP 3.0.4.

Note

L'utilisation des balises courtes doit être évitées lors du développement d'applications et de librairies destinées à être distribuées ou déployées sur des serveurs qui sont hors de votre contrôle, car ces balises ne seront pas forcément supportées par le serveur de destination. Pour rendre vos scripts portables et faciles à distribuer, évitez les balises courtes.

La marque de fermeture d'un bloc (?>) comprend le caractère de nouvelle ligne qui le suit immédiatement, s'il y en a un. De plus, la balise fermante implique un point−virgule. Vous n'êtes pas obligés de mettre le point−virgule à la fin d'un bloc de PHP .

PHP vous permet d'utiliser des structures telles que :

Méthode avancée

```
 <?php
  if ( expression−booléenne ) {
?>
  <strong>Ceci est vrai.</strong>
<?php
  } else {
?>
<strong>Ceci est faux.</strong>
<?php
  }
?>
```

Cela fonctionne comme on peut s'y attendre, car PHP affiche le texte entre les balises ?> et <?php sans modification. L'exemple ci−dessus est très simple, mais c'est un bon système à garder en tête pour afficher de grandes quantités de texte : c'est même plus efficace que d'utiliser les fonctions echo ou print .

## 4.1.2 Le séparateur d'instructions

Les instructions sont séparées par un point virgule à chaque fin d'instruction, comme en langage C ou en Perl.

La balise de fin (?>) implique la fin d'une instruction, et donc ajoute implicitement un point virgule. Les deux exemples suivants sont équivalents.

Le séparateur d'instructions

```
 <?php
    echo "Ceci est un test";
?>
<?php echo "Ceci est un test" ?>
```

## 4.1.3 Commentaires

PHP supporte les commentaires comme en C, C++ et Shell Unix. Par exemple :

Commentaires en PHP

```
 <?php
    echo "Ceci est un test"; // Ceci est un commentaire sur une ligne comme en C++
    /* Ceci est un commentaire sur plusieurs lignes,
       comme en C et C++  */
    echo "Ceci est encore un test";
    echo "Enfin, le test final"; # Ceci est un commentaire comme en Shell Unix
?>
```

Commentaires en PHP (2)

```
 <h1>Ceci est un <?php # echo "simple";?> exemple.</h1>
<p>L'entête ci−dessus affiche 'Ceci est un exemple'.
```

Faites attention à ne pas emboîter les commentaires de type langage C, ce qui arrive de temps en temps lorsque vous voulez commenter une grande partie de code.

Commentaires emboîtés

```
 <?php
 /*
    echo "Ceci est un test"; /* Ce commentaire va poser un problème */
 */
?>
```

Les commentaires // commentent jusqu'à la fin de la ligne ou bien jusqu'à la fin du bloc (le premier des deux). Cela signifie que du code HTML placé après // ?> sera affiché, car PHP retourne alors en mode texte juste après la balise ?>.

# 4.2 Les types

## 4.2.1 Introduction

PHP supporte les huit types basiques suivants :

PHP supporte quatre types scalaires :

- booléen
- entier
- nombre à virgule flottante
- chaîne de caractères

PHP supporte deux types composés :

- tableau
- objet

PHP supporte deux types spéciaux :

- ressource
- null

| Note |
| --- |
| Dans ce manuel, vous rencontrerez souvent le type mixed . C'est un pseudo−type, qui indique que le paramètre peut indifféremment prendre plusieurs types. |

Habituellement, le type d'une variable n'est pas déclaré par le programmeur. Il est décidé au moment de l'exécution par le PHP , en fonction du contexte dans lequel la variable est utilisée.

| Note |
| --- |
| Si vous voulez vérifier le type et la valeur d'une variable ou d'une expression , utilisez la fonction var_dump .<br><br>Si vous souhaitez simplement une représentation lisible pour du débogage, utilisez la fonction gettype . Pour vérifier la présence de certains types, **n'utilisez pas** gettype , mais plutôt les fonctions is_ type . Par exemple : |

Exemples de vérification de type de variable

```
 <?php
$bool = TRUE;   // un booléen
$str  = "foo";  // une chaîne de caractères
$int  = 12;     // un entier

echo gettype($bool); // affiche "boolean"
echo gettype($str);  // affiche "string"

// Ceci est un engier, on l'incrémente de 4
if (is_int($int)) {
    $int += 4;
}

// Si $bool est une chaîne, l'afficher
// (Ne pas imprimer n'importe quoi).
if (is_string($bool)) {
    echo "Chaine : $bool";
```

```
}
?>
```

Si vous voulez forcer une variable à être convertie en un certain type, vous devez transtyper ( cast ) la variable ou utiliser la fonction settype .

Il est à noter qu'une variable peut se comporter de manière différente suivant les situations, en fonction du type qui lui est affecté. Pour plus d'informations, voir le paragraphe transtypage .

## 4.2.2 Booléens

C'est le type le plus simple. Un booléen exprime les valeurs de TRUE ou FALSE .

### 4.2.2.1 Syntaxe

Vous pouvez utiliser les constantes ' TRUE ' et ' FALSE ' pour spécifier une valeur de type bool . Ces constantes sont insensibles à la casse.

Utilisation des constantes TRUE et FALSE

```
<?php
 $foo = True;
  // assigne la valeur TRUE à la variable $foo
?>
```

Généralement, vous les créerez avec un des opérateurs qui retourne une valeur bool , pour le passer à une structure de contrôle .

Utilisation des constantes TRUE et FALSE avec une structure de contrôle

```
 // == est un opérateur d'égalité
// qui retourne un booléen
if ($action == "show_version") {
    echo "La version est la 1.23";
}

// Ceci n'est pas nécessaire
if ($show_separators == TRUE) {
    echo "<hr>\n";
}

// car vous pouvez simplement utiliser
if ($show_separators) {
    echo "<hr>\n";
}
```

### 4.2.2.2 Conversion en booléen

Reportez−vous au chapitre " Définition du type " pour plus d'informations sur les conversions.

Lors des conversions de valeurs de type bool , les valeurs suivantes sont considérées comme fausse ( FALSE ) :

- Le booléen FALSE
- L' entier 0 (zéro)
- Le nombre à virgule flottante 0.0 (zéro)
- La chaîne vide, et la chaîne "0"
- Le tableau vide (aucun élément)
- L' objet vide (aucun élément)
- La constante spéciale NULL

Toutes les autres valeurs sont considérées comme vraies ( TRUE (y compris les ressources ).

| Attention |
| --- |
| −1 est considéré comme vrai! |

Conversion en booléen

```
<?php
echo gettype((bool) "");        // bool(false)
echo gettype((bool) 1);         // bool(true)
echo gettype((bool) −2);        // bool(true)
echo gettype((bool) "foo");     // bool(true)
echo gettype((bool) 2.3e5);     // bool(true)
echo gettype((bool) array(12)); // bool(true)
echo gettype((bool) array());   // bool(false)
?>
```

## 4.2.3 Entiers

Un entier est un nombre de l'ensemble des entiers naturels Z : Z = {...., −2, −1, 0, 1, 2, ...} . Il est possible de spécifier les nombres entiers ( integer ) de toutes les manières suivantes : décimale (base 10), hexadécimale (base 16), octale (base 8) éventuellement précédé du signe moins ( − ).

### 4.2.3.1 Syntaxe

Les entiers peuvent être spécifiés en base décimale (dite aussi base 10), en hexadécimale (base 16) ou octale (base 8). Les entiers peuvent être optionnellement précédés par le signe plus ou moins (− or +).

Pour utiliser la notation octale, vous devez préfixer le nombre avec un zéro; pour utiliser la notation hexadécimale, vous devez préfixer le nombre avec 0x .

Notations entières

```
$a = 1234; // nombre entier en base 10
$a = −123; // nombre entier négatif
$a = 0123; // nombre entier en base 8, octale (équivalent à 83 en base 10)
$a = 0x1A; // nombre entier en base 16, hexadécimale
           // (équivalent à 26 en base 10)
```

Techniquement, la structure d'un entier suit les règles suivantes :

Structure d'un entier

```
decimal     : [1−9][0−9]*
            | 0

hexadecimal : 0[xX][0−9a−fA−F]+

octal       : 0[0−7]+

integer     : [+−]?decimal
            | [+−]?hexadecimal
            | [+−]?octal
```

La taille des entiers dépend de la plate−forme de support, mais la valeur maximale est généralement de 2 milliards et des poussières (c'est un entier signé de 32 bits). PHP ne supporte pas les entiers non signés.

### 4.2.3.2 Dépassement de capacité des entiers

Si un nombre est hors de l'intervalle de validité des entiers, il sera interprèté comme un float .

Dépassement de capacité des entiers

```
<?php
$large_number =  2147483647;
var_dump($large_number);
// affiche : int(2147483647)

$large_number =  2147483648;
```

```
var_dump($large_number);
// affiche : float(2147483648)

// this goes also for hexadecimal specified integers:
var_dump( 0x80000000 );
// affiche : float(2147483648)

$million = 1000000;
$large_number =  50000 * $million;
var_dump($large_number);
// affiche : float(50000000000)
?>
```

De même, si une fonction ou un opérateur retourne un entier qui est hors des limites de validité des entiers, il sera aussi automatiquement converti en float .

En PHP , il n'y a pas de division entière. 1/2 sera un nombre à virgule flottante ( float ) de valeur 0.5 .

| Attention |
| --- |
| Malheureusement, il y a un bug dans le moteur (corrigé en 4.1.0), qui fait que PHP ne fonctionne pas toujours bien lorsque des nombres négatifs sont utilisés. Lorsque les deux opérandes sont positifs, il n'y a pas de problèmes. Par exemple : −50000 * $million , conduit à −429496728 . |

Il n'existe pas de division entière en PHP . 1/2 générera le float 0.5 . Vous pouvez forcer la valeur au format entier pour faire l'arrondi, ou bien utiliser les fonctions round , floor ou ceil .

Divisions entières

```
 <?php
var_dump(25/7);
  // float(3.5714285714286)
 var_dump((int) (25/7));
   // int(3)
var_dump(round(25/7));
  // float(4)
 ?>
```

### 4.2.3.3 Conversion en entiers

Pour explicitement convertir une valeur en entier , utilisez les opérateurs de transtypage (int) ou (integer) . Cependant, dans la plupart des situations, vous n'en aurez pas besoin, car une valeur sera automatiquement convertie si un opérateur, une fonction ou tout autre élément du langage requiert un entier .

Reportez−vous à la section définition de type pour plus d'informations sur les conversions.

#### 4.2.3.3.1 Depuis un booléen

FALSE devient 0 (zéro), et TRUE devient 1 (un).

#### 4.2.3.3.2 Depuis un nombre à virgule flottante

Lors de conversion entre un nombre à virgule flottante et un entier, le nombre sera arrondi à la valeur inférieure s'il est positif, et supérieure s'il est négatif (conversion dite 'vers zéro').

Si le nombre est hors de l'intervalle de validité des entiers, (généralement +/− 2.15e+9 = 2^31 ), le résultat est indéfini, car les nombres à virgule flottante n'ont pas assez de précision pour fournir une valeur exacte pour un entier.

| Attention |
| --- |

Aucune alerte, même pas le plus petit message ne sera affiché dans ce cas.

Attention

Ne transformez jamais une fraction inconnue en entier, car cela peut conduire à des résultats irrationnels.

Attention aux erreurs de fractions

```
<?php
 echo (int) ( (0.1+0.7) * 10 );
// affiche 7!
?>
```

Pour plus d'informations, reportez−vous aux alertes liées aux nombres à virgule flottante.

#### 4.2.3.3.3 A partir de chaînes de caractères

Reportez−vous à la section des conversions de chaînes .

#### 4.2.3.3.4 Conversion d'autres types

La conversion d'autres types en entier est indéfinie. Actuellement, PHP convertit d'abord la valeur en booléen .

Attention

Mais, ne vous fiez pas à ce comportement, car il est susceptible de changer sans préavis!

Voir aussi : Nombres de grande taille et Nombres à virgules flottantes .

## 4.2.4 Les nombres à virgule flottante

Les nombres à virgule flottante (connus aussi sous le vocable de " double " ou " float " " nombre rééls ") peuvent être spécifiés en utilisant la syntaxe suivante :

Exemple de nombres à virgule flottante

```
<?php
 $a = 1.234;
 $a = 1.2e3;
?>
```

Attention

Précision des nombres à virgule flottante

Il est fréquent que de simples fractions décimales telles que 0.1 ou 0.7 ne puissent être converties au format interne binaire sans une légère perte de précision. Cela peut conduire à des résultats étonnants : par exemple, floor((0.1+0.7)*10) retournera 7 au lieu de 8 car le résultat de la représentation interne est 7.9999999999... .

Tout ceci est lié au fait qu'il est impossible d'exprimer certaines fractions en un nombre fini de chiffres. Par exemple 1/3 s'écrira 0.3333333... en mode décimal.

Ne faites donc jamais confiance aux nombres à virgule jusqu'à leur dernière décimale, et ne comparez jamais ces nombres avec l'opérateur d'égalité. Si vous avez besoin d'une précision particulière, reportez−vous au traitement des nombres de grande taille avec les librairies BC ou GMP .

#### 4.2.4.2 Conversion d'un nombre à virgule flottante

Pour savoir quand et comment des chaînes sont converties en nombre à virgule flottante, voyez la section Conversion de chaînes de caractères . Pour les valeurs d'autres types, la conversion est la même que si la valeur était d'abord convertie en entier, puis en nombre décimal. Voyez la section Conversion d'un entier .

## 4.2.5 Les chaînes de caractères

Les chaînes de caractères sont des séquences de caractères. En PHP , un caractère est un octet, et il y en 256 de possibles. PHP n'a pas (encore?) de support natif d'Unicode.

| Note |
| --- |
| La taille n'est pas un problème majeur pour une chaîne. Elle peut devenir très grande sans problème. Il n'y a pas à s'en faire pour cela. |

### 4.2.5.1 Syntaxe

Une chaîne peut être spécifiée de trois manières différentes :

- guillemets simples
- guillemets doubles
- syntaxe HereDoc

#### 4.2.5.1.1 Guillemets simples

Le moyen le plus simple de spécifier une chaîne de caractères est d'utiliser les guillemets simples : ' .

Pour spécifier un guillemet simple littéral, vous devez l'échapper avec un anti−slash ( \ ), comme dans de nombreux langages. Si un anti−slash doit apparaître dans votre chaîne ou bien en fin de chaîne, il faudra le doubler. Notez que si vous essayez d'échapper n'importe quel autre caractère, l'anti−slash sera conservé! Il n'y a pas besoin d'échapper d'autres caractères que le guillemet lui−même.

| Note |
| --- |
| En PHP 3, une alerte sera affichée si cela arrive avec un niveau de rapport d'erreur de E_NOTICE . |

| Note |
| --- |
| Contrairement aux autres syntaxes, les variables présentes dans la chaîne ne seront **PAS** remplacées par leurs valeurs. |

Exemples d'interpolation des chaînes de caractères

```
<?php
  echo 'Ceci est une chaîne simple';
  echo 'Vous pouvez inclure des nouvelles lignes dans une chaîne,
  comme ceci.';
  echo 'Arnold a coutume de dire : "I\'ll be back"';
// affiche : ... "I'll be back"
  echo 'Etes vous sûr de vouloir effacer le dossier C:\\*.*?';
// affiche : Etes vous sûr de vouloir effacer le dossier C:\*.*?
  echo 'Etes vous sûr de vouloir effacer le dossier C:\*.*?';
// affiche : Etes vous sûr de vouloir effacer le dossier C:\*.*?
  echo 'Je suis en train de mettre une nouvelle ligne comme ceci : \n';
// affiche : Je suis en train de mettre une nouvelle ligne comme ceci : \n
?>
```

#### 4.2.5.1.2 Guillemets doubles

Si la chaîne est entourée de guillemets doubles, PHP va comprendre certaines séquences de caractères :

| Séquence | Valeur |
|---|---|
| \n | Nouvelle ligne (linefeed, LF ou 0x0A (10) en ASCII) |
| \r | Retour à la ligne ( carriage return , CR ou 0x0D (13) en ASCII) |
| \t | Tabulation horizontale (HT ou 0x09 (9) en ASCII) |
| \\ | Anti−slash |
| \$ | Caractère $ |
| \" | Guillemets doubles |
| \[0−7]{1,3} | Une séquence de caractères qui permet de rechercher un nombre en notation octale. |
| \x[0−9A−Fa−f]{1,2} | Une séquence de caractères qui permet de rechercher un nombre en notation hexadécimale. |

Si vous essayez d'utiliser l'anti−slash sur n'importe quelle autre séquence, l'anti−slash sera affiché dans votre chaîne.

Le plus important pour les chaînes à guillemets doubles est le fait que les variables qui s'y trouvent seront remplacées par leur valeur. Voir la section sur le traitement des variables dans les chaînes pour plus de détails.

#### 4.2.5.1.3 Syntaxe Heredoc

Un autre moyen de délimiter les chaînes est d'utiliser la syntaxe dite " Heredoc " (en français, documentation ici): <<<, suivi d'un identifiant arbitraire, puis de la chaîne. Cette séquence se termine par l'identifiant initial, placé en premier sur une nouvelle ligne.

L'identifiant utilisé doit suivre les mêmes règles que les étiquettes PHP : il ne doit contenir uniquement que des caractères alpha−numériques, et des soulignés ("_"), et enfin, commencer par un caractère alphabétique ou un souligné.

| Attention |
|---|
| Il est très important de noter que la ligne qui contient l'identifiant de fermeture ne doit contenir aucun autre caractère, hormis, éventuellement, un point−virgule ; . Cela signifie notamment que l'identifiant ne doit pas être indenté, et qu'il n'y a aucun caractère blanc entre le retour à la ligne et l'identifiant, ou bien entre l'identifiant et le ; .<br><br>Le plus dur est peut être qu'il ne faut pas qu'il y ait un retour à la ligne (( \r ) à la fin de cette ligne, mais seulement un retour chariot ( \n ). Puisque Microsoft Windows utilise la séquence \r\n comme terminaison de ligne, la syntaxe heredoc risque de ne pas fonctionner là. Cependant, la plupart des éditeurs PHP fournissent une sauvegarde au format UNIX. |

La syntaxe Heredoc se comporte exactement comme une chaîne à guillemets doubles, sans les guillemets doubles. Cela signifie que vous n'avez pas à échapper les guillemets (simples ou doubles) dans cette syntaxe. Les variables sont remplacées par leur valeur, et le même soin doit leur être apporté que dans les chaînes à guillemets doubles.

Exemple de chaîne HereDoc

```
<?php
 $str = <<<EOD
 Exemple de chaîne
 s'étalant sur
 plusieurs lignes
 avec la syntaxe heredoc
EOD;
/* Exemple plus complexe, avec des variables. */
 class foo {
   var $foo;
   var $bar;
   function foo() {
```

```
        $this->foo = 'Foo';
        $this->bar = array('Bar1', 'Bar2', 'Bar3');
    }
  }
  $foo = new foo();
  $name = 'MonNom';
  echo <<<EOT
  Mon nom est "$name". J'affiche des $foo->foo.
  Maintenant, j'affiche un {$foo->bar[1]}.
  Ceci se traduit par un 'A' majuscule: \x41
EOT;
?>
```

| Note |
| --- |
| Le support Here doc a été ajouté en PHP 4. |

#### 4.2.5.1.4 Traitement des variables dans les chaînes

Lorsqu'une chaîne est spécifiée avec des guillemets doubles, ou en utilisant la syntaxe heredoc , les variables qu'elle contient sont remplacées par leur valeur.

Il y a deux types de syntaxe, une simple et une complexe . La syntaxe simple est la plus courante, et la plus pratique : elle fournit un moyen d'utiliser les variables, que ce soient des chaînes, des tableaux ou des membres d'objets.

La syntaxe complexe a été introduite en PHP 4 et peut être reconnue grâce aux accolades entourant les expressions.

##### 4.2.5.1.4.1 Syntaxe simple

Dès qu'un signe dollar $ est rencontré, l'analyseur PHP va lire autant de caractère qu'il peut pour former un nom de variable valide. Entourez le nom de la variable avec des accolades pour indiquer explicitement son nom.

Utilisation des accolades {} dans les chaînes

```
 <?php
  $boisson = 'vin';
  echo "Du $boisson, du pain et du fromage!";
 // OK, car "," n'est pas autorisé dans les noms de variables
  echo "Il a goûté plusieurs $boissons";
 // Pas OK, car 's' peut faire partie d'un nom de variable, et PHP recherchera alors $boissons
  echo "Il a goûté plusieurs ${boisson}s";
 // OK
?>
```

Similairement, vous pouvez aussi utiliser un élément de tableau, ou un membre d'objet. Pour les éléments de tableau, le crochet fermant ' ] ' marquera la fin du nom de la variable. Pour les membres d'objets, les mêmes règles que ci−dessus s'appliquent. Cependant, il n'existe pas de truc comme pour les variables simples.

Utilisation des accolades {} dans les chaînes pour les tableaux

```
 <?php
// Ces exemples sont spécifiques à l'utilisation de tableaux dans une chaîne.
// Lorsque vous êtes hors d'une chaîne, utilisez toujours des guillemets
// autour des index de tableau, et n'utilisez pas d'{accolades}.

// Affichons toutes les erreurs
error_reporting(E_ALL);

$fruits = array('fraise' => 'rouge', 'banane' => 'jaune');

// Fonctionne mais notez que cela fonctionne pas comme
 // si cela était hors d'une chaîne
echo "Une banane est $fruits[banane].";

// Fonctionne
echo "Une banane est {$fruits['banane']}.";

// Fonctionne mais PHP cherche une constante appelée banane
```

```
// tel que décrit ci-dessous
echo "Une banane est {$fruits[banane]}.";

// Ne fonctionne pas, il manque les accolades. Cela donne une erreur d'analyse
echo "Une banane est  $fruits['banane'].";

// Fonctionne
echo "Une banane est " . $fruits['banane'] . ".";

// Fonctionne
echo "Ce carré a un coté de $square->width mètres de large.";

// Ne fonctionne pas. Pour une solution, voyez la syntaxe complexe.
echo "Ce carré a un coté de $square->width00 centimètres.";
?>
```

Pour tout ce qui sera plus compliqué, voyez la syntaxe complexe.

#### 4.2.5.1.4.2 Syntaxe complexe

La syntaxe est dite "complexe" car elle permet l'utilisation d' expressions complexes, et non pas parcequ'elle serait obscure. Nuance.

En fait, vous pouvez inclure n'importe quelle valeur qui est dans l'espace de nom avec cette syntaxe. Il suffit d'écrire une expression exactement comme si elle était hors de la chaîne, puis de l'entourer d'accolades {} . Puisque vous ne pouvez pas échapper les accolades, cette syntaxe ne commence qu'à partir du signe dollar, qui suit immédiatement l'accolade ouvrante. Par exemple, vous pouvez utiliser "{\$" pour obtenir un "{$" littéral. Voici quelques exemples :

Syntaxe complexe de chaîne

```
 <?php
// Affichons toutes les erreurs
error_reporting(E_ALL);

$super = 'fantastique';

// Ne fonctionne pas. Affiche : Ceci est { fantastique}
echo "Ceci est { $super}";

// Fonctionne. Affiche Ceci est fantastique
echo "This is {$super}";
echo "This is ${super}";

// Fonctionne
echo "Ce carré a un coté de {$square->width}00 centimètres.";

 // Fonctionne
echo "Ceci fonctionne : {$arr[4][3]}";

// Ceci est faut pour la même raison que $foo[bar] est faux
// hors d'une chaîne. En d'autres terme, cela va fonctionne
// car PHP recherche d'abord une constante appelée foo, mais
// il générera une note E_NOTICE (undefined constant).
echo "Ceci est faux : {$arr[foo][3]}";

 // Fonctionne. Lorsque vous utilisez un tableau multidimensionnel dans
// une chaîne, n'oubliez jamais les accolades.
echo "Ceci fonctionne : {$arr['foo'][3]}";

// Fonctionne
echo "Ceci fonctionne : " . $arr['foo'][3];

echo "Vous pouvez même écrire {$obj->values[3]->name}";

echo "Ceci est une valeur de variable variable : {${$name}}";
?>
```

#### 4.2.5.1.5 Accès et modification des caractères d'une chaîne

Les caractères d'une chaîne sont accessibles et modifiables en spécifiant leur offset (le premier caractère est d'offset 0) entre accolade, après le nom de la variable.

Note

Pour assurer la compatibilité ascendante, vous pouvez toujours accéder aux caractères avec des crochets. Mais cette syntaxe est obsolète en PHP 4.

Exemples de chaînes

```
<?php
// Lit le premier caractère de la chaîne
$str = 'Ceci est un test.';
$first = $str{0};

// Lit le troisième caractère de la chaîne
$third = $str{2};

// Lit le dernier caractère de la chaîne
$str = 'Ceci est un test.';
$last = $str{strlen($str)-1};

// Modifie le dernier caractère de la chaîne
$str = 'Ceci est un test.';
$str{strlen($str)-1} = '!';

?>
```

### 4.2.5.2 Fonctions et opérateurs pratiques

Les chaînes peuvent être concaténées avec l'opérateur '.' (point). Notez que l'opérateur d'addition '+' (plus) ne fonctionnera pas. Reportez−vous à la section opérateurs de chaînes .

Il y a une grande quantité de fonctions pratiques pour modifier les chaînes.

Reportez−vous à la section chaînes de caractères pour les fonctions les plus générales, à Expressions régulières Perl et Expressions régulières POSIX étendues pour les recherches et remplacements.

Il y a aussi les fonctions sur les URL , ainsi que des fonctions de chiffrage ( mcrypt et mhash ).

Finalement, si vous ne trouvez toujours pas votre bonheur, il y a les fonctions de types de caractères .

### 4.2.5.3 Conversion en une chaîne de caractères

Vous pouvez convertir une valeur en une chaîne de caractères en utilisant l'opérateur (string) , ou bien la fonction strval . La conversion en chaîne de caractères est automatiquement faite dans toutes les situations qui nécessitent une chaîne de caractères. Cela arrive lorsque vous utilisez des fonctions comme echo ou print , ou encore lorsque vous comparez une valeur avec une chaîne de caractères.

Le booléen TRUE est converti en la chaîne de caractères "1" , et le booléen FALSE est représenté par la chaîne de caractères vide "" . De cette façon, vous pouvez convertir des booléens en chaîne de caractères et vice−versa.

Un entier ou un nombre à virgule flottante est converti en une chaîne de caractères qui représente ce nombre en décimal (y compris l'exposant des nombres à virgule flottante).

Les tableaux sont toujours convertis en la chaîne de caractères "Array" , ce qui fait que vous ne pouvez pas afficher le contenu d'un tableau avec la fonction echo ou print . Voyez les informations ci−dessous pour plus de conseils.

Les objets sont automatiquement convertis en la chaîne de caractères "Object" . Si vous souhaitez afficher le membre d'un objet, lisez le paragraphe ci−dessous. Si vous voulez connaître le nom de

la classe de cet objet, utilisez la fonction get_class .

Les ressources sont toujours converties en chaîne de caractères "Resource id #1" où 1 est le nombre unique représentant la ressource, assigné par PHP au moment de l'exécution. Si vous souhaitez connaître le type d'une ressource, utilisez get_resource_type .

NULL est toujours converti en une chaîne de caractères vide.

Comme vous pouvez le voir, afficher des tableaux, des objets ou des ressources n'est pas pertinent par cette méthode. Reportez−vous aux fonctions print_r et var_dump pour avoir des outils plus pratiques.

Vous pouvez aussi convertir des valeurs PHP en chaînes pour les sauver dans un stockage. Cette méthode est appelée la linéarisation, et peut être exploitée avec la fonction serialize . Vous pouvez aussi linéariser des valeurs PHP en structure XML si vous avez le support WDDX .

#### 4.2.5.4 Conversion de chaînes de caractères

Lorsqu'une chaîne de caractère est évaluée comme une valeur numérique, le résultat et le type de la variable sont déterminés comme suit.

La chaîne de caractères est de type "double" si elle contient un des caractères '.', 'e' ou 'E'. Sinon, elle est de type entier ( "integer" ).

La valeur est définie par la première partie de la chaîne. Si la chaîne de caractères débute par une valeur numérique cette valeur sera celle utilisée. Sinon, la valeur sera égale à 0 (zéro).

Lorsque la première expression est une chaîne de caractères, le type de la variable dépend de la seconde expression.

Exemples de conversions automatiques

```
<?php
 $foo = 1 + "10.5";              // $foo est du type  float (11.5)
 $foo = 1 + "−1.3e3";            // $foo est du type  float (−1299)
 $foo = 1 + "bob−1.3e3";         // $foo est du type  integer (1)
 $foo = 1 + "bob3";              // $foo est du type  integer (1)
 $foo = 1 + "10 petits cochons";     // $foo est du type  integer (11)
 $foo = 1 + "10 gros cochons"; // $foo est du type  integer (11)
 $foo = "10.0 cochons " + 1;        // $foo est du type  integer (11)
 $foo = "10.0 cochons " + 1.0;      // $foo est du type  float (11)
?>
```

Pour plus d'informations sur les conversions de type, voir les pages de man à propos de la fonction strtod(3) .

Si vous voulez tester l'un des exemples de cette section, vous pouvez faire un copier/coller et l'insérer dans un script pour voir comment il se comporte.

Exemples de conversions

```
<?php
 echo "\$foo==$foo; type is " . gettype( $foo ) . "<br>\n";
?>
```

## 4.2.6 Les tableaux

Un tableau PHP est en fait une association ordonnée (littéralement, une map ). Une association est un type qui fait correspondre des valeurs à des **clés** . Ce type est optimisé de diverses façons, qui font que vous pouvez le manipuler comme un tableau à indices réels, une liste (vecteur), ou un

table de hachage (qui est une implémentation d'association), dictionnaire, collection, pile, queue et encore d'autres. Comme une valeur peut elle−même être un tableau, vous pouvez simuler facilement un arbre.

Les détails d'implémentation de ces structures sont hors du champs de ce manuel, mais vous trouverez ici un exemple de toutes ces structures.

### 4.2.6.1 Syntaxe

#### 4.2.6.1.1 Créer un tableau array

Un tableau array peut être créé avec la fonction array . Cette fonction prend en argument des structures key => value , séparées par des virgules.

array( *Array* value , ... ) // key est soit une string soit un integer positif // value peut être n'importe quoi

Affectations de tableaux

```
<?php
$arr = array("foo" => "bar", 12 => true);

echo $arr["foo"]; // bar
echo $arr[12];    // 1
?>
```

Une clé key est soit un entier ou bien une chaîne. Si une clé est la représentation standard d'un entier positif, elle sera interprètée comme tel. (i.e. '8' sera interprèté comme 8 , tandis que '08' sera interprèté comme '08' ).

Une valeur peut être n'importe quoi.

Création d'un tableau

```
<?php
 array("untableau" => array(6 => 5, 13 => 9, "a" => 43));

echo $arr["untableau"][6];    // 5
echo $arr["untableau"][13];   // 9
echo $arr["untableau"]["a"];  // 43

?>
```

Si vous omettez une clé lors de la spécification d'un tableau, l'indice maximum + 1 sera utilisé comme clé par défaut. Si aucun indice numérique n'a été généré, ce sera 0. Si vous en spécifiez une qui a déjà été assignée, la nouvelle valeur écrasera la précédente.

Indexation automatique de tableau

```
<?php
// Ce tableau est identique à
 array(5 => 43, 32, 56, "b" => 12);

// Celui ci
array(5 => 43, 6 => 32, 7 => 56, "b" => 12);
?>
```

Utiliser TRUE comme clé revient à utiliser le type integer de valeur 1 . Utiliser FALSE comme clé revient à utiliser le type integer de valeur 0 . Utiliser NULL comme clé revient à utiliser le type string de valeur "" . Utiliser la chaîne vide comme index va créer cet index, et ce n'est pas la même chose que d'utiliser des crochets vides.

Il n'est pas possible d'utiliser des tableaux ou des objets comme clés. Ce faisant, vous généreriez une alerte : Illegal offset type .

#### 4.2.6.1.2 La syntaxe à crochets

Vous pouvez aussi modifier un tableau existant en lui assignant simplement des valeurs.

L'assignation de valeurs de tableau se fait en spécifiant la clé entre crochets. Si vous omettez la clé (" $tableau[] "), la valeur sera ajoutée à la fin du tableau. $arr[ key ] = value ; $arr[] = value ; // key est soit une chaîne, soit un entier // value peut être n'importe quoi Si $arr n'existe pas, il sera créé. Cela en fait une alternative pour créer un tableau. Pour modifier une valeur, assignez−lui une nouvelle valeur. Pour supprimer une valeur, utilisez la fonction unset .

Manipuler des éléments de tableau

```
 <?php
$arr = array(5 => 1, 12 => 2);

$arr[] = 56;    // Ceci revient à $arr[13] = 56;
                // à ce moment du script

$arr["x"] = 42; // Ceci ajoute un nouvel élément
                // avec l'index "x"

unset($arr[5]); // Ceci efface un élément du tableau

unset($arr);    // Ceci efface tout le tableau
?>
```

### 4.2.6.2 Fonctions pratiques

Il y a toute une panoplie de fonctions pratiques pour travailler avec les tableaux .

Note

La fonction unset permet d'effacer des index dans un tableau. Sachez bien que le tableau ne sera PAS réindexé. Si vous utilisez uniquement des index numériques (commençant à zéro, et incrémentés de un), vous pouvez effectuer la réindexation en utilisant array_values .

Manipuler des éléments de tableau

```
 <?php
$a = array(1 => 'un', 2 => 'deux', 3 => 'trois');
unset($a[2]);
/* Cela va produire un tableau qui aurait été
    $a = array(1 => 'un', 3 => 'trois');
   et non pas
   $a = array(1 => 'un', 2 =>'trois');
*/

$b = array_values($a);
// Maintenant b est le tableau array(1 => 'un', 2 =>'trois')
?>
```

L'élément de langage foreach est spécifiquement dédié aux tableaux : il permet de passer en revue simplement les valeurs d'un tableau.

### 4.2.6.3 A faire et à ne pas faire avec les tableaux

#### 4.2.6.3.1 Pourquoi est−ce que $foo[bar] est invalide?

Dans vos vieux scripts, vous pouvez avoir utilisé la syntaxe suivante :

Accès aux index de tableaux, ancienne méthode

```
 <?php
  $foo[bar] = 'ennemi';
  echo $foo[bar];
?>
```

Cela est mauvais, mais ca marche. Pourquoi est−ce mauvais? La raison est que PHP attend une constante entre crochets (bar) plutôt qu'une chaîne ('bar', notez les guillemets). Or, PHP pourrait définir dans le futur des constantes qui, malheureusement pour votre code, auront ce nom. Actuellement, cela fonctionne car la constante indéfinie est transformée en chaîne ayant la même valeur.

D'autres exemples :

Exemples de conflits de constantes indéfinies

```
<?php
// Affichons toutes les erreurs
error_reporting(E_ALL);

$arr = array('fruit' => 'pomme', 'légume' => 'carotte');

// Correct
print $arr['fruit'];  // pomme
print $arr['légume']; // carotte

// Incorrect.  Cela fonctionne mais PHP affiche une alerte car
// il y a utilisation d'une constante indéfinie appelée fruit
//
 // Notice: Use of undefined constant fruit − assumed 'fruit' in...
print $arr[fruit];    // pomme

// Définissons une constante pour illustrer le problème.
// Nous créons une constante appelée fruit, qui vaut 'légume'.
define('fruit','légume');

// Notez la différence
print $arr['fruit'];  // pomme
print $arr[fruit];    // carotte

// La suite est correcte si cela se passe dans une chaîne.
 // Les constantes ne sont pas remplacées par leur valeur dans une chaîne
print "Bonjour $arr[fruit]";      // Bonjour pomme

// A une exception, les accolades entourant un tableau dans une chaîne
// permettent d'utiliser les constantes
print "Bonjour {$arr[fruit]}";    // Bonjour carotte
print "Bonjour {$arr['fruit']}";  // Bonjour pomme

// Ceci ne fonctionnera pas, et donnera une erreur d'analyse
// Parse error: parse error, expecting T_STRING' or T_VARIABLE' or T_NUM_STRING'
// Cela s'applique aussi aux autoglobales dans les chaînes
print "Bonjour $arr['fruit']";
print "Bonjour $_GET['foo']";

// La concatenation est une alternative
print "Bonjour " . $arr['fruit']; // Bonjour pomme
?>
```

Lorsque vous activez l' error_reporting pour afficher les erreurs de niveau E_NOTICE (en utilisant le niveau E_ALL ) alors, vous verrez ces erreurs. Par défaut, error_reporting est trop faible pour les afficher.

Comme indiqué dans la section syntaxe , il doit y avoir une expression entre les crochets (' [ ' et ' ] '). Cela signifie que vous pouvez écrire ceci :

Utilisation d'une expression dans un index de tableau

```
<?php
echo $arr[somefunc($bar)];
?>
```

Ceci est un exemple d'utilisation de retour de fonction comme index de tableau. PHP reconnaît aussi les constantes, comme la famille des E_* .

Utilisation d'une constante dans un index de tableau

```
<?php
$error_descriptions[E_ERROR]   = "Une erreur fatale est survenue";
$error_descriptions[E_WARNING] = "PHP émet une alerte";
$error_descriptions[E_NOTICE]  = "Ceci est juste une alerte informationnelle";
?>
```

Notez que E_ERROR est aussi un identifiant valide, tout comme bar dans le premier exemple. Mais le dernier exemple est identique à ceci :

Utilisation d'une constante dans un index de tableau(2)

```
<?php
$error_descriptions[1] = "Une erreur fatale est survenue";
$error_descriptions[2] = "PHP émet une alerte";
$error_descriptions[8] = "Ceci est juste une alerte informationnelle";
?>
```

car E_ERROR vaut 1 , etc...

Comme nous l'avons déjà expliqué dans les exemples ci−dessus, $foo[bar] fonctionne mais est dangereux. Cela fonctionne car bar est identifié comme une constante. Mais, si la constante n'existe pas, PHP suppose que vous souhaitez utiliser bar littéralement, sous la forme "bar" , mais que vous avez oublié les guillemets.

Pourquoi est−ce aussi dangereux?

A l'avenir, le groupe PHP peut décider d'ajouter une constante ou un autre mot clé, ou bien vous souhaiterez aussi ajouter une constante dans votre application, et vous serez bloqué. Par exemple, vous ne pouvez pas utiliser des mots comme empty et default de cette manière, car il sont déjà réservés .

Note

Pour insister, à l'intérieur d'une chaîne de caractères ( string ) à guillemets doubles, il est correct de ne pas utiliser les guillemets simples dans les index, et "$foo[bar]" est valide. Voyez les exemples ci−dessus pour plus de détails sur la syntaxe des variables dans les chaînes .

### 4.2.6.4 Conversion en tableau

Pour les types scalaires ( integer , float , string , boolean et resource ), convertir une valeur en tableau retourne un tableau qui contient un élément (à l'index 0) qui est la valeur originale.

Si vous convertissez un tableau en objet, vous obtiendrez des propriétés ou bien cet objet comme éléments de tableau. Les clés créées sont les noms des membres.

Si vous convertissez la valeur NULL en tableau, vous obtiendrez un tableau vide.

### 4.2.6.5 Exemples

Le type tableau de PHP est très souple. Voici quelques exemples d'utilisation :

Utilisation des tableaux

```
<?php
// ceci
  $a = array( 'couleur' => 'rouge'
            , 'gout' => 'sucre'
            , 'forme' => 'rond'
            , 'nom'  => 'pomme'
            ,            4        // cette clé sera 0
          );
// est complétement équivalent à
  $a['couleur'] = 'rouge';
  $a['gout'] =    'sucre';
  $a['forme'] =   'rond';
  $a['nom'] =     'pomme';
  $a[]        = 4;        // cette clé sera 0
  $b[] = 'a';
  $b[] = 'b';
  $b[] = 'c';
// va créer le tableau array( 0 => 'a' , 1 => 'b' , 2 => 'c' )
// ou plus simplement array('a' , 'b' , 'c' )
?>
```

Utilisation de array

```
<?php
// Array comme correspondance
```

```
  $map = array( 'version'    => 4
              , 'OS'         => 'Linux'
              , 'langue'       => 'français'
              , 'short_tags' => TRUE             );
// valeur strictement numériques
  $array = array( 7
                , 8
                , 0
                , 156
                , -10
                );
// ceci est la même chose que array( 0 => 7, 1 => 8, ...)
  $switching = array(         10 // clé = 0
                    , 5    =>  6
                    , 3    =>  7
                    , 'a'  =>  4
                    ,         11 // clé = 6 (index maximum : 5)
                    , '8'  =>  2 // clé = 8 (entier!)
                    , '02' => 77 // clé = '02'
                    , 0    => 12 // la valeur de la clé 10 sera remplacée par 12
                    );
// empty array
$empty = array();
?>
```

Collection

```
 <?php
  $couleurs = array('rouge','bleu','vert','jaune');
  foreach ( $couleurs as $couleur ){
    echo "Aimez-vous la couleur $couleur?\n";
  }
/* Affiche :
Aimez-vous la couleur rouge?
Aimez-vous la couleur bleu?
Aimez-vous la couleur vert?
Aimez-vous la couleur jaune?
*/
?>
```

Notez qu'il n'est pas possible actuellement de modifier les valeurs d'un tableau avec une telle boucle. Une solution pour cela est :

Collection

```
 <?php
  foreach( $couleurs as $cle => $couleur ){
// ne marche pas
//$couleur = strtoupper($couleur);
//marche :
    $couleurs[$cle] = strtoupper($couleur);
  }
  print_r($couleur);
/* Affiche :
Array
(
    [0] => ROUGE
    [1] => BLEU
    [2] => VERT
    [3] => JAUNE
)
*/
?>
```

Cet exemple crée un tableau d'index minimal 1.

Tableau en 1

```
 <?php
  $firstquarter  = array(1 => 'Janvier', 'Février', 'Mars');
  print_r($firstquarter);
/* Affiche:
Array
(
    [1] => 'Janvier'
    [2] => 'Février'
    [3] => 'Mars'
)
*/
?>
```

Remplissage d'un tableau

```
 <?php
// remplit un tableau avec les noms de fichiers d'un dossier
  $handle = opendir('.');
  while ( $file = readdir($handle) ){
    $files[] = $file;
```

```
  }
  closedir($handle);
?>
```

Les tableaux sont ordonnés. Vous pouvez modifier l'ordre des valeurs avec de nombreuses fonctions de classement. Voyez les fonctions de tableaux .

Tri de tableaux

```
 <?php
  sort($files);
  print_r($files);
?>
```

Comme une valeur de tableau peut être n'importe quoi, elle peut aussi être un autre tableau. Comme cela, vous pouvez avoir des tableaux multi−dimensionnels, ou récursifs.

Tableaux multi−dimensionnels, et récursifs

```
 <?php
$fruits = array ( "fruits"  => array ( "a" => "orange",
                                       "b" => "banane",
                                       "c" => "pomme"
                                     ),
                  "nombre" => array ( 1,
                                       2,
                                       3,
                                       4,
                                       5,
                                       6,
                                     ),
                  "trou"   => array (      "premier",
                                       5 => "second",
                                            "troisième"
                                     )
                );

// Exemples d'utilisations des tableaux ci−dessus
 echo $fruits["trou"][5];    // affiche "second"
echo $fruits["fruits"]["a"]; // affiche "orange"
unset($fruits["trou"][0]);  // supprime "premier"

// Créer un tableau multidimensionnel
$juices["pomme"]["vert"] = "bon";
 ?>
```

## 4.2.7 Les objets

### 4.2.7.1 Initialisation d'un objet

Pour initialiser un objet, vous devez utiliser la commande "new" afin de créer l'instance de l'objet.

Création d'un objet

```
 <?php
class foo
{
    function faire_foo()
    {
        echo "Faisant foo.";
     }
}

$bar = new foo;
$bar−>faire_foo();
?>
```

### 4.2.7.2 Conversion en objet

Si un objet est converti en objet, il n'est pas modifié. Si une valeur d'un autre type est convertie en objet, une nouvelle instance de la classe stdClass sera créée. Si la valeur était NULL la nouvelle instance sera vide. Pour les autres valeurs, un membre de l'objet appelé scalar contiendra la valeur.

Transtypage

```
<?php
 $obj = (object) 'au revoir';
 echo $obj->scalar;  // affiche 'au revoir'
?>
```

## 4.2.8 Ressources

Une ressource ( "resource" en anglais), est un type spécial, qui représente une référence sur une ressource externe. Les ressources sont créées par des fonctions dédiées. Reportez vous à l'annexe Types des ressources PHP pour une liste exhaustive des fonctions créant et utilisant ces ressources.

Note

Le type de données "resource" a été introduit en PHP 4.

### 4.2.8.1 Conversion en ressource

Comme les ressources contiennent des valeurs très spéciales, comme des fichiers ouverts, des connexions aux bases de données, des images ou n'importe quoi d'autre, vous ne pouvez pas convertir une valeur en ressource.

### 4.2.8.2 Libérer des ressources

Grâce au système de comptabilisation des références introduit en PHP 4 (avec le moteur Zend), PHP détecte automatiquement qu'une ressource n'est plus utilisée (comme Java). Dans ce cas, toutes les ressources systèmes utilisées par cette ressource sont libérées automatiquement.

Note

Les connexions persistantes représentent un cas particulier, elles ne seront **PAS** détruites. Voyez connexions persistantes .

## 4.2.9 La valeur NULL

La valeur spéciale NULL représente l'absence de valeur. Une variable avec la valeur NULL n'a pas de valeur.

### 4.2.9.1 Syntaxe

Il y a seulement une valeur de type NULL , et c'est la constante NULL , insensible à la casse.

La valeur NULL

```
<?php
 $var = Null;
?>
```

Note

La valeur NULL a été introduite en PHP 4.

## 4.2.10 Pseudo−types utilisés dans cette documentation

### 4.2.10.1 mixed

mixed indique qu'un paramètre accepte plusieurs types, mais pas forcément tous les types.

gettype par exemple, accepte tous les types de variables PHP , mais str_replace n'accepte que des chaîne de caractères et des tableau .

### 4.2.10.2 number

number indique qu'un paramètre peut être du type entier ou nombre à virgule flottante .

### 4.2.10.3 callback

Certaines fonctions comme par exemple call_user_function ou usort accepte une fonction définie par l'utilisateur comme fonction de callback. Les fonctions de callback peuvent être des fonctions simples, mais aussi des objets ou des méthodes statiques de classe.

Une fonction PHP est passée simplement en indiquant son nom, sous forme de chaîne de caractères. Vous pouvez passer une fonction intégrée au langage, ou une fonction définie par vos soins, à l'exception de array , echo , empty , eval , exit , isset , list , print et unset .

Une méthode d'un objet instancié est passée sous forme de tableau , contenant un objet comme élément d'index 0, et un nom de méthode comme élément 1.

Les méthodes statiques de classe peuvent aussi être passées dans l'objet instancié, mais en passant le nom de la classe comme valeur de l'index 0.

Exemple de fonction de callback

```php
<?php

// simple callback example
function foobar() {
    echo "hello world!";
}
call_user_function("foobar");

// method callback examples
class foo {
  function bar() {
    echo "hello world!";
  }
}

$foo = new foo;

call_user_function(array($foo, "bar")); // object method call

call_user_function(array("foo", "bar")); // static class method call

?>
```

## 4.2.11 Définition du type

PHP ne nécessite pas de déclaration explicite du type d'une variable. Le type d'une variable est déterminé par le contexte d'utilisation. Par exemple, si vous assignez une chaîne de caractères à la variable var , var devient une chaîne de caractère. Si vous assignez un nombre entier à var , elle devient un entier.

Un exemple de convertisseur automatique de type est l'opérateur '+'. Si un des opérandes est de type double, alors tous les opérandes sont évalués comme des variables de type double et le résultat est de type double. Sinon, tous les opérandes sont évalués comme des variables de type entier et le résultat sera du type entier. Il est à noter que cela NE CHANGE PAS le type des opérandes. Le seul changement est la manière dont les opérandes sont évaluées.

Conversion automatique

```
<?php
 $foo = "0";  // $foo est une chaîne de caractères (ASCII 48)
 $foo += 2;   // $foo est maintenant du type entier (2)
 $foo = $foo + 1.3;  // $foo est maintenant du type double (3.3)
 $foo = 5 + "10 Petits cochons"; // $foo est du type entier (15)
 $foo = 5 + "10 cochonnets";     // $foo est du type entier (15)
?>
```

Si les deux derniers exemples vous semblent obscurs ou si vous voulez forcer une variable a être évaluée avec un certain type, reportez−vous au paragraphe Conversion de type ci−dessous. Si vous voulez changer le type d'une variable, intéressez−vous à aux fonctions de conversion de chaînes .

Si vous voulez forcer le type d'une variable, vous pouvez vous reporter à la section transtypage . Si vous voulez changer le type d'une variable, utilisez settype .

Pour tester les exemples de cette section, il suffit d'en faire un copier/coller, et d'insérer les lignes dans un script PHP .

Exploitation des exemples

```
<?php
 echo "\$foo==$foo; le type est " . gettype( $foo ) . "<br>\n";
?>
```

Note

Le comportement de la conversion automatique est actuellement indéfinie.

Conversion automatique indéfinie

```
<?php
$a = "1";     // $a est une chaîne
$a[0] = "f";  // Mais que sont les index de chaîne?
?>
```

Etant donné que PHP supporte l'indexation de chaîne avec des offsets identiques à celles des tableaux, l'exemple ci−dessus conduit à un problème : est ce que $a est un tableau, dont le premier élément est " f ", ou bien est ce que "f" est le premier élément de la chaîne de caractères $a ?

Les versions courantes de PHP interprète la seconde assignation comme un index de chaîne, et donc $a devient " f ", et le résultat de cette conversion automatique doit être considéré comme indéfini. PHP 4 a introduit la syntaxe avec les accolades pour accéder aux caractères d'une chaîne. Utilisez plutôt le code ci−dessous :

Conversion automatique indéfinie(2)

```
<?php
$a    = "abc"; // $a est une chaîne
$a{1} = "f";   // $a vaut maintenant "afc"
?>
```

Voyez la section Accès aux caractères d'une chaîne pour plus de détails.

#### 4.2.11.1 Transtypage

La conversion de type en PHP fonctionne de la même manière qu'en C: le nom du type désiré est écrit entre parenthèses devant la variable à transtyper ("cast").

Transtypage

```
<?php
 $foo = 10;   // $foo est un entier
 $bar = (double) $foo;   // $bar est un double
?>
```

Les conversions autorisées sont:

- (int) , (integer) − type entier
- (bool) , (boolean) − booléen
- (real) , (double) , (float) − type double
- (string) − type chaîne
- (array) − type tableau
- (object) − type objet

Il est à noter que les tabulations et les espaces sont autorisés à l'intérieur des parenthèses, donc les lignes suivantes sont équivalentes:

Transtypage et espaces

```
<?php
 $foo = (int) $bar;
 $foo = ( int ) $bar;
?>
```

Note

Au lieu de transtyper une variable en chaîne, vous pouvez aussi l'insérer entre deux guillemets doubles :

Transtypage en chaîne

```
<?php
$foo = 10;            // $foo est un entier
$str = "$foo";        // $str est une chaîne
$fst = (string) $foo; // $fst est aussi une chaîne

// Ceci affiche : "Identique"
if ($fst === $str) {
    echo "Identique";
}
?>
```

Le transtypage n'a pas toujours de résultat prévisible. Pour plus d'informations, voyez :

- Conversion en booléen
- Conversion en entier
- Conversion en nombre à virgule flottante
- Conversion en chaîne de caractères
- Conversion en tableau
- Conversion en objet
- Conversion en ressource

## 4.3 Les variables

## 4.3.1 Essentiel

En PHP , les variables sont représentées par un signe dollar "$" suivi du nom de la variable. Le nom est sensible à la casse (i.e. $x != $X ).

Les noms de variables suivent les mêmes règles de nommage que les autres entités PHP . Un nom de variable valide doit commencer par une lettre ou un souligné (_), suivi de lettres, chiffres ou soulignés. Exprimé sous la forme d'une expression régulière, cela donne : ' [a−zA−Z_\x7f−\xff][a−zA−Z0−9_\x7f−\xff]* '

Note

Dans nos propos, une lettre peut être une des lettres minuscules (a à z) ou majuscules (A à Z), et les caractères ASCII de 127 à 255 ( 0x7f−0xff ).

Validité des noms de variables

```
<?php
 $var = "Jean";
 $Var = "Paul";
 echo "$var, $Var";         // affiche "Jean, Paul"
 $4site = 'pas encore';     // invalide : commence par un nombre
 $_4site = 'pas encore';    // valide : commence par un souligné
 $ma&#239;s = 'jaune';      // valide; 'Ô' est ASCII 239.
?>
```

En PHP 3, les variables sont toujours assignées par valeur. C'est−à−dire, lorsque vous assignez une expression à une variable, la valeur de l'expression est recopiée dans la variable. Cela signifie, par exemple, qu'après avoir assigné la valeur d'une variable à une autre, modifier l'une des variables n'aura pas d'effet sur l'autre. Pour plus de détails sur ce genre d'assignation, reportez−vous à Expressions .

PHP 4 permet aussi d'assigner les valeurs aux variables **par référence** . Cela signifie que la nouvelle variable ne fait que référencer (en d'autres terme, "devient un alias de", ou encore "pointe sur") la variable originale. Les modifications de la nouvelle variable affecteront l'ancienne, et vice versa. Cela signifie aussi qu'aucune copie n'est faite : l'assignation est donc beaucoup plus rapide. Cela se fera notamment sentir dans des boucles, ou lors d'assignation de grands objets (tableaux).

Pour assigner par référence, ajoutez simplement un & (ET commercial) au début de la variable qui est assignée (la variable source). Dans l'exemple suivant, Mon nom est Pierre s'affichera deux fois :

Assignation de référence

```
<?php
$foo = 'Pierre';              // Assigne la valeur 'Pierre' à $foo
$bar = &$foo;                 // Référence $foo avec $bar.
$bar = "Mon nom est Pierre";  // Modifie $bar...
echo $foo;                    // $foo est aussi modifiée
echo $bar;
?>
```

Une chose importante à noter est que seules les variables nommées peuvent être assignées par référence.

Assignation de référence et variables anonymes

```
<?php
$foo = 25;
$bar = &$foo;      // assignation valide .
$bar = &(24 * 7);  // assignation invalide : référence une expression sans nom
function test() {
   return 25;
}
$bar = &test();    // assignation invalide.
?>
```

PHP suit les conventions de Perl pour la gestion des opérateurs arithmétiques, et non pas celles du langage C. Par exemple, en Perl 'Z'+1 devient 'AA' , tandis que en langage C 'Z'+1 devient '[' (

ord('Z') == 90, ord('[') == 91 ).

Opérations arithmétiques et caractères

```
 <?php
$i = 'W';
for($n=0; $n<6; $n++)
  echo ++$i . "\n";

/*
  Affiche ceci :

 X
Y
Z
AA
AB
AC

*/
?>
```

## 4.3.2 Variables prédéfinies

PHP fourni un grand nombre de variables prédéfinies. Cependant, beaucoup de ces variables ne peuvent pas être présentées ici, car elles dépendent du serveur sur lequel elles tournent, de la version du serveur, et de la configuration du serveur, ou encore d'autres facteurs. Certaines de ces variables ne seront pas accessibles lorsque PHP fonctionne en exécutable.

Attention

Depuis la version PHP 4.2.0, la valeur par défaut de la directive PHP register_globals est **off** . Ceci est une évolution majeure de PHP . Avoir la directive register_globals à **off** affecte les variables prédéfinies du contexte globale. Par exemple, pour lire DOCUMENT_ROOT vous devez utiliser $_SERVER['DOCUMENT_ROOT'] au lieu de $DOCUMENT_ROOT , ou bien il faut lire $_GET['id'] dans l'URL http://www.example.com/test.php?id=3 au lieu de $id , ou encore $_ENV['HOME'] au lieu de $HOME .

Pour des informations liées à cette évolution, lisez la documentation de la directive register_globals , le chapitre sur la sécurité, à propos de l' Utilisation des variables superglobales , ainsi que les annonces de PHP 4.1.0 et 4.2.0 .

L'utilisation des variables prédéfinies de PHP , comme les tableaux superglobaux , est recommandé.

Depuis la version 4.1.0, PHP fournit un jeu de tableaux prédéfinis, contenant les variables du serveur (si possible), les variables d'environnement et celle d'entrées. Ces nouveaux tableaux sont un peut particuliers, car ils ont automatiquement globaux : ils sont automatiquement disponibles dans tous les environnements d'exécution, sans avoir à utiliser le mot réservé global . Pour cette raison, ils sont dits 'auto−globaux' ou bien encore 'superglobaux' (il n'y a pas de mécanisme PHP pour créer de telles variables. Les superglobales sont listées ci−dessous. Cependant, pour connaître le détails de leur contenu, et une présentation approfondie sur les variables prédéfinies PHP , et leur nature, reportez vous à la section variables prédéfinies . De plus, vous noterez que les anciennes variables prédéfinies ( $HTTP_*_VARS ) existent toujours. Depuis PHP 5.0.0, les tableaux prédéfinis PHP peuvent être désactivés avec l'option de configuration register_long_arrays .

Note

Variables variables

Les superglobales ne peuvent pas être utilisées comme variable variables .

Si certaines variables de variables_order ne sont pas définies, leur tableau prédéfini PHP correspondant est laissé vide.

Tableaux superglobaux de PHP

*$GLOBALS*
: Contient une référence sur chaque variable qui est actuellement disponible dans l'environnement d'exécution global. Les clés de ce tableau sont les noms des variables globales.

*$_SERVER*
: Les variables fournies par le serveur web, ou bien directement liées à l'environnement d'exécution du script courant. C'est la nouvelle version de l'ancienne variable $HTTP_SERVER_VARS , qui est maintenant obsolète, mais toujours là.

*$_GET*
: Les variables fournies par le protocole HTTP en méthode GET. C'est la nouvelle version de l'ancienne variable $HTTP_GET_VARS qui est maintenant obsolète, mais toujours là.

*$_POST*
: Les variables fournies par le protocole HTTP en méthode POST. C'est la nouvelle version de l'ancienne variable $HTTP_POST_VARS qui est maintenant obsolète, mais toujours là.

*$_COOKIE*
: Les variables fournies par le protocole HTTP, dans les cookies. C'est la nouvelle version de l'ancienne variable $HTTP_COOKIE_VARS qui est maintenant obsolète, mais toujours là.

*$_FILES*
: Les variables fournies par le protocole HTTP, suite à un téléchargement de fichier. C'est la nouvelle version de l'ancienne variable $HTTP_POST_FILES qui est maintenant obsolète, mais toujours là. Voir Téléchargement par méthode POST , pour plus d'informations.

*$_ENV*
: Les variables fournies par l'environnement. C'est la nouvelle version de l'ancienne variable $HTTP_ENV_VARS qui est maintenant obsolète, mais toujours là.

*$_REQUEST*
: Les variables fournies au script par n'importe quel mécanisme d'entrée et qui ne doit recevoir une confiance limitée. Note : lorsque vous exécutez un script en ligne de commande, cette variable ne va **pas** inclure les variables argv et argc . Elles seront présentes dans la variable $_SERVER . La présence et la valeur des entrées de ce tableau sont réglés par la directive variables_order . Ce tableau n'est l'évolution d'aucune variable d'avant PHP 4.1.0.

*$_SESSION*
: Les variables qui sont actuellement enregistrées dans la session attachée au script. C'est la nouvelle version de l'ancienne variable $HTTP_SESSION_VARS . Voir le chapitre gestion des sessions pour plus d'informations.

## 4.3.3 Portée des variables

La portée d'une variable dépend du contexte dans lequel la variable est définie. Pour la majorité des variables, la portée concerne la totalité d'un script PHP . Mais, lorsque vous définissez une fonction, la portée d'une variable définie dans cette fonction est locale à la fonction. Par exemple:

Les variables sont locales à la fonction

```
<?php
$a = 1;
include "b.inc";
```

```
?>
```

Ici, la variable $a sera accessible dans le script inclus b.inc . Cependant, dans les fonctions définies par l'utilisateur, une nouvelle définition de cette variable sera donnée, limitée à la fonction. Toute variable utilisée dans une fonction est par définition, locale. Par exemple :

Les variables sont locales à la fonction (2)

```
<?php
$a = 1; /* portée globale */
function test() {
    echo $a; /* portée locale */
}
test();
?>
```

Le script n'affichera rien à l'écran car la fonction echo utilise la variable locale $a , et celle−ci n'a pas été assignée préalablement dans la fonction. Vous pouvez noter que ce concept diffère un petit peu du langage C dans lequel une variable globale est automatiquement accessible dans les fonctions, à moins d'être redéfinie localement dans la fonction. Cela peut poser des problèmes si vous redéfinissez des variables globales localement. En PHP , une variable globale doit être déclarée à l'intérieur de chaque fonction afin de pouvoir être utilisée dans cette fonction.

### 4.3.3.1 Le mot clé global

Commençons par un exemple avec global :

Exemple avec global

```
<?php
$a = 1;
$b = 2;
function somme() {
    global $a, $b;
    $b = $a + $b;
}
somme();
echo $b;
```

Le script ci−dessus va afficher la valeur 3 . En déclarant globales les variables $a et $b locales de la fonction somme(), toutes les références à ces variables concerneront les variables globales. Il n'y a aucune limite au nombre de variables globales qui peuvent être manipulées par une fonction.

Une deuxième méthode pour accéder aux variables globales est d'utiliser le tableau associatif prédéfini $GLOBALS . Le précédent exemple peut être réécrit de la manière suivante:

Les variables globales et $GLOBALS

```
<?php
 $a = 1;
 $b = 2;
 function somme() {
   $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"];
 }
 somme();
 echo $b;
?>
```

Le tableau $GLOBALS est un tableau associatif avec le nom des variables globales comme clef et les valeurs des éléments du tableau comme valeur des variables. Notez que $GLOBALS existe dans tous les contextes, car $GLOBALS est un superglobal . Voici un exemple des super globaux :

Les variables super globales

```
<?php
function test_global()
{
    // La plupart des variables prédéfinies ne sont pas des "superglobales" et
    // requiert le mot clé 'global' pour être disponible dans une fonction.
    global $HTTP_POST_VARS;
```

```
    print $HTTP_POST_VARS['name'];

    // Les superglobales sont accessibles dans tous les contextes
    // et ne requièrent pas 'global'.  Les superglobales sont disponibles
    // depuis
 <font face="courrier">PHP</font>
 4.1.0
    print $_POST['name'];
}
?>
```

### 4.3.3.2 Utilisation des variables static

Une autre caractéristique importante de la portée des variables est la notion de variable static . Une variable statique a une portée locale uniquement, mais elle ne perd pas sa valeur lorsque le script appelle la fonction. Prenons l'exemple suivant:

```
Les variables statiques
 <?php
function test() {
    $a = 0;
    echo $a;
    $a++;
}
?>
```

Cette fonction est un peu inutile car à chaque fois qu'elle est appelée, elle initialise $a à 0 et affiche "0". L'incrémentation de la variable ( $a ++) ne sert pas à grand chose, car dès que la fonction est terminée la variable disparaît. Pour faire une fonction de comptage utile, c'est−à−dire qui ne perdra pas la trace du compteur, la variable $a est déclarée comme une variable statique:

```
Les variables statiques (2)
 <?php
function test() {
    static $a = 0;
    echo $a;
    $a++;
}
?>
```

Maintenant, à chaque fois que la fonction Test() est appelée, elle affichera une valeur de $a incrémentée de 1.

Les variables statiques sont essentielles lorsque vous faites des appels récursifs à une fonction. Une fonction récursive est une fonction qui s'appelle elle−même. Il faut faire attention lorsque vous écrivez une fonction récursive car il est facile de faire une boucle infinie. Vous devez vérifier que vous avez bien une condition qui permet de terminer votre récursivité. La fonction suivante compte récursivement jusqu'à 10:

```
Les variables statiques et la récursivité
 <?php
function test() {
    static $count = 0;
    $count++;
    echo $count;
    if ($count < 10) {
        test();
    }
    $count−−;
}
?>
```

### 4.3.3.3 Les références avec la variables global et static

Le Zend Engine 1, sur qui repose PHP 4 , implémente les options static et global pour les variables, en terme de références. Par exemple, une vraie variable globale est importée dans un contexte de fonction avec global . Cette commande crée en fait une référence sur la variable globale. Ce peut

vous mener à des comportement inattendus, comme par exemple :

```
Les variables statiques et les références
 <?php
function test_global_ref() {
    global $obj;
    $obj = &new stdclass;
}

function test_global_noref() {
    global $obj;
    $obj = new stdclass;
}

test_global_ref();
var_dump($obj);
test_global_noref();
var_dump($obj);
?>
```

Exécuter cet exemple vous donnera :

NULL
object(stdClass)(0) {
}

Un comportement similaire s'applique à la commande static . Les références ne sont pas stockées dynamiquement :

```
Les variables statiques et les références (2)
 <?php
function &get_instance_ref() {
    static $obj;

    echo "Objet statique : ";
    var_dump($obj);
    if (!isset($obj)) {
        // Assigne une référence à une variable statique
        $obj = &new stdclass;
    }
    $obj->property++;
    return $obj;
}

function &get_instance_noref() {
    static $obj;

    echo "Objet statique : ";
    var_dump($obj);
    if (!isset($obj)) {
        // Assigne une objet à une variable statique
        $obj = new stdclass;
    }
    $obj->property++;
    return $obj;
}

$obj1 = get_instance_ref();
$still_obj1 = get_instance_ref();
echo "\n";
$obj2 = get_instance_noref();
$still_obj2 = get_instance_noref();
?>
```

Exécuter cet exemple vous donnera :

Objet statique : NULL
Objet statique : NULL

Objet statique : NULL
Objet statique : object(stdClass)(1) {
["property"]=>
int(1)

}

Ces exemples illustrent les problèmes rencontrés lors de l'assignation de référence à des variables statiques, qui sont **oubliées** lorsque vous appelez &get_instance_ref() une seconde fois.

## 4.3.4 Les variables dynamiques

Il est pratique d'avoir parfois des noms de variables qui sont variables. C'est−à−dire un nom de variable qui est affectée et utilisée dynamiquement. Une variable classique est affecté avec l'instruction suivante:

Une variable classique

```
<?php
$a = "bonjour";
?>
```

Une variable dynamique prend la valeur d'une variable et l'utilise comme nom d'une autre variable. Dans l'exemple ci−dessous, **bonjour** peut être utilisé comme le nom d'une variable en utilisant le "$$" précédent la variable. C'est−à−dire

Les variables variables

```
<?php
$$a = "monde";
?>
```

A ce niveau, deux variables ont été définies et stockées dans l'arbre des symboles PHP : $a avec comme valeur "bonjour" et $bonjour avec comme valeur "monde". Alors, l'instruction

Equivalent de variables variables

```
<?php
echo "$a ${$a}";
?>
```

produira le même affichage que :

Equivalent de variables variables

```
<?php
echo "$a $bonjour";
?>
```

c'est−à−dire : **bonjour monde** .

Afin de pouvoir utiliser les variables dynamiques avec les tableaux, vous avez à résoudre un problème ambigu. Si vous écrivez $$a[1] , l'analyseur a besoin de savoir si vous parler de la variable qui a pour nom $a[1] ou bien si vous voulez l'index [1] de la variable $$a . La syntaxe pour résoudre cette ambiguÔté est la suivante: ${$a[1]} pour le premier cas, et ${$a}[1] pour le deuxième.

## 4.3.5 Variables externes à PHP

### 4.3.5.1 Formulaires HTML (GET et POST)

Lorsqu'un formulaire est envoyé à un script PHP , toutes les variables du formulaire seront automatiquement disponibles dans le script. Par exemple, considérons le formulaire suivant:

Exemple avec un formulaire simple

```
<form action="foo.php" method="post">
   Name:  <input type="text" name="username"><br>
   Email: <input type="text" name="email"><br>
   <input type="submit" name="submit" value="Submit me!">
</form>
```

Suivant votre configuration particulière et vos préférences, vous avez plusieurs méthodes pour accéder aux variables du formulaires. Voici quelques exemples :

```
Accéder simplement à des variables de formulaires POST
<?php
// Disponibles depuis
<font face="courrier">PHP</font>
4.1.0

  print $_POST['username'];
  print $_REQUEST['username'];

  import_request_variables('p', 'p_');
  print $p_username;

// Disponibles depuis
<font face="courrier">PHP</font>
3.

  print $HTTP_POST_VARS['username'];

// Disponibles si la directive register_globals = on.  Depuis
//
<font face="courrier">PHP</font>
4.2.0 la valeur par défaut de cette directive est register_globals = off.
// Utiliser ou présumer cette méthode est découragé.

  print $username;
?>
```

Utiliser un formulaire de type GET est similaire, hormis le fait que vous deviez utiliser les variables prédéfinies du GET à la place. GET s'applique aussi à la QUERY_STRING (les informations disponibles après le '?' dans une URL). De ce fait, par exemple, http://www.exemple.com/test.php?id=3 contient les données de GET, qui sont accessibles via $_GET['id'] . Voyez aussi $_REQUEST et import_request_variables .

```
Note
```

Les tableaux superglobaux , comme $_POST et $_GET , sont disponibles depuis PHP 4.1.0

Comme nous l'avons déjà dis, avant PHP 4.2.0 la valeur par défaut de register_globals était **on** . Dans PHP 3 elle était toujours à on. La communauté PHP n'encourage personne à utiliser cette directive et privilégie la valeur **off** et un code accordé.

```
Note
```

La directive de configuration magic_quotes_gpc affecte les valeurs de GET, POST et cookies. Si elle est activée, une valeur comme celle de (C'est " PHP !") sera magiquement transformée en (C\'est \" PHP !\"). La protection des caractères est nécessaire pour l'insertion dans les bases de données. Voyez aussi les fonctions addslashes , stripslashes et magic_quotes_sybase .

PHP comprend aussi les tableaux dans le contexte des formulaires. (voir aussi la FAQ ). Vous pouvez, par exemple, grouper des variables ensembles, ou bien utiliser cette fonctionnalité pour lire des valeurs multiples d'un menu déroulant. Par exemple, voici un formulaire qui se poste lui même des données, et les affiche :

```
Variables de formulaires complexes
<?php
if ($_POST['action'] == 'submitted') {
    print '<pre>';

    print_r($_POST);
    print '<a href="'. $_SERVER['PHP_SELF'] .'">Essayez à nouveau</a>';

    print '</pre>';
} else {
?>
```

```
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    Name:  <input type="text" name="personal[name]"><br>
    Email: <input type="text" name="personal[email]"><br>
    Beer: <br>
    <select multiple name="vin[]">
        <option value="bordeaux">bordeaux</option>
        <option value="beaujolais">beaujolais</option>
        <option value="loire">loire</option>
    </select><br>
    <input type="hidden" name="action" value="submitted">
    <input type="submit" name="submit" value="submit me!">
</form>
<?php
}
?>
```

En PHP 3, la syntaxe des tableaux HTML est limitée aux tableaux uni−dimensionnels. En PHP 4, ces limitations ont été levées.

#### 4.3.5.1.1 Nom de variables IMAGE de type SUBMIT

Lors de la soumission d'un formulaire, il est possible d'utiliser une image au lieu d'un bouton standard, comme ceci :

```
Utilisation d'une image pour soumettre un formulaire
 <input type="image" src="image.gif" name="sub">
```

Lorsque l'utilisateur clique sur cette image, le formulaire associé est envoyé au serveur, avec deux données supplémentaires, sub_x et sub_y . Elles contiennent les coordonnées du clic de l'utilisateur dans l'image. Vous noterez que ces variables sont envoyées par le navigateur avec un point dans leur nom, mais PHP convertit ces points en soulignés.

### 4.3.5.2 Cookies HTTP

PHP supporte les cookies HTTP de manière totalement transparente, comme défini dans les spécifications de Netscape . Les cookies sont un mécanisme permettant de stocker des données sur la machine cliente à des fins d'authentification de l'utilisateur. Vous pouvez établir un cookie grâce à la fonction setcookie . Les cookies font partie intégrante des entêtes HTTP, et donc la fonction setcookie doit être appelée avant que le moindre affichage ne soit envoyé au navigateur. C'est la même restriction que pour la fonction header . Tout cookie envoyé depuis le client sur le serveur sera automatiquement stocké sous forme de variable, comme pour la méthode POST ou GET.

Si vous souhaitez assigner plusieurs valeurs à un seul cookie, il vous faut ajouter les caractères **[]** au nom de votre cookie. Par exemple :

```
Utilisation de tableaux avec les cookies
 <?php
setcookie ("MonCookie[]", "test", time()+3600);
?>
```

Il est à noter qu'un cookie remplace le cookie précédent par un cookie de même nom tant que le " path " ou le domaine sont identiques. Donc, pour une application de caddie, vous devez implémenter un compteur et l'incrémenter au fur et à mesure. C'est−à−dire:

```
Exemple avec setcookie
 <?php
$compte++;
SetCookie ("Compte", $compte, time()+3600);
SetCookie ("Caddie[$compte]", $item, time()+3600);
?>
```

#### 4.3.5.3 Cas des points dans les noms de variables

Typiquement, PHP ne modifie pas les noms des variables lorsqu'elles sont passées à un script. Cependant, il faut noter que les points (.) ne sont pas autorisés dans les noms de variables PHP . Pour cette raison, jetez un oeil sur :

```
 <?php
  $varname.ext;  /* nom de variable invalide */
?>
```

Dans ce cas, l'analyseur croit voir la variable nommée $varname , suivie par l'opérateur de concaténation, et suivi encore par la chaîne sans guillemets (une chaîne sans guillemets, et qui n'a pas de signification particulière). Visiblement, ce n'est pas ce qu'on attendait...

Pour cette raison, il est important de noter que PHP remplacera automatiquement les points des noms de variables entrantes par des soulignés ( underscore ).

#### 4.3.5.4 Détermination du type des variables

Parceque PHP détermine le type des variables et les convertit (généralement) comme il faut, ce n'est pas toujours le type de variable que vous souhaitez. PHP inclut des fonctions permettant de déterminer le type d'une variable : gettype , is_long , is_double , is_string , is_array et is_object .

## 4.4 Les constantes

Une constante est un identifiant (un nom) qui représente une valeur simple. Comme son nom le suggére, cette valeur ne peut jamais étre modifiée durant l'exécution du script (les constantes magiques __FILE__ et __LINE__ sont les seules exceptions). Le nom d'une constante est sensible à la casse, par défaut. Par convention, les constantes sont toujours en majuscules.

Les noms de constantes suivent les mémes régles que n'importe quel nom en PHP. Un nom de constante valide commence par une lettre ou un souligné ( _ ), suivi d'un nombre quelconque de lettre, chiffres ou soulignés. Sous forme d'expression réguliére, cela peut s'exprimer comme ceci : [a−zA−Z_\x7f−\xff][a−zA−Z0−9_\x7f−\xff]*

| Note |
| --- |
| Dans cette documentation, une lettre peut étre un des caractères suivants : de a à z, de A à Z et tous les caractères ASCII de 127 à 255 ( 0x7f−0xff ). |

Tous comme les superglobals , les constantes sont accessibles de maniére globale. Vous pouvez la définir n'importe où, et y accéder depuis n'importe quelle fonction.

### 4.4.1 Syntaxe

Vous pouvez définir une constante en utilisant la fonction define . Une fois qu'une constante est définie, elle ne peut jamais étre modifiée, ou détruite.

Seuls les types de données scalaires peuvent étre placés dans une constante : c'est à dire les types booléen, entier, double et chaîne de caractères (soit bool , integer , double et string ).

Vous pouvez accéder à la valeur d'une constante en spécifiant simplement son nom. Contrairement aux variables, vous ne devez **PAS** préfixer le nom de la constante avec $ . Vous pouvez aussi

utiliser la fonction constant , pour lire dynamiquement la valeur d'une constante, dont vous obtenez le nom dynamiquement (retour de fonction, par exemple). Utilisez la fonction get_defined_constants pour connaître la liste de toutes les constantes définies.

| Note |
| --- |
| Les constantes et les variables globales utilisent deux espaces de noms différents. Ce qui implique que TRUE et $TRUE sont généralement différents (en tous cas, ils peuvent avoir des valeurs différentes). |

Lorsque vous utilisez une constante non définie, PHP suppose que vous utilisez le nom de la constante. Une note sera générée. Utilisez la fonction defined pour savoir si une constante existe ou pas.

Il y a des différences entre les constantes et les variables :

- Les constantes ne commencent pas par le signe ( $ );
- Les constantes sont définies et accessibles à tout endroit du code, globalement.
- Les constantes ne peuvent pas étre redéfinies ou indéfinies une fois qu'elles ont été définies.
- Les constantes ne peuvent contenir que des scalaires.

Définir une constante

```
 <?php
  define("CONSTANTE", "Bonjour le monde.");
  echo CONSTANTE; // affiche "Bonjour le monde."
  echo Constante; // affiche "Constante" et une note.
?>
```

### 4.4.2 Constantes prédéfinies

PHP fournit un grand nombre de constantes prédéfinies. Certaines constantes sont définies par différentes extensions, et ne seront présentes que si ces extensions sont compilées avec PHP , ou bien si l'extension a été chargée dynamiquement.

Une liste des constantes prédéfinies est disponible dans la section Constantes prédéfinies .

## 4.5 Les expressions

Les expressions sont la partie la plus importante du PHP . En PHP , presque tout ce que vous écrivez est une expression. La manière la plus simple de définir une expression est : "tout ce qui a une valeur".

Les formes les plus simples d'expressions sont les constantes et les variables. Lorsque vous écrivez " $a = 5", vous assignez la valeur '5' à la variable $a . Bien évidemment, '5' vaut 5 ou, en d'autres termes, '5' est une expression avec pour valeur 5 (dans ce cas, '5' est un entier constant).

Après cette assignation, vous pouvez considérer que $a a pour valeur 5 et donc, écrire $b = $a , revient à écrire $b = 5. En d'autres termes, $a est une expression avec une valeur de 5. Si tout fonctionne correctement, c'est exactement ce qui arrive.

Un exemple plus complexe concerne les fonctions. Par exemple, considérons la fonction suivante :

```
 <?php
function foo () {
    return 5;
```

```
}
?>
```

Considérant que vous êtes familier avec le concept de fonction, (si ce n'est pas le cas, jetez un oeil au chapitre concernant les fonctions), vous serez d'accord que $c = foo() est équivalent à $c = 5 , et vous aurez tout à fait raison. Les fonctions sont des expressions qui ont la valeur de leur "valeur de retour". Si foo() renvoie 5, la valeur de l'expression ' foo() ' est 5. Habituellement, les fonctions ne font pas que renvoyer une valeur constante mais réalisent des traitements.

Bien sûr, les valeurs en PHP n'ont pas à être des valeurs numériques, comme c'est souvent le cas. PHP supporte 3 types de variables scalaires : les valeurs entières, les nombres à virgule flottante et les chaînes de caractères (une variable scalaire est une variable que vous ne pouvez pas scinder en morceaux, au contraire des tableaux par exemple). PHP supporte aussi deux types composés : les tableaux et les objets. Chacun de ces types de variables peut être affecté ou renvoyé par une fonction.

Les utilisateurs de PHP /FI 2 ne verront aucun changement. Malgré tout, PHP va plus loin dans la gestion des expressions, comme le font d'autres langages. PHP est un langage orienté expression, dans le sens où presque tout est une expression. Considérons l'exemple dont nous avons déjà parlé, ' $a = 5'. Il est facile de voir qu'il y a deux valeurs qui entrent en jeu ici, la valeur numérique constante '5' et la valeur de la variable $a qui est mise à jour à la valeur 5. Mais, la vérité est qu'il y a une autre valeur qui entre en jeu ici et c'est la valeur de l'assignation elle−même. L'assignation elle−même est assignée à une valeur, dans ce cas−là 5. En pratique, cela signifie que ' $a = 5' est une expression qui a pour valeur 5. Donc, écrire ' $b = ( $a = 5)' revient à écrire ' $a = 5; $b = 5;' (un point virgule marque la fin d'une instruction). Comme les assignations sont analysées de droite à gauche, vous pouvez aussi bien écrire ' $b = $a = 5'.

Un autre bon exemple du langage orienté expression est la pré−incrémentation et la post−incrémentation, (ainsi que la décrémentation). Les utilisateurs de PHP /FI 2 et ceux de nombreux autres langages sont habitués à la notation " variable++ " et " variable−− ". Ce sont les opérateurs d'incrémentation et de décrémentation. En PHP /FI 2, l'instruction ' $a ++' n'a aucune valeur (c'est−à−dire que ce n'est pas une expression) et vous ne pouvez donc pas l'utiliser. PHP ajoute les possibilités d'incrémentation et de décrémentation comme c'est le cas dans le langage C. En PHP , comme en C, il y a deux types d'opérateurs d'incrémentation (pré−incrémentation et post−incrémentation). Les deux types d'opérateur d'incrémentation jouent le même rôle (c'est−à−dire qu'ils incrémentent la variable). La différence vient de la valeur de l'opérateur d'incrémentation. L'opérateur de pré−incrémentation, qui s'écrit '++ $variable ', évalue la valeur incrémentée ( PHP incrémente la variable avant de lire la valeur de cette variable, d'où le nom de 'pré−incrémentation'). L'opérateur de post−incrémentation, qui s'écrit ' $variable ++', évalue la valeur de la variable avant de l'incrémenter ( PHP incrémente la variable après avoir lu sa valeur, d'où le nom de 'post−incrémentation').

Un type d'expression très commun est l'expression de comparaison. Ces expressions sont évaluées à 0 ou 1, autrement dit FALSE ou TRUE (respectivement). PHP supporte les opérateurs de comparaison > (plus grand que), => (plus grand ou égal), == (égal à), < (plus petit que), <= (plus petit ou égal). Ces expressions sont utilisées de manière courante dans les instructions conditionnelles, comme l'instruction if .

Pour le dernier exemple d'expression, nous allons parler des combinaisons d'opérateurs/assignation. Vous savez que si vous voulez incrémenter la variable $a d'une unité, vous devez simplement écrire ' $a ++'. Mais si vous voulez ajouter la valeur '3' à votre variable ? Vous pouvez écrire plusieurs fois ' $a ++', mais ce n'est pas la meilleure des méthodes. Un pratique plus courante est d'écrire ' $a = $a + 3'. L'expression ' $a + 3' correspond à la valeur $a plus 3, et est de nouveau assignée à la variable $a . Donc, le résultat est l'incrémentation de 3 unités. En PHP , comme dans de nombreux autres langages comme le C, vous pouvez écrire cela de manière

plus concise, manière qui avec le temps se révélera plus claire et plus rapide à comprendre. Ajouter 3 à la valeur de la variable $a peut s'écrire ' $a += 3'. Cela signifie précisément : "on prend la valeur de la variable $a , on ajoute la valeur 3 et on assigne cette valeur à la variable $a ". Et pour être plus concis et plus clair, cette expression est plus rapide. La valeur de l'expression ' $a += 3', comme l'assignation d'une valeur quelconque, est la valeur assignée. Il est à noter que ce n'est pas 3 mais la combinaison de la valeur de la variable $a plus la valeur 3. (c'est la valeur qui est assignée à la variable $a ). N'importe quel opérateur binaire peut utiliser ce type d'assignation, par exemple ' $a −= 5' (soustraction de 5 de la valeur de la variable $a ), ' $b *= 7' (multiplication de la valeur de la variable $b par 7).

Il y a une autre expression qui peut paraître complexe si vous ne l'avez pas vue dans d'autres langages, l'opérateur conditionnel ternaire :

```
<?php
 $first ? $second : $third
?>
```

Si la valeur de la première sous−expression est vraie (différente de 0), alors la deuxième sous−expression est évaluée et constitue le résultat de l'expression conditionnelle. Sinon, c'est la troisième sous−expression qui est évaluée et qui constitue le résultat de l'expression.

Les exemples suivants devraient vous permettre de mieux comprendre la pré−incrémentation, la post−incrémentation et le concept des expressions en général :

```
<?php
function double($i) {
    return $i*2;
}

$b = $a = 5;        /* assigne la valeur 5 aux variables $a et $b  */
$c = $a++;          /* post−incrémentation de la variable $a et assignation de
                       la valeur à la variable $c */
$e = $d = ++$b;     /* Pré−incrémentation, et assignation de la valeur aux
                       variables $d et $e  */
/* à ce niveau, les variables $d et $e sont égales à 6 */
$f = double($d++);  /* assignation du double de la valeur de $d à la variable $f ($f vaut 12),
                       puis incrémentation de la valeur de $d  */
$g = double(++$e);  /* assigne deux fois la valeur de $e <emphasis>après</emphasis>
                       incrémentation, 2*7 = 14 to $g */
$h = $g += 10;      /* Tout d'abord, $g est incrémentée de 10, et donc $g vaut 24.
                       Ensuite, la valeur de $g, (24) est assignée à la variable $h,
                       qui vaut donc elle aussi 24. */
?>
```

Au début de ce chapitre, nous avons dit que nous allions décrire les différents types d'instructions, et donc, comme promis, nous allons voir que les expressions peuvent être des instructions. Mais, attention, toutes les expressions ne sont pas des instructions. Dans ce cas−là, une instruction est de la forme ' expr ; ', c'est−à−dire, une expression suivie par un point−virgule. L'expression '$b = $a = 5;', '$a = 5' est valide, mais ce n'est pas une instruction en elle−même. '$b = $a = 5' est une instruction valide.

La dernière chose qui mérite d'être mentionnée est la véritable valeur des expressions. Lorsque vous faites des tests sur une variable, dans une boucle conditionnelle par exemple, cela ne vous intéresse pas de savoir quelle est la valeur exacte de l'expression. Mais vous voulez seulement savoir si le résultat signifie TRUE ou FALSE ( PHP n'a pas de type booléen). La véritable valeur d'une expression en PHP est calculée de la même manière qu'en Perl. Toute valeur numérique différente de 0 est considérée comme étant TRUE . Une chaîne de caractères vide et la chaîne de caractère 0 sont considérées comme FALSE . Toutes les autres valeurs sont vraies. Avec les types de variables non−scalaires (les tableaux et les objets), s'ils ne contiennent aucun élément, renvoient FALSE , sinon, ils renvoient TRUE .

PHP propose une implémentation complète et détaillée des expressions. PHP documente toutes ses expressions dans le manuel que vous êtes en train de lire. Les exemples qui vont suivre

devraient vous donner une bonne idée de ce qu'est une expression et comment construire vos propres expressions. Dans tout ce qui va suivre, nous écrirons expr pour indiquer toute expression PHP valide.

# 4.6 Les opérateurs

## 4.6.1 La précédence des opérateurs

La priorité des opérateurs spécifie l'ordre dans lequel les valeurs doivent être analysées. Par exemple, dans l'expression 1 + 5 * 3, le résultat est 16 et non 18, car la multiplication ("*") a une priorité supérieure par rapport à l'addition ("+").

Le tableau suivant dresse une liste de la priorité des différents opérateurs dans un ordre croissant de priorité.

| Associativité | Opérateurs |
|---|---|
| gauche | , |
| gauche | or |
| gauche | xor |
| gauche | and |
| droite | print |
| droite | = += -= *= /= .= %= &= \|= ^= <<= >>= |
| gauche | ? : |
| gauche | \|\| |
| gauche | && |
| gauche | \| |
| gauche | ^ |
| gauche | & |
| non-associatif | == != === !== |
| non-associatif | < <= > >= |
| gauche | << >> |
| gauche | + - . |
| gauche | * / % |
| droite | ! ~ ++ -- (int) (float) (string) (array) (object) @ |
| droite | [ |
| non-associatif | new |

| Note |
|---|
| Bien que ! soit prioritaire sur = , PHP va tout de même exécuter des expressions comme : if (!$a = foo()) . Dans cette situation, le résultat de foo() sera placé dans la variable $a . |

## 4.6.2 Les opérateurs arithmétiques

Vous rappelez-vous des opérations élémentaires apprises à l'école ?

| Exemple | Nom | Résultat |
|---|---|---|
| $a + $b | Addition | Somme de $a et $b . |
| $a - $b | Soustraction | Différence de $a et $b . |
| $a * $b | Multiplication | Produit de $a et $b . |
| $a / $b | Division | Quotient de $a et $b . |
| $a % $b | Modulo | Reste de $a divisé par $b . |

L'opérateur de division ("/") retourne une valeur entière (le résultat d'une division entière) si les deux opérandes sont entiers (ou bien des chaînes converties en entier. Si l'un des opérandes est un nombre à virgule flottante, ou bien le résultat d'une opération qui retourne une valeur non entière, un nombre à virgule flottante sera retourné.

Voir aussi le manuel sur les fonctions mathématiques .

## 4.6.3 Les opérateurs d'assignation

L'opérateur d'assignation le plus simple est le signe "=". Le premier réflexe est de penser que ce signe veut dire "égal à". Ce n'est pas le cas. Il signifie que l'opérande de gauche se voit affecter la valeur de l'expression qui est à droite du signe égal.

La valeur d'une expression d'assignation est la valeur assignée. Par exemple, la valeur de l'expression ' $a = 3 ' est la valeur 3. Cela permet d'utiliser des astuces telles que :

Astuce d'assignation

```
<?php
 $a = ($b = 4) + 5;
// $a est maintenant égal à 9, et $b vaut 4.
?>
```

En plus du simple opérateur d'assignation, il existe des "opérateurs combinés" pour tous les opérateurs arithmétiques et pour les opérateurs sur les chaînes de caractères. Cela permet d'utiliser la valeur d'une variable dans une expression et d'affecter le résultat de cette expression à cette variable. Par exemple :

Exemples avec les opérateurs d'assignation

```
<?php
 $a = 3;
 $a += 5;
// affecte la valeur 8 à la variable $a.
// correspond à l'instruction '$a = $a + 5');
 $b = "Bonjour ";
 $b .= " tout le monde!";
// affecte la valeur "Bonjour tout le monde!" à
//  la variable $b
//  identique à $b = $b." tout le monde!";
?>
```

On peut noter que l'assignation copie le contenu de la variable originale dans la nouvelle variable (assignation par valeur), ce qui fait que les changements de valeur d'une variable ne modifieront pas la valeur de l'autre. Cela peut se révéler important lors de la copie d'un grand tableau durant une boucle. PHP 4 supporte aussi l'assignation par référence, en utilisant la syntaxe $var =  , mais ce n'était pas possible en PHP 3. 'L'assignation par référence' signifie que les deux variables contiennent les mêmes données, et que la modification de l'une affecte l'autre. D'un autre côté, la recopie est très rapide.

## 4.6.4 Opérateurs sur les bits

Les opérateurs sur les bits vous permettent de manipuler les bits dans un entier.

Exemples avec les opérateurs sur les bits

```
<?php
   echo 12 ^ 9;
// Affiche '5'

   echo "12" ^ "9";
// Affiche le caractère d'effacement (ascii 8)
// ('1' (ascii 49)) ^ ('9' (ascii 57)) = #8

echo "hallo" ^ "hello";
// Affiche la valeur asciie : s #0 #4 #0 #0 #0
// 'a' ^ 'e' = #4
?>
```

| Exemple | Nom | Résultat |
|---|---|---|
| $a &$b | ET (AND) | Les bits positionnés à 1 dans $a ET dans $b sont positionnés à 1. |

| | | |
|---|---|---|
| $a \| $b | OU (OR) | Les bits positionnés à 1 dans $a OU $b sont sont positionnés à 1. |
| $a ^ $b | Xor | Les bits positionnés à 1 dans $a OU dans $b mais pas dans les deux sont positionnés à 1. |
| ~ $a | NON ( Not ) | Les bits qui sont positionnés à 1 dans $a sont positionnés à 0, et vice versa. |
| $a << $b | Décalage à gauche | Décale les bits de $a $b fois sur la gauche (chaque décalage équivaut à une multiplication par 2). |
| $a >> $b | Décalage à droite | Décalage des bits de $a $b fois par la droite (chaque décalage équivaut à une division par 2). |

## 4.6.5 Opérateurs de comparaison

Les opérateurs de comparaison, comme leur nom l'indique, vous permettent de comparer deux valeurs.

| Exemple | Nom | Résultat |
|---|---|---|
| $a == $b | Egal | Vrai si $a est égal à $b . |
| $a === $b | Identique | Vrai si $a est égal à $b et qu'ils sont de même type ( PHP 4 seulement). |
| $a != $b | Différent | Vrai si $a est différent de $b . |
| $a <> $b | Différent | Vrai si $a est différent de $b . |
| $a < $b | Plus petit que | Vrai si $a est plus petit strictement que $b . |
| $a > $b | Plus grand | Vrai si $a est plus grand strictement que $b . |
| $a <= $b | Inférieur ou égal | Vrai si $a est plus petit ou égal à $b . |
| $a >= $b | Supérieur ou égal | Vrai si $a est plus grand ou égal à $b . |

Un autre opérateur conditionnel est l'opérateur ternaire (":?"), qui fonctionne comme en langage C.

```
 <?php
// Exemple d'utilisation pour l'opérateur ternaire
$action = (empty($_POST['action'])) ? 'default' : $_POST['action'];

// L'expression ci-dessus est identique à
 if (empty($_POST['action'])) {
    $action = 'default';
} else {
    $action = $_POST['action'];
}
?>
```

L'expression (expr1) ? (expr2) : (expr3) renvoie la valeur de l'expression expr2 si l'expression expr1 est vraie, et l'expression expr3 si l'expression expr1 est fausse.

Voir aussi strcasecmp , strcmp et le chapitre sur les types .

## 4.6.6 Opérateur de contrôle d'erreur

PHP supporte un opérateur de contrôle d'erreur : c'est @. Lorsque cet opérateur est ajouté en préfixe d'une expression PHP , les messages d'erreur qui peuvent être générés par cette expression seront ignorés.

Si l'option track_errors est activée, les messages d'erreurs générés par une expression seront sauvés dans la variable globale $php_errormsg . Cette variable sera écrasée à chaque erreur. Il faut alors la surveiller souvent pour pouvoir l'utiliser.

```
 <?php
/* Erreur intentionnelle (le fichier n'existe pas): */
  $my_file = @file ('non_persistent_file') or
    die ("Impossible d'ouvrir le fichier : L'erreur est : '$php_errormsg'");
// Cela fonctionne avec n'importe quelle expression, pas seulement les fonctions
  $value = @$cache[$key];
// la ligne ci-dessus n'affichera pas d'alerte si la clé $key du tableau n'existe pas
?>
```

| Note |
|---|
| L'opérateur @ ne fonctionne qu'avec les expressions. La règle générale de fonctionnement est la suivante : si vous pouvez prendre la valeur de quelque chose, vous pouvez le préfixer avec @. Par exemple, vous pouvez ajouter @ aux variables, fonctions, à include , aux constantes, etc. Vous ne |

pourrez pas le faire avec des éléments de langage tels que les classes, if et foreach , etc.

| Note |
| --- |
| La plupart des fonctions d'accès aux bases de données ne retournent pas d'erreur PHP . Il faut y accéder avec une fonction du type base_de_donnees_get_error() . |

Voir aussi error_reporting .

## 4.6.7 Opérateur d'exécutions

PHP supporte un opérateur d'exécution : guillemets obliques ("``"). Notez bien la différence entre les guillemets simples (sur la touche 4), et ceux−ci (sur la touche de la livre anglaise). PHP essaiera d'exécuter le contenu de ces guillemets obliques comme une commande shell. Le résultat sera retourné (i.e. : il ne sera pas simplement envoyé à la sortie standard, il peut être assigné à une variable).

```
<?php
 $output = `ls -al`;
 echo "<pre>$output</pre>";
?>
```

| Note |
| --- |
| Cet opérateur est désactivé lorsque le safe mode est activé ou bien que la fonction shell_exec est désactivée. |

Voir aussi system , passthru , exec , shell_exec , popen et escapeshellcmd .

## 4.6.8 Opérateurs d'incrémentation/Décrémentation

PHP supporte les opérateurs de pré et post incrémentation et décrémentation, comme en C.

| Exemple | Nom | Résultat |
| --- | --- | --- |
| ++$a | Pré−incrémente | Incrémente $a de 1, puis retourne $a . |
| $a++ | Post−incrémente | Retourne $a , puis l'incrémente de 1. |
| −−$a | Pré−décrémente | Décrémente $a de 1, puis retourne $a . |
| $a−− | Post−décrémente | Retourne $a , puis décrémente $a de 1. |

Voici un exemple simple

```
<?php
 echo "<h3>Post−incrémentation</h3>";
 $a = 5;
 echo "Devrait valoir  5: " . $a++ . "<br>\n";
 echo "Devrait valoir  6: " . $a . "<br>\n";
 echo "<h3>Pré−incrémentation</h3>";
 $a = 5;
 echo "Devrait valoir  6: " . ++$a . "<br>\n";
 echo "Devrait valoir  6: " . $a . "<br>\n";
 echo "<h3>Post−décrémentation</h3>";
 $a = 5;
 echo "Devrait valoir  5: " . $a−− . "<br>\n";
 echo "Devrait valoir  4: " . $a . "<br>\n";
 echo "<h3>Pré−décrémentation</h3>";
 $a = 5;
 echo "Devrait valoir  4: " . −−$a . "<br>\n";
 echo "Devrait valoir  4: " . $a . "<br>\n";
?>
```

PHP suit les conventions de Perl pour la gestion des opérateurs arithmétiques, et non pas celle du C. Par exemple, en Perl 'Z'+1 retourne 'AA' , alors qu'en C, 'Z'+1 retourne '[' ( ord('Z') == 90 , donc ord('[') == 91 ). Notez que les variables de caractères peuvent être incrémentées, mais pas décrémentées.

Opérations arithmétiques sur un caractère

```
<?php
$i = 'W';
for($n=0; $n<6; $n++)
  echo ++$i . "\n";

/*
  Produit un résultat identique à celui-ci :

X
Y
Z
AA
AB
AC

*/
?>
```

## 4.6.9 Les opérateurs logiques

| Exemple | Nom | Résultat |
|---|---|---|
| $a and $b | ET (And) | Vrai si $a ET $b sont vrais. |
| $a or $b | OU (Or) | Vrai si $a OU $b est vrai |
| $a xor $b | XOR (Xor) | Vrai si $a OU $b est vrai, mais pas les deux en même temps. |
| ! $a | NON (Not) | Vrai si $a est faux. |
| $a &$b | ET (And) | Vrai si $a ET $b sont vrais. |
| $a \|\| $b | OU (Or) | Vrai si $a OU $b est vrai. |

La raison pour laquelle il existe deux types de "ET" et de "OU" est qu'ils ont des priorités différentes. Voir le paragraphe précédence d'opérateurs .

## 4.6.10 Opérateurs de chaînes

Il y a deux opérateurs de chaînes. Le premier est l'opérateur de concaténation ('.'), qui retourne la concaténation de ses deux arguments. Le second est l'opérateur d'assignation concaténant ( .= ). Reportez−vous à opérateurs d'assignation pour plus de détails.

```
<?php
  $a = "Bonjour ";
  $b = $a . "Monde!";
// $b contient "Bonjour Monde!"
  $a = "Bonjour ";
  $a = $a . "Monde!";
// $a contient "Bonjour Monde!"
?>
```

## 4.6.11 Opérateurs de tableaux

Le seul opérateur en PHP pour les tableaux est l'opérateur + . Cet opérateur ajoute le tableau de droite au tableau de gauche. Les clés communes sont laissées inchangées.

Addition de tableaux

```
<?php
$a = array("a" => "pomme", "b" => "banane");
$b = array("a" =>"poire", "b" => "fraise", "c" => "cerise");

$c = $a + $b;

var_dump($c);
?>



array(3) {
["a"]=>
string(5) "pomme"
["b"]=>
string(6) "banane"
["c"]=>
```

```
string(6) "cerise"
}
```

# 4.7 Les structures de contrôle

Tous les scripts PHP sont une suite d'instructions. Une instruction peut être une assignation, un appel de fonction, une instruction conditionnelle ou bien une instruction qui ne fait rien (une instruction vide). Une instruction se termine habituellement par un point virgule (";"). De plus, plusieurs instructions peuvent être regroupées en bloc, délimité par des accolades (" {} "). Un bloc est considéré comme une instruction. Les différents types d'instruction sont décrits dans ce chapitre.

## 4.7.1 if

L'instruction if est une des plus importantes instructions de tous les langages, PHP inclus. Elle permet l'exécution conditionnelle d'une partie de code. Les fonctionnalités de l'instruction if sont les mêmes en PHP qu'en C :

Instruction if ()

```
<?php
 if (expression)
   commandes
?>
```

Comme nous l'avons vu dans le paragraphe consacré aux expressions , expr est convertie en sa valeur booléenne. Si l'expression expr vaut TRUE , PHP exécutera l'instruction et si elle vaut FALSE , l'instruction sera ignorée. Plus de détails sur les valeurs qui valent FALSE sont disponibles dans la section Conversion en booléen .

L'exemple suivant affiche la phrase a est plus grand
que b si $a est plus grand que $b :

Instruction if () (2)

```
<?php
 if ($a > $b)
   print "a est plus grand que b";
?>
```

Souvent, vous voulez que plusieurs instructions soient exécutées après un branchement conditionnel. Bien évidemment, il n'est pas obligatoire de répéter l'instruction conditionnelle autant de fois que vous avez d'instructions à exécuter. A la place, vous pouvez rassembler toutes les instructions dans un bloc. L'exemple suivant affiche a
est plus grand que b , et assigne la valeur de la variable $a à la variable $b :

Instruction if () et bloc

```
<?php
if ($a > $b) {
    print "a est plus grand que b";
    $b = $a;
}
?>
```

Vous pouvez imbriquer indéfiniment des instructions if les unes dans les autres, ce qui permet une grande flexibilité dans l'exécution d'une partie de code suivant un grand nombre de conditions.

## 4.7.2 else

Souvent, vous voulez exécuter une instruction si une condition est remplie, et une autre instruction si cette condition n'est pas remplie. C'est à cela que sert else . else fonctionne après un if et exécute les instructions correspondantes au cas oú l'expression du if est FALSE . Dans l'exemple suivant, ce bout de code affiche a est plus grand que b si la variable $a est plus grande que la variable $a , et a est plus petit que b sinon:

Instruction if () else

```
<?php
if ($a > $b) {
    print "a est plus grand que b";
} else {
    print "a est plus petit que b";
}
?>
```

Les instructions après le else ne sont exécutées que si l'expression du if est FALSE , et si elle n'est pas suivi par l'expression elseif .

## 4.7.3 elseif

elseif , comme son nom l'indique, est une combinaison de if et de else . Comme l'expression else , il permet d'exécuter une instruction après un if dans le cas oú le "premier" if est évalué comme FALSE . Mais, à la différence de l'expression else , il n'exécutera l'instruction que si l'expression conditionnelle elseif est évaluée comme TRUE . L'exemple suivant affichera a est plus grand que b , a est égal à b ou a est plus petit que b :

Instruction elseif ()

```
<?php
if ($a > $b) {
    print "a est plus grand que b";
} elseif ($a == $b) {
    print "a est égal à b";
} else {
    print "a est plus petit que b";
}
?>
```

Vous pouvez avoir plusieurs elseif qui se suivent les uns après les autres, après un if initial. Le premier elseif qui sera évalué à TRUE sera exécuté. En PHP , vous pouvez aussi écrire " else if " en deux mots et son comportement sera identique à la version en un seul mot. La sémantique des deux expressions est légèrement différente, mais au bout du compte, le résultat sera exactement le même.

L'expression elseif est exécutée seulement si le if précédent et tout autre elseif précédent sont évalués comme FALSE , et que votre elseif est évalué à TRUE .

## 4.7.4 Syntaxe alternative

PHP propose une autre manière de rassembler des instructions à l'intérieur d'un bloc, pour les fonctions de contrôle if , while , for , foreach et switch . Dans chaque cas, le principe est de remplacer l'accolade d'ouverture par deux points (:) et l'accolade de fermeture par, respectivement, endif; , endwhile; , endfor; , ou endswitch; .

Instruction if () alternative

```
<?php if ($a == 5): ?>
A égal 5
<?php endif; ?>
```

Dans l'exemple ci−desssus, le bloc HTML "A = 5" est inclus à l'intérieur d'un if en utilisant cette nouvelle syntaxe. Ce code HTML ne sera affiché que si la variable $a est égale à 5.

Cette autre syntaxe fonctionne aussi avec le else et elseif . L'exemple suivant montre une structure avec un if , un elsif et un else utilisant cette autre syntaxe:

Instruction alternative if () elseif et else

```
<?php
if ($a == 5):
    print "a égale 5";
    print "...";
elseif ($a == 6):
    print "a égale 6";
    print "!!!";
else:
    print "a ne vaut ni 5 ni 6";
endif;
?>
```

Voir aussi while , for , et if pour d'autres exemples.

## 4.7.5 while

La boucle while est le moyen le plus simple d'implémenter une boucle en PHP . Cette boucle se comporte de la même manière qu'en C. L'exemple le plus simple d'une boucle while est le suivant :

Instruction while ()

```
<?php
 while (expression) commandes
?>
```

La signification d'une boucle while est très simple. PHP exécute l'instruction tant que l'expression de la boucle while est évaluée comme TRUE . La valeur de l'expression est vérifiée à chaque début de boucle, et, si la valeur change durant l'exécution de l'instruction, l'exécution ne s'arrêtera qu'à la fin de l'itération (chaque fois que PHP exécute l'instruction, on appelle cela une itération). De temps en temps, si l'expression du while est FALSE avant la première itération, l'instruction ne sera jamais exécutée.

Comme avec le if , vous pouvez regrouper plusieurs instructions dans la même boucle while en les regroupant à l'intérieur de parenthèses ou en utilisant la syntaxe suivante:

Instruction while () avec bloc

```
<?php
while (expression): commandes ... endwhile;
?>
```

Les exemples suivants sont identiques, et affichent tous les nombres de 1 à 10:

Exemple avec while ()

```
<?php
/* exemple 1 */
$i = 1;
while ($i <= 10) {
    print $i++;  /* La valeur affiche est $i avant l'incrémentation
                    (post−incrémentation)  */
}
/* exemple 2 */
$i = 1;
while ($i <= 10):
    print $i;
    $i++;
endwhile;
?>
```

## 4.7.6 do..while

Les boucles do..while ressemblent beaucoup aux boucles while , mais l'expression est testée à la fin de chaque itération plutôt qu'au début. La principale différence par rapport à la boucle while est que la première itération de la boucle do..while est toujours exécutée (l'expression n'est testée qu'à la fin de l'itération), ce qui n'est pas le cas lorsque vous utilisez une boucle while (l'expression est vérifiée au début de chaque itération).

Il n'y a qu'une syntaxe possible pour les boucles do..while :

Instruction do ... while ()

```
<?php
$i = 0;
do {
   print $i;
} while ($i>0);
?>
```

La boucle ci−dessus ne va être exécutée qu'une seule fois, car lorsque l'expression est évaluée, elle vaut FALSE (car la variable $i n'est pas plus grande que 0) et l'exécution de la boucle s'arrête.

Les utilisateurs familiers du C sont habitués à une utilisation différente des boucles do..while , qui permet de stopper l'exécution de la boucle au milieu des instructions, en l'encapsulant dans un do..while (0) la fonction break . Le code suivant montre une utilisation possible :

Instruction while () et break

```
<?php
do {
    if ($i < 5) {
        print "i n'est pas suffisamment grand";
        break;
    }
    $i *= $factor;
    if ($i < $minimum_limit) {
        break;
    }
    print "i est bon";
     ...process i...
} while(0);
?>
```

Ne vous inquiétez pas si vous ne comprenez pas tout correctement. Vous pouvez écrire des scripts très très puissants sans utiliser cette fonctionnalité.

## 4.7.7 for

Les boucles for sont les boucles les plus complexes en PHP . Elles fonctionnent comme les boucles for du langage C. La syntaxe des boucles for est la suivante :

Instruction for()

```
<?php
 for (expr1; expr2; expr3) statement
?>
```

La première expression ( expr1 ) est évaluée (exécutée), quoi qu'il arrive au début de la boucle.

Au début de chaque itération, l'expression expr2 est évaluée. Si l'évaluation vaut TRUE , la boucle continue et l'instruction est exécutée. Si l'évaluation vaut FALSE , l'exécution de la boucle s'arrête.

A la fin de chaque itération, l'expression expr3 est évaluée (exécutée).

Les expressions peuvent éventuellement être laissées vides. Si l'expression expr2 est laissée vide,

cela signifie que c'est une boucle infinie ( PHP considère implicitement qu'elle vaut TRUE , comme en C). Cela n'est pas vraiment très utile, à moins que vous souhaitiez terminer votre boucle par l'instruction conditionnelle break .

Considérons les exemples suivants. Tous affichent les chiffres de 1 à 10:

Exemple de boucles

```
<?php
/* exemple 1 */
for ($i = 1; $i <= 10; $i++) {
    print $i;
}
/* exemple 2 */
for ($i = 1;;$i++) {
    if ($i > 10) {
        break;
    }
    print $i;
}
/* exemple 3 */
$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
}
/* exemple 4 */
for ($i = 1; $i <= 10; print $i, $i++) ;
?>
```

Bien évidemment, le premier exemple est le plus simple de tous (ou peut être le quatrième), mais vous pouvez aussi pensez qu'utiliser une expression vide dans une boucle for peut être utile parfois.

PHP supporte aussi la syntaxe alternative suivante pour les boucles for :

Instruction for() alternative

```
<?php
 for (expr1; expr2; expr3): statement; ...; endfor;
?>
```

Les autres langages ont l'instruction foreach pour accéder aux éléments d'un tableau. PHP 3 ne dispose pas d'une telle fonction; PHP 4 en dispose (voir foreach ). En PHP 3, vous pouvez combiner while avec list et each pour obtenir le même résultat. Reportez−vous aux exemples de la documentation.

## 4.7.8 foreach

PHP 4 (mais pas PHP 3) inclut une commande foreach , comme en Perl ou d'autres langages. C'est un moyen simple de passer en revue un tableau. Il y a deux syntaxes possibles : la seconde est une extension mineure mais pratique de la première:

Instruction foreach()

```
<?php
 foreach(array_expression as $value) commandes
 foreach(array_expression as $key => $value) commandes
?>
```

La première forme passe en revue le tableau array_expression . A chaque itération, la valeur de l'élément courant est assignée à $value et le pointeur interne de tableau est avancé d'un élément (ce qui fait qu'à la prochaine itération, on accédera à l'élément suivant).

La deuxième forme fait exactement la même chose, mais c'est la clé de l'élément courant qui est assigné à la variable $key .

Lorsque foreach démarre, le pointeur interne de fichier est automatiquement ramené au premier élément du tableau. Cela signifie que vous n'aurez pas à faire appel à reset avant foreach .

| Note |
| --- |
| De plus, notez que foreach travaille sur une copie du tableau spécifié, et pas sur le tableau lui−même. Par conséquent, le pointeur de tableau n'est pas modifié, comme il le serait avec le fonction each , et les modifications faites dans le tableau ne seront pas prises en compte dans le tableau original. |

| Note |
| --- |
| foreach n'accepte pas l'opérateur de suppression des erreurs @ . |

Vous pouvez remarquer que les exemples suivants fonctionnent de manière identique :

Instruction foreach() et while()

```
<?php
 reset($arr);
 while (list(, $value) = each ($arr)) {
   echo "Valeur: $value<br>\n";
 }
 foreach ($arr as $value) {
   echo "Valeur: $value<br>\n";
 }
?>
```

Les exemples suivants sont aussi fonctionnellement identiques :

Instruction foreach() et while() (2)

```
<?php
 reset($arr);
 while (list($key, $value) = each ($arr)) {
   echo "Clé: $key; Valeur: $value<br>\n";
 }
 foreach ($arr as $key => $value) {
   echo "Clé: $key; Valeur: $value<br>\n";
 }
?>
```

Voici quelques exemples de plus :

Instruction foreach() et while() (3)

```
<?php
/* exemple foreach 1 : la valeur seulement */

$a = array (1, 2, 3, 17);

foreach ($a as $v) {
   print "Current value of \$a: $v.\n";
}

/* exemple foreach 2 : la valeur et sa clé d'index */

$a = array (1, 2, 3, 17);

$i = 0; /* uniquement pour l'illustration */

foreach($a as $v) {
    print "\$a[$i] => $v.\n";
    $i++;
}

/* exemple foreach 3 : la clé et la valeur */

$a = array (
    "un" => 1,
    "deux" => 2,
    "trois" => 3,
    "dix−sept" => 17
);

foreach($a as $k => $v) {
    print "\$a[$k] => $v.\n";
}

/* exemple foreach 4 : tableaux multi−dimensionnels */

$a[0][0] = "a";
$a[0][1] = "b";
$a[1][0] = "y";
```

```
$a[1][1] = "z";

foreach($a as $v1) {
    foreach ($v1 as $v2) {
        print "$v2\n";
    }
}

/* exemple foreach 5 : tableaux dynamiques */

foreach(array(1, 2, 3, 4, 5) as $v) {
    print "$v\n";
}

?>
```

## 4.7.9 break

L'instruction break permet de sortir d'une structure for , while , foreach ou switch .

break accepte un argument numérique optionnel qui vous indiquera combien de structures emboîtées ont été interrompues.

Instruction break

```
 <?php
$arr = array ('un', 'deux', 'trois', 'quatre', 'stop', 'cinq');
while (list (, $val) = each ($arr)) {
    if ($val == 'stop') {
        break;    /* Vous pourriez aussi utiliser 'break 1;' ici. */
    }
    echo "$val<br>\n";
}

/* Utilisation de l'argument optionnel. */

$i = 0;
while (++$i) {
    switch ($i) {
    case 5:
        echo "A 5<br>\n";
        break 1;  /* Termine uniquement le switch. */
    case 10:
        echo "At 10; quitting<br>\n";
        break 2;  /* Termine le switch et la boucle while. */
    default:
        break;
    }
}

?>
```

## 4.7.10 continue

L'instruction continue est utilisée dans une boucle afin d'éluder les instructions de l'itération courante afin de passer directement à l'itération suivante.

Note

Notez qu'en PHP , la structure switch est considérée comme une boucle par continue .

continue accepte un argument numérique optionnel qui vous indiquera combien de structures emboitées ont été ignorées.

Instruction continue

```
 <?php
while (list ($cle, $valeur) = each ($arr)) {
   if (!($cle % 2)) { // évite les membres impairs
       continue;
   }
   fonction_quelconque($valeur);
}
$i = 0;
while ($i++ < 5) {
    echo "Dehors<br>\n";
```

```
    while (1) {
        echo "  Milieu<br>\n";
        while (1) {
             echo "  Intérieur<br>\n";
             continue 3;
        }
        echo "Ceci n'est jamais atteint.<br>\n";
    }
    echo "Ceci non plus.<br>\n";
}
?>
```

Oublier le point virgule après continue peut porter à confusion. Voici un exemple de ce que vous ne devez pas faire :

```
<?php
 for ($i = 0; $i < 5; ++$i) {
     if ($i == 2)
         continue
     print "$i\n";
 }
?>
```

On peut s'attendre à ce que le résultat soit :

```
0
1
3
4
```

mais ce script affichera :

```
2
```

car la valeur de retour de l'appel à print est int(1) , et cela se comportera alors comme si on avait fournit l'argument optionnel mentionné plus haut.

## 4.7.11 switch

L'instruction switch équivaut à une série d'instructions if . En de nombreuses occasions, vous aurez besoin de comparer la même variable (ou expression) avec un grand nombre de valeurs différentes, et d'exécuter différentes parties de code suivant la valeur à laquelle elle est égale. C'est exactement à cela que sert l'instruction switch .

| Note |
| --- |
| Notez que contrairement à d'autres langages, la structure continue s'applique aux structures switch et de comporte différemment de break . Si vous avez un switch dans une boucle, et que vous souhaitez continuer jusqu'à la prochaine itération de la boucle extérieure, vous vous devez utiliser continue 2 . |

Les deux exemples suivants sont deux manières différentes d'écrire la même chose, l'une en utilisant une séries de if , et l'autre en utilisant l'instruction switch :

Instruction switch()

```
<?php
if ($i == 0) {
    print "i égale 0";
}
if ($i == 1) {
    print "i égale 1";
}
if ($i == 2) {
    print "i égale 2";
}
switch ($i) {
    case 0:
        print "i égale 0";
```

```
        break;
    case 1:
        print "i égale 1";
        break;
    case 2:
        print "i égale 2";
        break;
}
?>
```

Il est important de comprendre que l'instruction switch exécute chacune des clauses dans l'ordre. L'instruction switch est exécutée ligne par ligne. Au début, aucun code n'est exécuté. Seulement lorsqu'un case est vérifié, PHP exécute alors les instructions correspondantes. PHP continue d'exécuter les instructions jusqu'à la fin du bloc d'instructions du switch , ou bien dès qu'il trouve l'instruction break . Si vous ne pouvez pas utiliser l'instruction break à la fin de l'instruction case , PHP continuera à exécuter toutes les instructions qui suivent. Par exemple :

Instruction switch et fall through

```
 <?php
switch ($i) {
    case 0:
        print "i égale 0";
    case 1:
        print "i égale 1";
    case 2:
        print "i égale 2";
}
?>
```

Dans cet exemple, si $i est égal à 0, PHP va exécuter quand même toutes les instructions qui suivent. Si $i est égal à 1, PHP exécutera les deux dernières instructions. Et seulement si $i est égal à 2, vous obtiendrez le résultat escompté, c'est−à−dire, l'affiche de "i égal 2". Donc, l'important est de ne pas oublier l'instruction break (même s'il est possible que vous l'omettiez dans certaines circonstances).

Dans une commande switch , une condition n'est évaluée qu'une fois, et le résultat est comparé à chaque case . Dans une structure elseif , les conditions sont évaluées à chaque comparaison. Si votre condition est plus compliquée qu'une simple comparaison, ou bien fait partie d'une boucle, switch sera plus rapide.

La liste de commandes d'un case peut être vide, auquel cas PHP utilisera la liste de commandes du cas suivant.

Instruction switch() et case vide

```
 <?php
switch ($i) {
    case 0:
    case 1:
    case 2:
        print "i est plus petit que 3 mais n'est pas négatif";
        break;
    case 3:
        print "i égale 3";
}
?>
```

Un case spécial est default . Ce cas est utilisé lorsque tous les case ont échoués. Il doit être le dernier cas listé. Par exemple :

Instruction switch() et default

```
 <?php
switch ($i) {
    case 0:
        print "i égale 0";
        break;
    case 1:
        print "i égale 1";
        break;
    case 2:
        print "i égale  2";
        break;
    default:
```

```
        print "i n'est ni égal à 2, ni à 1, ni à 0.";
}
?>
```

Une autre chose à mentionner est que l'instruction case peut être une expression à de type scalaire, c'est−à−dire nombre entier, nombre à virgule flottante et chaîne de caractères. Les tableaux sont sans intérêt dans ce contexte−là.

La syntaxe alternative pour cette structure de contrôle est la suivante : pour plus d'informations, voir syntaxes alternatives ).

Instruction switch() alternative

```
<?php
 switch ($i):
   case 0:
       print "i égale 0";
       break;
   case 1:
       print "i égale 1";
       break;
   case 2:
       print "i égale 2";
       break;
   default:
       print "i n'est ni égal à 2, ni à 1, ni à 0";
 endswitch;
?>
```

## 4.7.12 declare

L'élément de langage declare sert à ajouter des directives d'exécutions dans un bloc de code. La syntaxe de declare est similaire à la syntaxe des autres fonctions de contrôle :

Instruction declare

```
<?php
 declare (directive) statement
?>
```

L'expression directive permet de contrôler l'intervention du bloc declare . Actuellement, une seule directive est reconnue : la directive ticks (Voir plus bas pour plus de détails) sur les ticks ).

L'expression statement du bloc de declare sera exécutée. Comment elle sera exécutée, et quels effets cela aura dépend de la directive utilisée dans le bloc directive .

### 4.7.12.1 Ticks

Un tick est un événement qui intervient toutes les N commandes bas niveau, exécutées par l'analyseur dans le bloc de declare . La valeur de N est spécifiée avec la syntaxe ticks= N dans le bloc de directive declare .

Un événement qui intervient à chaque tick est spécifié avec la fonction register_tick_function . Reportez vous à l'exemple ci−dessous pour plus de détails. Notez que plus d'un événement peut intervenir par tick.

Profiler une section de code PHP

```
<?php
// Un fonction qui enregistre l'heure à laquelle elle est appelée
 function profile($dump = FALSE){
   static $profile;
   // Retourne les horaires stockés dans le profile, et l'efface
   if ($dump) {
       $temp = $profile;
       unset ($profile);
       return ($temp);
   }
```

```
    $profile[] = microtime ();
  }

// Enregistre un gestionnaire de tick
  register_tick_function("profile");

// Initialise la fonction avant le bloc de déclaration
  profile();

// Exécute un bloc de code, et appele un tick toutes les deux secondes
  declare (ticks=2) {
    for ($x = 1; $x < 50; ++$x) {
        echo similar_text(md5($x), md5($x*$x)), "<br>";
    }
  }

// Affiche les données de la variable $profile
  print_r(profile(TRUE));
?>
```

Cet exemple profile le code PHP dans le bloc de déclaration, et enregistre l'heure de chaque commande bas niveau. Cette information peut être réutilisée pour débusquer les segments de code lents. Vous pouvez implémenter d'autres méthodes, mais les ticks sont plus rapides et plus efficaces.

Les ticks sont bien pratiques pour débugger, implémenter un multi−tâches simple, des entrées sorties en tâche de fond, ou bien d'autres choses, avec PHP .

Voir aussi register_tick_function et unregister_tick_function .

## 4.7.13 return

Si appelée depuis une fonction, la commande return termine immédiatement la fonction, et retourne l'argument qui lui est passé. return interromp aussi l'exécution de commande eval ou de scripts.

Si appelée depuis l'environnement global, l'exécution du script est interrompue. Si le script courant était include ou require , alors le contrôle est rendu au script appelant, et la valeur retournée sera utilisée comme résultat de la fonction include . Si return est appelée depuis le script principal, alors l'exécution du script s'arrête. Si le script courant est auto_prepend_file ou auto_append_file dans le fichier php.ini , alors l'exécution du script s'arrête.

Pour plus d'informations, voyez retourner des valeurs .

| Note |
| --- |
| Notez que puisque return est une structure de langage et non pas une fonction, les parenthèses **ne sont pas** requises : en fait, il est classique de les ignorer, même si cela ne change rien. |

## 4.7.14 require

require inclut et exécute un fichier PHP .

La commande require se remplace elle−même par le contenu du fichier spécifié, tel que décrit dans la documentation de la fonction include .

require et include sont identiques, sauf dans leur façon de gérer les erreurs. include produite une Alerte (warning) tandis que require génère une erreur fatale . En d'autres termes, n'hésitez pas à utiliser require si vous voulez qu'un fichier d'inclusion manquant interrompe votre script. include ne se comporte pas de cette façon, et le script continuera son exécution. Assurez vous d'avoir bien configuré le include_path aussi.

Exemples avec require

```
<?php
 require 'prepend.php';

 require $somefile;

 require ('unfichier.txt');
?>
```

Voyez la documentation de la fonction include pour plus d'exemples.

Note

Avant PHP 4.0.2, ceci s'appliquait : require tentait de lire le fichier cible, même si les lignes n'étaient pas utilisées. Une condition if n'avait aucun effet sur require . Cependant, si la ligne du require n'est jamais utilisée, le code inclus ne le sera pas non plus. De même, les boucles n'ont pas d'effet sur le comportement de require . Bien que le code contenu dans le fichier subit la boucle, la fonction elle même require n'est appelée qu'une fois.

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Attention

La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

Voir aussi include , require_once , include_once , eval , file , readfile , virtual et include_path .

## 4.7.15 include

La fonction include inclut et exécute le fichier spécifié en argument.

Cette documentation s'applique aussi à la fonction require . Les deux structures de langage sont identiques, hormis dans leur gestion des erreurs. include produit une Alerte (warning) tandis que require génère une erreur fatale . En d'autres termes, n'hésitez pas à utiliser require si vous voulez qu'un fichier d'inclusion manquant interrompe votre script. include ne se comporte pas de cette façon, et le script continuera son exécution. Assurez−vous d'avoir bien configuré le include_path aussi.

Lorsqu'un fichier est inclus, le code qu'il contient hérite des variables qui existent dans ce contexte . Toute variable accessible à la ligne qui réalise l'exécution est aussi accessible dans le fichier inclus.

Exemple avec include

```
 vars.php
<?php

$couleur = 'verte';
$fruit = 'pomme';

?>

test.php
<?php

echo "Une $couleur $fruit"; // Une

include 'vars.php';

echo "Une $coleur $fruit"; // Une verte pomme

 ?>
```

Si l'inclusion intervient à l'intérieure d'une fonction, le code inclus sera alors considéré comme faisant partie de la fonction. Cela modifie donc le contexte de variables accessibles.

Inclusion de fichiers dans une fonction

```
<?php

function foo() {
    global $couleur;

    include 'vars.php';

    echo "Une $couleur $fruit";
}

/* vars.php est dans le contexte de foo()    *
 * donc $fruit n'est pas dispnibles hors de  *
 * cette fonction. $couleur l'est, car c'est *
 * une variable globale                      */

foo();                      // Une verte pomme
echo "Une $couleur $fruit"; // Une verte

?>
```

Il est important de noter que lorsqu'un fichier est include ou require , les erreurs d'analyse apparaîtront en HTML tout au début du fichier, et l'analyse du fichier parent ne sera pas interrompue. Pour cette raison, le code qui est dans le fichier doit être placé entre les balises habituelles de PHP .

Si les Gestionnaires d'URL sont activés dans PHP (ce qui est le cas par défaut), vous pouvez localiser le fichier avec une URL (via HTTP ou bien avec un gestionnaire adapté : voir Liste des protocoles supportés pour une liste des protocoles), au lieu d'un simple chemin local. Si le serveur distant interprête le fichier comme du code PHP , des variables peuvent être transmises au serveur distant via l'URL et la méthode GET. Ce n'est pas, à strictement parler, la même chose que d'hériter du contexte de variable. Le fichier inclus est en fait un script exécuté à distance, et son résultat est inclus dans le code courant.

Attention

La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

Utiliser include via HTTP

```
<?php

/* Cet exemple suppose que www.exemple.com est configur pour traiter
 * les fichiers .php et non pas les fichiers .txt. De plus,
 * 'Work' signifie ici que les variables
 * $foo et $bar sont disponibles dans le fichier inclus
 */

// Ne fonctionne pas : file.txt n'a pas été traité par www.exemple.com comme du PHP
include 'http://www.example.com/file.txt?foo=1&bar=2';

// Ne fonctionne pas : le script cherche un fichier nommé
 // 'file.php?foo=1&bar=2' sur le système local
include 'file.php?foo=1&bar=2';

// Réussi
include 'http://www.example.com/file.php?foo=1&bar=2';

$foo = 1;
$bar = 2;
include 'file.txt';  // OK.
include 'file.php';  // OK.

?>
```

Voir aussi travailler avec les fichiers distants , fopen et file pour des informations reliées.

Comme include et require sont des structures du langage particulières, vous devez les placer dans un bloc s'ils sont placés dans un bloc conditionnel.

include et les conditions

```
<?php

// Ceci est FAUX et échouera
if ($condition)
```

```
    include $file;
else
    include $other;


// Ceci est CORRECT.
if ($condition) {
    include $file;
} else {
    include $other;
}

?>
```

Gestion du retour : il est possible d'exécuter une commande return dans un fichier inclus pour en terminer le traitement et retourner au fichier appelant. De plus, il est possible de retourner des valeurs des fichiers inclus. vous pouvez prendre et traiter la valeur retournée par la fonction, comme toute autre fonction.

| Note |
| --- |
| En PHP 3, le return n'est pas utilisable à moins que ce ne soit dans un bloc de fonction, auquel cas le return s'applique à une fonction et non pas au fichier. |

include et return

```
 return.php
<?php

$var = 'PHP';

return $var;

?>

noreturn.php
<?php

$var = 'PHP';

?>

testreturns.php
<?php

$foo = include 'return.php';

echo $foo; // prints 'PHP'

$bar = include 'noreturn.php';

echo $bar; // prints 1

?>
```

$bar a la valeur de 1 car l'inclusion était réussie. Notez la différence entre les deux exemples ci−dessus. Le premier utilise la commande return dans le fichier inclus, alors que le second ne le fait pas. Un autre moyen d'inclure des fichiers dans des variables est d'utiliser la fonction fopen ou file ou d'utiliser include avec les fonctions de contrôle de sortie .

| Note |
| --- |
| Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions . |

Voir aussi require , require_once , include_once , readfile , virtual et include_path .

## 4.7.16 require_once

La commande require_once se remplace elle−même par le fichier spécifié, un peu comme les commandes de préprocesseur C #include , et ressemble sur ce point à require . La principale différence est qu'avec require_once , vous êtes assurés que ce code ne sera ajouté qu'une seule fois, évitant de ce fait les redéfinitions de variables ou de fonctions, génératrices d'alertes.

require_once doit être utilisée dans les cas où le même fichier risque d'être inclus et évalué plusieurs fois durant l'exécution d'un script. Vous serez alors sûrs qu'il ne sera inclut qu'une fois et vous éviterez les problèmes de re−déclarations de fonctions ou autres.

Pour des exemples avec require_once et include_once , reportez−vous au code PEAR inclus dans les dernières distributions PHP .

| Note |
| --- |
| require_once a été ajouté en PHP 4.0.1pl2 |

| Note |
| --- |
| Attention, require_once et include_once sont insensibles à la casse sous Windows. |

require_once est insensible à la casse sur Windows

```
 require_once("a.php"); // ceci inclut le fichier a.php
require_once("A.php"); // ceci inclut encore le fichier a.php sous Windows!
```

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

Voir aussi require , include , include_once , get_required_files , get_included_files , readfile et virtual .

## 4.7.17 include_once

La commande include_once inclut et évalue le fichier spécifié durant l'exécution du script. Le comportement est similaire à include , mais la différence est que si le code a déjà été inclus, il ne le sera pas une seconde fois.

La fonction include_once est utilisée de préférence lorsque le fichier doit être inclus ou évalué plusieurs fois dans un script, ou bien lorsque vous voulez être sûr qu'il ne sera inclus qu'une seule fois, pour éviter des redéfinitions de fonction.

Pour plus d'exemples avec require_once et include_once , jetez un oeil dans le code de PEAR inclus dans la dernière distribution de PHP .

| Note |
| --- |
| include_once a été ajouté en PHP 4.0.1pl2 |

| Note |
| --- |
| Attention, require_once et include_once sont insensibles à la casse sous Windows. |

include_once est insensible à la casse sous Windows

```
 include_once("a.php"); // ceci inclut le fichier a.php
include_once("A.php"); // ceci inclut encore le fichier a.php sous Windows!
```

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

Voir aussi require , include , require_once , get_required_files , get_included_files , readfile et virtual .

# 4.8 Les fonctions

## 4.8.1 Les fonctions

Une fonction peut être définie en utilisant la syntaxe suivante :

Définition d'une fonction

```
 <?php
function foo ($arg_1, $arg_2, ..., $arg_n) {
    echo "Exemple de fonction.\n";
    return $retval;
}
?>
```

Tout code PHP , correct syntaxiquement, peut apparaître dans une fonction et dans une définition de classe .

En PHP 3, les fonctions doivent être définies avant qu'elles ne soient utilisées. Ce n'est plus le cas en PHP 4.

Lorsqu'une fonction est définie de manière conditionnelle, comme dans les exemples ci−dessous, leur définition doit **précéder** leur utilisation.

Fonctions conditionnelles

```
 <?php

$makefoo = true;

/* Impossible d'appeler foo() ici,
   car cette fonction n'existe pas.
   Mais nous pouvons utiliser bar() */

bar();

if ($makefoo) {
  function foo ()
  {
    echo "Je n'existe pas tant que le programme n'est pas passé ici.\n";
  }
}

/* Maitenant, nous pouvons appeler foo()
   car $makefoo est maintenant vrai */

if ($makefoo) foo();

function bar() {
{
  echo "J'existe dès de début du programme.\n";
}

?>
```

Fonctions dans une autre fonctions

```
 <?php
function foo()
 {
  function bar()
   {
    echo "Je n'existe pas tant que foo() n'est pas appelé.\n";
  }
}

/* Impossible d'appeler bar() ici
   car il n'existe pas. */

foo();

/* Maintenant, nous pouvons appeler bar(),
   car l'utilisation de foo() l'a rendu
   accessible. */

bar();

?>
```

PHP ne supporte pas le surchargement de fonction, ni la destruction ou la redéfinition de fonctions déjà déclarées.

PHP 3 ne supporte pas un nombre variable d'arguments (voir valeurs par défaut d'arguments pour plus d'informations). PHP 4 supporte les deux : voir liste variable d'arguments de fonction et les fonctions de références que sont func_num_args , func_get_arg , et func_get_args pour plus d'informations.

## 4.8.2 Les arguments de fonction

Des informations peuvent être passées à une fonction en utilisant un tableau d'arguments, dont chaque élément est séparé par une virgule. Un élément peut être une variable ou une constante.

PHP supporte le passage d'arguments par valeur (méthode par défaut), par référence . Les listes variables d'arguments sont supportées par PHP 4 et les versions plus récentes. Voir liste variable d'arguments de fonction et les fonctions utiles que sont func_num_args , func_get_arg , et func_get_args . Fonctionnellement, on peut arriver au même résultat en passant un tableau comme argument :

Nombre variable d'argument sous forme de tableau

```
<?php
function takes_array($input) {
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
?>
```

### 4.8.2.1 Passage d'arguments par référence

Par défaut, les arguments sont passés à la fonction par valeur (donc vous pouvez changer la valeur d'un argument dans la fonction, cela ne change pas sa valeur à l'extérieur de la fonction). Si vous voulez que vos fonctions puissent changer la valeur des arguments, vous devez passer ces arguments par référence.

Si vous voulez qu'un argument soit toujours passé par référence, vous pouvez ajouter un ' & ' devant l'argument dans la déclaration de la fonction :

Passage d'arguments par référence

```
<?php
function add_some_extra(&$string) {
    $string .= ', et un peu plus.';
}
$str = 'Ceci est une chaîne';
add_some_extra($str);
echo $str;    // affiche 'Ceci est une chaîne, et un peu plus.'
?>
```

Si vous souhaitez passer une variable par référence à une fonction mais de manière ponctuelle, vous pouvez ajouter un '&' devant l'argument dans l'appel de la fonction:

Passage de valeur par référence, ponctuellement

```
<?php
function foo ($bar) {
    $bar .= ', et un peu plus.';
}
$str = Ceci est une chaîne';
foo ($str);
echo $str;    // affiche 'Ceci est une chaîne'
foo (&$str);
echo $str;    // affiche 'Ceci est une chaîne, et un peu plus.'
?>
```

#### 4.8.2.2 Valeur par défaut des arguments

Vous pouvez définir comme en C++ des valeurs par défaut pour les arguments de type scalaire :

```
Valeur par défaut des arguments de fonctions
<?php
function servir_apero ($type = "ricard") {
    return "Servir un verre de $type.\n";
}
echo servir_apero();
echo servir_apero("whisky");
?>
```

La fonction ci−dessus affichera :
Servir un verre de ricard.
Servir un verre de whisky.

La valeur par défaut d'un argument doit obligatoirement être une constante, et ne peut être ni une variable, ni un membre de classe.

Il est à noter que si vous utilisez des arguments avec valeur par défaut avec d'autres sans valeur par défaut, les premiers doivent être placés à la suite de tous les paramètres sans valeur par défaut. Sinon, cela ne fonctionnera pas. Considérons le code suivant :

```
Les arguments à valeur par défaut doivent être en premier : erreur
<?php
function faireunyaourt ($type = "acidophilus", $flavour) {
    return "Préparer un bol de $type $flavour.\n";
}
echo faireunyaourt ("framboise");   // ne fonctionne pas comme voulu
?>
```

L'affiche du code ci−dessus est le suivant :
Warning: Missing argument 2 in call to faireunyaourt() in
/usr/local/etc/httpd/htdocs/PHP 3test/functest.html on line 41
Préparer un bol de framboise.

Maintenant comparons l'exemple précédent avec l'exemple suivant :

```
Les arguments à valeur par défaut doivent être en premier : valide
<?php
function faireunyaourt ($flavour, $type = "acidophilus") {
    return "Préparer un bol de $type $flavour.\n";
}
echo faireunyaourt ("framboise");   // fonctionne comme voulu
?>
```

L'affichage de cet exemple est le suivant :
Préparer un bol de acidophilus framboise.

#### 4.8.2.3 Nombre d'arguments variable

PHP 4 supporte les fonctions à nombre d'arguments variable. C'est très simple à utiliser, avec les fonctions func_num_args , func_get_arg , et func_get_args .

Aucune syntaxe particulière n'est nécessaire, et la liste d'argument doit toujours être fournie explicitement avec la définition de la fonction, et se comportera normalement.

## 4.8.3 Les valeurs de retour

Les valeurs sont renvoyées en utilisant une instruction de retour optionnelle. Tous les types de variables peuvent être renvoyés, tableaux et objets compris.

Retourner des valeurs de fonction

```
<?php
function carre ($num) {
    return $num * $num;
}
echo carre (4);   // affiche '16'.
?>
```

Vous ne pouvez pas renvoyer plusieurs valeurs en même temps, mais vous pouvez obtenir le même résultat en renvoyant un tableau.

Retourner plusieurs valeurs d'une fonction

```
<?php
function petit_nombre() {
    return array (0, 1, 2);
}
list ($zero, $one, $two) = petit_nombre();
?>
```

Pour retourner une référence d'une fonction, utilisez l'opérateur & aussi bien dans la déclaration de la fonction que dans l'assignation de la valeur de retour.

Retourner une référence d'une fonction

```
<?php
function &retourne_reference() {
    return $uneref;
}
$newref =& retourne_reference();
?>
```

## 4.8.4 Variables fonctions

PHP supporte le concept de variables fonctions. Cela signifie que si le nom d'une variable est suivi de parenthèses, PHP recherchera une fonction de même nom, et essaiera de l'exécuter. Cela peut servir, entre autre, pour faire des fonctions callback, des tables de fonctions...

Les variables fonctions ne peuvent pas fonctionner avec les éléments de langage comme les echo , unset , isset , empty , etc... C'est une des différences majeures entre les fonctions PHP et les éléments de langage.

Exemple de variable fonction

```
<?php
function foo() {
    echo "dans foo()<br>\n";
}
function bar( $arg = '' ) {
    echo "Dans bar(); l'argument était '$arg'.<br>\n";
}
$func = 'foo';
$func();
$func = 'bar';
$func( 'test' );
?>
```

Voir aussi call_user_func , les variables variables et function_exists .

### 4.8.5 Fonctions internes

PHP dispose de nombreuses fonctions et structures standards. Il y a aussi des fonctions qui requièrent des extensions spécifiques de PHP , sans lesquelles vous obtiendrez l'erreur fatale undefined function . Par exemple, pour utiliser les fonctions d'images , telles que imagecreatetruecolor , vous aurez besoin du support de GD dans PHP . Ou bien, pour utiliser mysql_connect , vous aurez besoin de l'extension MySQL . Il y a des fonctions de base qui sont inclues dans toutes les versions de PHP , telles que les fonctions de chaînes de caractères et les fonctions de variables . Utilisez phpinfo ou get_loaded_extensions pour savoir quelles sont les extensions qui sont compilées avec votre PHP . Notez aussi que de nombreuses extensions sont activées par défaut, et que le manuel PHP est compartimenté par extension. Voyez les chapitres de configuration , installation ainsi que les détails particuliers à chaque extension, pour savoir les mettre en place.

Lire et comprendre le prototype d'une fonction est décrit dans l'annexe Comment lire la définition d'une fonction (prototype) . Il est important de comprendre ce qu'une fonction retourne, ou si une fonction travaille directement sur la valeur des paramètres fournis. Par exemple, str_replace va retourne une chaîne modifiée, tandis que usort travaille directement sur la variable passée en paramètre. Chaque page du manuel a des informations spécifiques sur chaque fonction, comme le nombre de paramètres, les évolutions de spécifications, les valeurs retournées en cas de succès ou d'échec, et la disponibilité en fonction des versions. Bien connaître ces différences, parfois subtiles, est crucial pour bien coder en PHP .

Voir aussi function_exists , l'index des fonctions , get_extension_funcs et dl .

## 4.9 Les classes et les objets

### 4.9.1 Les classes : class

Une classe est une collection de variables et de fonctions qui fonctionnent avec ces variables. Une classe est définie en utilisant la syntaxe suivante :

Les classes en PHP

```
 <?php
class Caddie {
// Eléments de notre panier
    var $items;

// Ajout de $num articles de type $artnr au panier
    function add_item ($artnr, $num) {
        $this->items[$artnr] += $num;
    }

// Suppression de $num articles du type $artnr du panier
    function remove_item ($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return
 <font face="courrier">TRUE</font>
;
        } else {
            return FALSE;
        }
    }
}
?>
```

L'exemple ci−dessus définit la classe Caddie qui est composée d'un tableau associatif contenant les articles du panier et de deux fonctions, une pour ajouter et une pour enlever des éléments au panier.

Attention

Les notes suivantes ne sont valable que pour PHP 4.

Le nom stdClass est utilisé en interne par Zend et ne doit pas être utilisé. Vous ne pouvez pas nommer une classe stdClass en PHP .

Les noms de fonctions __sleep et __wakeup sont magiques en PHP . Vous ne pouvez pas utiliser ces noms de fonctions dans vos classes, à moins que vous ne souhaitiez utiliser la magie qui y est associée.

PHP se réserve l'usage de tous les noms de fonctions commençants par __ , pour sa propre magie. Il est vivement recommandé de ne pas utiliser des noms de fonctions commençants par __ , à moins que vous ne souhaitiez utiliser la magie qui y est associée.

Note

En PHP 4, seuls les initialiseurs constants pour les variables var sont autorisés. Utilisez les constructeurs pour les initialisations variables, ou utilisant des expressions.

Initialisation de membres de classe

```
<?php
/* Aucune de ces syntaxes ne fonctionnera en PHP 4 */
class Caddie {
    var $date_du_jour = date("d/m/Y");
    var $name = $firstname;
    var $owner = 'Fred ' . 'Jones';
    var $items = array("DVD", "Télé","Magnétoscope");
}
/* Voici comment cela doit se faire désormais. */
class Caddie {
    var $date_du_jour;
    var $name;
    var $owner;
    var $items;
    function Caddie() {
        $this->date_du_jour = date("d/m/Y");
        $this->name = $GLOBALS['firstname'];
        /* etc... */
    }
}
?>
```

Les classes forment un type de variable. Pour créer une variable du type désiré, vous devez utiliser l'opérateur new.

Création d'un objet de classe

```
<?php
$cart = new Caddie;
$cart->add_item("10", 1);
$another_cart = new Caddie;
$another_cart->add_item("0815", 3);
?>
```

L'instruction ci−dessus crée l'objet $cart de la classe Caddie . La fonction add_idem() est appelée afin d'ajouter l'article numéro 10 dans le panier. 3 articles numéro 0815 sont ajoutés au caddie $another_cart .

$cart et $another_cart disposent des fonctions add_item() , remove_item() et de la variable items . Ce sont des fonctions et variables distinctes. Vous pouvez vous représenter les objets comme des dossiers sur votre disque dur. Vous pouvez avoir deux fichiers lisez−moi.txt sur votre disque dur, tant qu'ils ne sont pas dans le même répertoire. De même que vous devez alors taper le chemin complet jusqu'au fichier, vous devez spécifier le nom complet de la méthode avant de l'employer : en termes PHP , le dossier racine est l'espace de nom global, et le séparateur de dossier est −> . Par exemple, les noms $cart−>items et $another_cart−>items représentent deux variables distinctes. Notez que le nom de la variable est alors $cart−>items, et non pas $cart−>$items : il n'y a qu'un seul signe $ dans un nom de variable.

Accès aux membres d'une classe

```
<?php
// correct, le signe $ est unique
$cart->items  = array("10" => 1);
// incorrect, car $cart->$items devient $cart->""
$cart->$items = array("10" => 1);
// correct, mais risque de ne pas se comporter comme prévu
// $cart->$myvar devient $cart->items
$myvar = 'items';
$cart->$myvar = array("10" => 1);
?>
```

A l'intérieur d'une définition de classe, vous ne savez pas le nom de la variable à partir duquel l'objet sera accessible dans le script. On ne peut prévoir que l'objet créé sera affecté à la variable $cart ou $another_cart. Donc, vous ne pouvez pas utiliser la syntaxe $cart−>items. Mais pour pouvoir accéder à aux méthodes et membres d'un objet, vous pouvez utiliser la variable spéciale $this, qui peut s'interpréter comme 'moi−même', ou bien 'l'objet courant'. Par exemple, '$this−>items[$artnr] += $num;' peut se lire comme 'ajouter $num au compteur $artnr de mon propre tableau de compteur' ou bien 'ajouter $num au compteur $artnr du tableau de compteurs de l'objet courant'.

## 4.9.2 extends : héritage

Souvent, vous aurez besoin d'une classe avec des méthodes et fonctions similaires à une autre classe. En fait, il est bon de définir des classes génériques, qui pourront être réutilisées et adaptées à tous vos projets. Pour faciliter cela, une classe peut être une extension d'une autre classe. La classe dérivée hérite alors de toutes les méthodes et variables de la classe de base (cet héritage a de bien que personne ne meurt pour en profiter), mais peut définir ses propres fonctions et variables, qui s'ajouteront. Une classe ne peut hériter que d'une seule autre classe, et l'héritage multiple n'est pas supporté. Les héritages se font avec le mot clé ' extends '.

Héritage de classes

```
<?php
class Caddie_nomme extends Caddie {
    var $owner;
    function set_owner ($name) {
        $this->owner = $name;
    }
}
?>
```

L'exemple ci−dessus définit la classe Caddie_nomme qui possède les mêmes variables que la classe Caddie et la variable $owner en plus, ainsi que la fonction set_owner() . Vous créez un panier nominatif de la même manière que précédemment, et vous pouvez alors affecter un nom au panier ou en connaître le nom. Vous pouvez de toutes les façons utiliser les mêmes fonctions que sur un panier classique.

Héritage de classes (2)

```
<?php
$ncart = new Caddie_nomme;    // Création d'un panier nominatif
$ncart->set_owner ("kris"); // Affectation du nom du panier
print $ncart->owner;        // Affichage du nom du panier
$ncart->add_item ("10", 1); // (héritage des fonctions de la classe père)
?>
```

## 4.9.3 Constructor : constructeur

Attention

En PHP 3 et PHP 4, les constructeurs se comportent différemment. La sémantique de PHP 4 est fortement recommandée.

Le constructeur est la fonction qui est appelée automatiquement par la classe lorsque vous créez une nouvelle instance d'une classe a l'aide de l'opérateur new . La fonction constructeur a le même nom que la classe. En PHP 3, une fonction devient le constructeur si elle porte le même nom que la classe. En PHP 4, une fonction devient un constructeur si elle porte le même nom que la classe dans laquelle elle est définie. La différence est subtile, mais cruciale.

Constructeur de classe

```
<?php
class Auto_Caddie extends Caddie {
    function Auto_Caddie () {
        $this->add_item ("10", 1);
    }
}
// Cette syntaxe est valable en PHP 3 et 4
?>
```

L'exemple ci−dessus définit la classe Auto_Caddie qui hérite de la classe Caddie et définit le constructeur de la classe. Ce dernier initialise le panier avec 1 article de type numéro 10 dès que l'instruction "new" est appelée. La fonction constructeur peut prendre ou non des paramètres optionnels, ce qui la rend beaucoup plus pratique. Pour pouvoir utiliser cette classe sans paramètre, tous les paramètres du constructeurs devraient être optionnels, en fournissant une valeur par défaut, comme ci−dessous.

Constructeur de classe à argument optionnel

```
<?php
// Cette syntaxe est valable en PHP 3 et 4
class Constructor_Cart extends Cart {
    function Constructor_Cart ($item = "10", $num = 1) {
        $this->add_item ($item, $num);
    }
}
// Création du caddie
$default_cart   = new Constructor_Cart;
// Création d'un vrai caddie
$different_cart = new Constructor_Cart ("20", 17);
?>
```

Attention

En PHP 3, les classes dérivées et les constructeurs ont un certains nombre de limitations. Les exemples suivants doivent être lus avec beaucoup d'attention pour comprendre ces limitations.

Constructeur de classe en PHP 4

```
<?php
class A {
  function A() {
    echo "Je suis le constructeur de A.<br>\n";
  }
}
class B extends A {
  function C() {
    "Je suis une fonction standard.<br>\n";
  }
}
// Aucun constructeur n'est appelé en PHP 3!!
$b = new B;
?>
```

En PHP 3, aucun constructeur ne sera appelé dans l'exemple ci−dessus. La règle en PHP 3 est : 'Un constructeur est une méthode qui a le même nom que la classe'. Le nom de la classe est B, et il n'y a pas de fonction qui s'appelle B() dans la classe B. Rien ne se passe.

Ceci est corrigé en PHP 4, avec l'introduction d'une nouvelle règle : Si une classe n'a pas de constructeur, le constructeur de la classe de basse est appelé, s'il existe. L'exemple ci−dessus affichera ' Je suis le constructeur de A.<br> ' en PHP 4.

Constructeur de classe en PHP 4

```
<?php
class A {
  function A() {
    echo "Je suis le constructeur de A.<br>\n";
  }
```

```
  function B() {
    echo "Je suis une fonction standard appelée B dans la classe A.<br>\n";
    echo "Je ne suis pas le constructeur de A.<br>\n";
  }
}

class B extends A {
  function C() {
    echo "Je suis une fonction standard.<br>\n";
  }
}
// Cette syntaxe va appeler B() comme constructeur.
$b = new B;
?>
```

En PHP 3, la fonction B() de la classe A va soudainement devenir le constructeur de la classe B, bien qu'il n'ai pas été prévu pour. La règle de PHP 3 est 'Un constructeur est une méthode qui a le même nom que la classe'. PHP 3 ne se soucie guère si la fonction est définie dans la classe B ou si elle a été héritée.

Ceci est corrigé en PHP 4, avec l'introduction d'une nouvelle règle : 'Un constructeur est une méthode de même nom que la classe, définie dans la classe elle−même'. Donc, en PHP 4, la classe B n'a pas de constructeur par elle−même, et le constructeur de la classe A aura été appelé, affichant : ' Je suis le constructeur de A.<br> '.

| Attention |
| --- |
| Ni PHP 3 ni PHP 4 n'appelle automatiquement le constructeur de la classe supérieure depuis le constructeur de la classe dérivée. Il est de votre responsabilité de propager l'appel des constructeurs. |

| Note |
| --- |
| Il n'y a pas de destructeurs en PHP 3 et PHP 4. Vous pouvez utiliser la fonction register_shutdown_function à la place, pour simuler un destructeur. |

Les destructeurs sont des fonctions qui sont appelées lorsqu'un objet est détruit, soit avec la fonction unset soit par simple sortie d'une fonction (cas des variables locales). Il n'y a pas de destructeurs en PHP .

## 4.9.4 Opérateur ::

| Attention |
| --- |
| La documentation suivante n'est valable que pour PHP 4 et plus récent. |

Parfois, il est pratique de faire référence aux fonctions est variables d'une classe de base, ou bien d'utiliser des méthodes de classes qui n'ont pas encore d'objets créés. L'opérateur :: est là pour ces situations.

Exemples avec l'opérateur ::

```
 <?php
class A {
  function example() {
    echo "Je suis la fonction originale A::example().<br>\n";
  }
}
class B extends A {
  function example() {
    echo "Je suis la fonction redéfinie B::example().<br>\n";
    A::example();
  }
}
// Il n'y a pas d'objets de classe A.
// L'affichage est :
//   Je suis la fonction originale A::example().<br>
A::example();
// Création d'un objet de la classe B.
$b = new B;
// L'affichage est :
```

```
//   Je suis la fonction redéfinie B::example().<br>
//   Je suis la fonction originale A::example().<br>
$b->example();
?>
```

Les exemples ci−dessus appellent la fonction example() dans la classe A, mais il n'y a pas encore d'objet de classe A, alors il n'est pas possible d'écrire $a−>example() . A la place, on appelle la fonction example() comme une fonction de classe, c'est−à−dire avec le nom de la classe elle−même, et sans objet.

Il y a des fonctions de classe, mais pas de variables de classe. En fait, il n'y a aucun objet au moment de l'appel de la fonction. Donc, une fonction de classe ne peut accéder à aucune variable (mais elle peut accéder aux variables locales et globales). Il faut proscrire l'utilisation de $this .

Dans l'exemple ci−dessus, la classe B redéfinit la fonction example() . La définition originale dans la classe A est remplacée par celle de B, et n'est plus accessible depuis B, à moins que vous n'appeliez spécifiquement la fonction example() de la classe A avec l'opérateur ::. Ecrivez A::example() pour cela (en fait, il faudrait écrire parent::example() , comme expliqué dans la section suivante).

Dans ce contexte, il y a un objet courant, qui peut avoir d'autres variables objets. De ce fait, lorsqu'il est utilisé depuis une méthode d'un objet, vous pouvez utiliser $this.

## 4.9.5 parent

Il arrive que vous ayez à écrire du code qui faire référence aux variables et fonctions des classes de base. C'est particulièrement vrai si votre classe dérivée est une spécialisation de votre classe de base.

Au lieu d'utiliser le nom littéral de votre classe de base dans votre code, vous pouvez utiliser le mot réservé parent , qui représente votre classe de base (celle indiqué par extends , dans la déclaration de votre classe). En faisant cela, vous éviter d'appeler le nom de votre classe de base directement dans votre code. Si votre héritage change, vous n'aurez plus qu'à modifier le nom de la classe dans la déclaration extends de votre classe.

Exemples avec l'opérateur parent

```
 <?php
class A {
  function example() {
    echo "Je suis A::example() et je fournis une fonctionnalité de base.<br>\n";
  }
}
class B extends A {
  function example() {
    echo "Je suis B::example() et je fournis une fonctionnalité supplémentaire.<br>\n";
    parent::example();
  }
}
$b = new B;
// Cette syntaxe va appeler B::example(), qui, à sont tour, va appeler A::example().
$b->example();
?>
```

## 4.9.6 Sauvegarde d'objets − cas des sessions

Note

En PHP 3, les objets perdent leur association de classe à travers le processus de sauvegarde et relecture. Le type de la variable après relecture est bien objet mais il n'a plus de méthode ou de nom de classe. Cela rend la fonctionnalité plutôt inutile (l'objet est devenu un tableau avec une syntaxe étrange).

| Attention |
| --- |
| La documentation suivante n'est valable que pour PHP 4. |

serialize retourne une chaîne représentant une valeur qui peut être stockée dans les sessions de PHP , ou une base de données. unserialize peut relire cette chaîne pour recréer la valeur originale. serialize va sauver toutes les variables d'un objet. Le nom de la classe sera sauvé mais par les méthodes de cet objet.

Pour permettre à unserialize de lire un objet, la classe de cet objet doit être définie. C'est−à−dire, si vous avez un objet $a de la classe A dans une page php1.php, et que vous le linéarisez avec serialize , vous obtiendrez une chaîne qui fait référence à la classe A, et contient toutes les valeurs de $a. Pour pouvoir le relire avec la fonction unserialize dans une page page2.php, il faut que la définition de la classe A soit présente dans cette deuxième page. Cela peut se faire de manière pratique en sauvant la définition de la classe A dans un fichier séparé, et en l'incluant dans les deux pages page1.php et page2.php.

```
 <?php
classa.inc:
  class A {
    var $one = 1;
    function show_one() {
      echo $this->one;
    }
  }
?>
page1.php:
<?php
  include("classa.inc");
  $a = new A;
  $s = serialize($a);
  // enregistrez $s où la page2.php pourra le trouver.
  $fp = fopen("store", "w");
  fputs($fp, $s);
  fclose($fp);
?>
page2.php:
<?php
  // Ceci est nécessaire pour que unserialize() fonctionne correctement
  include("classa.inc");
  $s = implode("", @file("store"));
  unserialize($s);
  // maintenant, utilisez la méthode show_one de l'objet $a.
  $a->show_one();
?>
```

Si vous utilisez les sessions et la fonction session_register pour sauver des objets, ces objets seront linéarisés automatiquement avec la fonction serialize à la fin de chaque script, et relus avec unserialize au début du prochain script. Cela signifie que ces objets peuvent apparaître dans n'importe quelle page qui utilise vos sessions.

Il est vivement recommandé d'inclure la définition de classe dans toutes vos pages, même si vous n'utilisez pas ces classes dans toutes vos pages. Si vous l'oubliez et qu'un tel objet est présent, il perdra sa classe, et deviendra un objet de classe stdClass sans aucune fonction, et donc, plutôt inutile.

Si, dans l'exemple ci−dessus, $a devient un objet de session avec l'utilisation de session_register("a") , vous devez pensez à inclure le fichier classa.inc dans toutes vos pages, et pas seulement page1.php et page2.php.

## 4.9.7 Les fonctions magiques __sleep et __wakeup

serialize s'assure que votre classe a une méthode avec le nom magique __sleep . Si c'est le cas, cette fonction est appelée avant toute linéarisation. Elle peut alors nettoyer l'objet et on s'attend à ce qu'elle retourne un tableau avec la liste des noms de variables qui doivent être sauvées.

Le but de cette fonction __sleep est de fermer proprement toute connexion à une base de données, de valider les requêtes, de finaliser toutes les actions commencées. Cette fonction est aussi pratique si vous ave de très grands objets qui n'ont pas besoin d'être sauvé entièrement.

A l'inverse, unserialize s'assure de la présence de la fonction magique __wakeup . Si elle existe, cette fonction reconstruit toutes les ressources d'un objet.

Le but de cette fonction __wakeup est de rétablit toutes les connexions aux bases de données, et de recréer les variables qui n'ont pas été sauvées.

## 4.9.8 Références dans un constructeur

Créer des références dans un constructeur peut conduire à des résultats étranges. Ce tutoriel vous guide pour éviter ces problèmes.

```
<?php
class foo {
    function foo($name) {
        // crée une référence dans le tableau global $globalref
        global $globalref;
        $globalref[] = &$this;
        // donne le nom de la variable
        $this->setName($name);
        // et l'affiche
        $this->echoName();
    }
    function echoName() {
        echo "<br>",$this->Name;
    }
    function setName($name)    {
        $this->Name = $name;
    }
}
?>
```

Vérifions maintenant qu'il y a une différence entre $bar1 qui a été créé avec = et $bar2 qui a été créé avec l'opérateur de référence =& :

```
<?php
   $bar1 = new foo('crée dans le constructeur');
   $bar1->echoName();
   $globalref[0]->echoName();
   /* affiche :
   crée dans le constructeur
   crée dans le constructeur
   crée dans le constructeur */
   $bar2 =&new foo('crée dans le constructeur');
   $bar2->echoName();
   $globalref[1]->echoName();
   /* affiche :
   crée dans le constructeur
   crée dans le constructeur
   crée dans le constructeur */
?>
```

Apparemment, il n'y a pas de différence, mais en fait, il y en a une significative : $bar1 et $globalref[0] ne sont pas référencées, ces deux variables sont différentes. Cela est du au fait que l'opérateur "new"ne retourne par de référence, mais retourne une copie.

| Note |
| --- |
| Il n'y a aucune perte de performances (puisque PHP 4 utilise un compteur de références) à retourner des copies au lieu de références. Au contraire, il est souvent mieux de travailler sur les copies plutôt que sur les références, car créer une référence prend un peu plus de temps que de créer une copie qui ne prend virtuellement pas de temps (à moins de créer un tableau géant ou un objet monstrueux, auquel cas il est préférable de passer par des références). |

Pour prouver ceci, regardez le code suivant :

Références et constructeurs

```
<?php
        // maintenant, nous allons changer de nom. Qu'attendez-vous?
        // Vous pouvez vous attendre à ce que les deux variables $bar
        // et  $globalref[0] changent de nom...
        $bar1->setName('modifié');
        // comme prédit, ce n'est pas le cas
        $bar1->echoName();
        $globalref[0]->echoName();
        /* affiche :
        crée dans le constructeur
        modifié */
        // quelle est la différence entre $bar2 et $globalref[1]
        $bar2->setName('modifié');
        // Heureusement, elles sont non seulement égales, mais
        // elles représentent la même variable.
        // donc $bar2->Name et $globalref[1]->Name sont les mêmes
        $bar2->echoName();
        $globalref[1]->echoName();
        /* affiche :
        modifié
        modifié */
?>
```

Un dernier exemple pour bien comprendre.

```
<?php
class a {
    function a($i) {
        $this->value = $i;
        // Essayez de comprendre on n'a pas besoin de
        // référence ici
        $this->b = new b($this);
    }
    function createRef() {
        $this->c = new b($this);
    }
    function echoValue() {
        echo "<br>","class ",get_class($this),': ',$this->value;
    }
}
class b  {
    function b(&$a) {
        $this->a = &$a;
    }
   function echoValue() {
        echo "<br>","class ",get_class($this),': ',$this->a->value;
        }
}
// Essayez de comprendre pourquoi une copie simple va
// conduire à un résultat indésirable à
// la ligne marquée d'une étoile
$a =&new a(10);
$a->createRef();
$a->echoValue();
$a->b->echoValue();
$a->c->echoValue();
$a->value = 11;
$a->echoValue();
$a->b->echoValue(); // *
$a->c->echoValue();
/*
output:
class a: 10
class b: 10
class b: 10
class a: 11
class b: 11
class b: 11
*/
?>
```

## 4.9.9 Comparer des objets en PHP 4

En PHP 4, les objets sont comparés de manière très simple, à savoir : deux instances sont égales si elles ont les mêmes attributs et valeurs, et qu'elles sont de la même classe. Des règles similaires s'appliquent lors de la comparaison avec l'opérateur === .

Si vous exécutons le code suivant :

Exemple de comparaison d'objets en PHP 4

```
 function bool2str($bool) {
    if ($bool === false) {
            return 'FALSE';
    } else {
            return 'TRUE';
    }
}

function compareObjects(&$o1, &$o2) {
    echo 'o1 == o2 : '.bool2str($o1 == $o2)."\n";
    echo 'o1 != o2 : '.bool2str($o1 != $o2)."\n";
    echo 'o1 === o2 : '.bool2str($o1 === $o2)."\n";
    echo 'o1 !== o2 : '.bool2str($o1 !== $o2)."\n";
}

class Flag {
    var $flag;

    function Flag($flag=true) {
            $this->flag = $flag;
    }
}

class SwitchableFlag extends Flag {

    function turnOn() {
        $this->flag = true;
    }

    function turnOff() {
        $this->flag = false;
    }
}

$o = new Flag();
$p = new Flag(false);
$q = new Flag();

$r = new SwitchableFlag();

echo "Compare des instances créées avec les mêmes paramètres\n";
compareObjects($o, $q);

echo "\nCompare des instances créées avec différents paramètres\n";
compareObjects($o, $p);

echo "\nCompare une instance d'un parent avec celle d'une sous-classe\n";
compareObjects($o, $r);
```

Cet exemple va afficher :
Compare des instances créées avec les mêmes paramètres
o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : TRUE
o1 !== o2 : FALSE

Compare des instances créées avec différents paramètres
o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE

Compare une instance d'un parent avec celle d'une sous−classe
o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE
Ce qui est le résultat que nous attendions, au vue des règles édictées. Seules deux instances avec les mêmes valeurs d'attributs, et issues de la même classe sont considérées comme égales.

Même lorsque nous avons une composition d'objet, la même règle de comparaison s'applique. Dans l'exemple ci−dessous, nous allons créer une classer conteneur, qui stocke un tableau associatif Flag .

Comparaison d'objets composés en PHP 4

```
 class FlagSet {
    var $set;

    function FlagSet($flagArr = array()) {
        $this->set = $flagArr;
    }

    function addFlag($name, $flag) {
        $this->set[$name] = $flag;
    }

    function removeFlag($name) {
        if (array_key_exists($name, $this->set)) {
            unset($this->set[$name]);
        }
    }
}


$u = new FlagSet();
$u->addFlag('flag1', $o);
$u->addFlag('flag2', $p);
$v = new FlagSet(array('flag1'=>$q, 'flag2'=>$p));
$w = new FlagSet(array('flag1'=>$q));

echo "\nObjects composés u(o,p) et v(q,p)\n";
compareObjects($u, $v);

echo "\nu(o,p) et w(q)\n";
compareObjects($u, $w);
```

Qui va donner l'affichage suivant :
Objects composés u(o,p) et v(q,p)
o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : TRUE
o1 !== o2 : FALSE

u(o,p) et w(q)
o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE

## 4.9.10 Comparer des objets en PHP 5

Attention

Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez ce module à vos risques et périls.

En PHP 5, la comparaison d'objets et plus compliquée qu'en PHP 4, afin d'être plus proche du comportement des langages orienté objet (bien que PHP n'en soit pas un).

Lors de l'utilisation de l'opérateur de comparaison == , les objets sont comparées de manière simple, à savoir : deux objets sont égaux si ils ont les mêmes attributs et valeurs, et qu'ils sont des instances de la même classe.

D'un autre coté, lors de l'utilisation de l'opérateur d'identité ( === ), les objets sont identiques uniquement si ils font référence au même objet de la même classe.

Un exemple va illustrer ces règles.

Exemple de comparaison d'objets en PHP 5

```
 function bool2str($bool) {
    if ($bool === false) {
```

```
            return 'FALSE';
    } else {
            return 'TRUE';
    }
}

function compareObjects(&$o1, &$o2) {
    echo 'o1 == o2 : '.bool2str($o1 == $o2)."\n";
    echo 'o1 != o2 : '.bool2str($o1 != $o2)."\n";
    echo 'o1 === o2 : '.bool2str($o1 === $o2)."\n";
    echo 'o1 !== o2 : '.bool2str($o1 !== $o2)."\n";
}

class Flag {
    var $flag;

    function Flag($flag=true) {
            $this->flag = $flag;
    }
}


class OtherFlag {
    var $flag;

    function OtherFlag($flag=true) {
            $this->flag = $flag;
    }
}


$o = new Flag();
$p = new Flag();
$q = $o;
$r = new OtherFlag();

echo "Deux instances de la même classe\n";
compareObjects($o, $p);

echo "\nDeux références sur le même objet\n";
compareObjects($o, $q);

echo "\nInstances de classes différentes\n";
compareObjects($o, $r);
```

Cet exemple va afficher :
Deux instances de la même classe
o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : FALSE
o1 !== o2 : TRUE

Deux références sur le même objet
o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : TRUE
o1 !== o2 : FALSE

Instances de classes différentes
o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE

# 4.10 Les références

## 4.10.1 Qu'est ce qu'une référence?

En PHP, les références sont destinées à appeler le contenu d'une variable avec un autre nom. Ce

n'est pas comme en C : ici, les références sont des alias dans la table des symboles. Le nom de la variable et son contenu ont des noms différents, ce qui fait que l'on peut donner plusieurs noms au même contenu. On peut faire l'analogie avec les fichiers sous Unix, et leur nom de fichier : les noms des variables sont les entrées dans un répertoire, tandis que le contenu de la variable est le contenu même du fichier. Faire des références en PHP revient alors à faire des liens sous Unix.

## 4.10.2 Que font les références ?

Les références vous permettent de faire pointer deux variables sur le même contenu. Par exemple, lorsque vous faites :

```
<?php
$a =& $b
?>
```

cela signifie que $a et $b pointent sur la même variable.

| Note |
| --- |
| $a et $b sont complètement égales ici : ce n'est pas $a qui pointe sur $b , ou vice versa. C'est bien $a et $b qui pointent sur le même contenu. |

La même syntaxe peut être utilisée avec les fonctions qui retournent des références, et avec l'opérateur new (PHP 4.0.4 et plus récent):

```
<?php
 $bar =& new fooclass();
 $foo =& find_var ($bar);
?>
```

| Note |
| --- |
| A moins d'utiliser la syntaxe ci−dessus, le résultat de $bar = new fooclass() ne sera pas la même variable que $this dans le constructeur, ce qui signifie que si vous avez utilisé la référence $this dans le constructeur, vous devez assigner la référence, ou bien obtenir deux objets différents. |

Le deuxième intérêt des références est de pouvoir passer des variables par référence. On réalise ceci en faisant pointer des variables locales vers le contenu des variables de fonction. Exemple :

```
<?php
 function foo(&$var) {
   $var++;
 }
 $a=5;
 foo($a);
?>
```

$a vaut 6. Cela provient du fait que dans la fonction foo , la variable $var pointe sur le même contenu que $a . Voir aussi les explications détaillées dans passage par référence .

Le troisième intérêt des références est de retourner des valeurs par référence .

## 4.10.3 Ce que les références ne sont pas

Comme précisé ci−dessus, les références ne sont pas des pointeurs. Cela signifie que le script suivant ne fera pas ce à quoi on peut s'attendre :

```
<?php
 function foo(&$var) {
   $var =& $GLOBALS["baz"];
 }
 foo($bar);
?>
```

Il va se passer que $var dans foo() sera lié à $bar , mais il sera aussi relié à $GLOBALS["baz"] . Il

n'y a pas moyen de lier $bar à quelque chose d'autre en utilisant le mécanisme de référence, car $bar n'est pas accessible dans la fonction foo() (certes, il est représenté par $var et $var possède la même valeur, mais n'est pas relié par la table des symboles).

## 4.10.4 Passage par référence

Vous pouvez passer des variables par référence, de manière à ce que la fonction modifie ses arguments. La syntaxe est la suivante :

```
<?php
 function foo(&$var) {
   $var++;
 }
 $a=5;
 foo ($a);
// $a vaut 6 maintenant
?>
```

Notez qu'il n'y a pas de signe de référence dans l'appel de la fonction, uniquement sur sa définition. La définition de la fonction est suffisante pour passer correctement des arguments par référence.

Les objets suivants peuvent être passés par référence :

- Une variable, i.e. foo($a)
- Un nouvel objet, i.e. foo(new foobar())
- Une référence, retournée par une fonction :

```
<?php
 function &bar() {
  $a = 5;
  return $a;
 }
 foo(bar());
?>
```

Voir aussi des détails dans retourner des références .

Toutes les autres expressions ne doivent pas être passées par référence, car le résultat sera indéfini. Par exemple, les passages par référence suivants sont invalides :

```
<?php
 function bar() // Notez l'absence de &
 {
       $a = 5;
       return $a;
 }
 foo(bar);
 foo($a = 5) // Expression, pas une variable
 foo(5) // Constante, pas une variable
?>
```

Ces fonctionnalités sont valables à partir de PHP 4.0.4.

## 4.10.5 Retourner des références

Retourner des références est toujours utile lorsque vous voulez utiliser une fonction pour savoir à quoi est liée une variable. Lorsque vous retournez une variable par paramètre, utilisez le code suivant

```
<?php
 function &find_var($param) {
 //    ...code...
   return $found_var;
 }
 $foo =& find_var ($bar);
 $foo->x = 2;
?>
```

Dans cet exemple, la propriété de l'objet est retournée dans find_var et lui sera affectée, et non pas à la copie, comme cela sera le cas avec une syntaxe par référence.

| Note |
| --- |
| Contrairement au passage de paramètre, vous devez utiliser & aux deux endroits, à la fois pour indiquer que vous retournez par référence (pas une copie habituelle), et pour indiquer que vous assignez aussi par référence (pas la copie habituelle). |

## 4.10.6 Détruire une référence

Lorsque vous détruisez une référence, vous ne faites que casser le lien entre le nom de la variable et son contenu. Cela ne signifie pas que le contenu est détruit. Par exemple,

```
<?php
$a = 1;
$b =& $a;
unset ($a);
?>
```

Cet exemple ne détruira pas $b , mais juste $a .

Encore une fois, on peut comparer cette action avec la fonction unlink d'Unix.

## 4.10.7 Repérer une référence

De nombreuses syntaxes de PHP sont implémentées via le mécanisme de référence, et tout ce qui a été vu concernant les liaisons entre variables s'applique à ces syntaxes. Par exemple, le passage et le retour d'arguments par référence. Quelques autres exemples de syntaxes :

### 4.10.7.1 Références globales

Lorsque vous déclarez une variable global $var , vous créez en fait une référence sur une variable globale. Ce qui signifie que

```
<?php
$var =& $GLOBALS["var"];
?>
```

Et que, si vous détruisez la variable $var , la variable globale ne sera pas détruite.

### 4.10.7.2 $this

Dans une méthode d'objet $this est toujours une référence sur l'objet courant.

# 5 Caractéristiques

## 5.1 Identification HTTP avec PHP

Les fonctions d'identification HTTP de PHP ne sont disponibles que si PHP est exécuté comme module Apache, et non pas sous la forme d'un CGI. Sous cette forme, il est possible d'utiliser la fonction header pour demander une identification ( "Authentication Required" ) au client, générant ainsi l'apparition d'une fenêtre de demande d'utilisateur et de mot de passe. Une fois que les champs ont été remplis, l'URL sera de nouveau appelée, avec les variables $PHP_AUTH_USER , $PHP_AUTH_PW et $PHP_AUTH_TYPE contenant respectivement le nom d'utilisateur, le mot de passe et le type d'identification. Actuellement, seule l'identification simple ("Basic") est supportée. Reportez−vous à la fonction header pour plus d'informations.

Note

Note sur les versions de PHP

Les auto−globales , telles que $_SERVER , ont été ajoutée en PHP à partir de la version 4.1.0 . $HTTP_SERVER_VARS est disponible depuis PHP 3.

Voici un exemple de script qui force l'identification du client pour accéder à une page :

Exemple d'identification HTTP

```
<?php
 if (!isset($_SERVER['PHP_AUTH_USER'])) {
   header('WWW-Authenticate: Basic realm="My Realm"');
   header('HTTP/1.0 401 Unauthorized');
   echo 'Texte utilisé si le visiteur utilise le bouton d\'annulation';
   exit;
 } else {
   echo "<p>Bonjour, {$_SERVER['PHP_AUTH_USER']}.</p>";
   echo "<p>Votre mot de passe est {$_SERVER['PHP_AUTH_PW']}.</p>";
 }
?>
```

Note

Note de compatibilité

Soyez bien prudent lorsque vous utilisez des en−têtes HTTP avec PHP . Afin de garantir un maximum de compatibilité entre les navigateurs, le mot clé "Basic" doit être écrit avec un B majuscule, et le texte de présentation doit être placé entre guillemets simples, et exactement un espace doit précéder le code **401** dans l'en−tête **HTTP/1.0 401** .

Au lieu d'afficher simplement les variables globales $PHP_AUTH_USER et $PHP_AUTH_PW , vous préférerez sûrement vérifier la validité du nom d'utilisateur et du mot de passe. Par exemple, en envoyant ces informations à une base de données, ou en recherchant dans un fichier dbm.

Méfiez−vous des navigateurs buggés, tels que Internet Explorer. Ils semblent très susceptibles concernant l'ordre des en−têtes. Envoyer l'en−tête d'identification ( WWW−Authenticate ) avant le code de HTTP/1.0 401 semble lui convenir jusqu'à présent.

Pour éviter que quelqu'un écrive un script qui révèle les mots de passe d'une page, à la quelle on a accédé par une identification traditionnelle, les variables globales PHP_AUTH ne seront pas assignées si l'identification externe a été activée pour cette page. Dans ce cas, la variable REMOTE_USER peut être utilisée pour identifier l'utilisateur à l'extérieur. De même que $_SERVER['REMOTE_USER'] .

Note

Note de configuration

PHP utilise la présence de la directive AuthType pour déterminer si une identification externe est activé. évitez donc cette directive de contexte si vous voulez utiliser l'identification de PHP (sinon, les deux identifications se contrediront, et échoueront).

Notez cependant que les manipulations ci−dessus n'empêchent pas quiconque possède une page non authentifiée de voler les mots de passe des pages protégées, sur le même serveur.

Netscape et Internet Explorer effaceront le cache d'identification client s'ils reçoivent une réponse 401. Cela permet de déconnecter un utilisateur, pour le forcer à saisir à nouveau son nom de compte et son mot de passe. Certains programmeurs l'utilisent pour donner un délai d'expiration, ou alors, fournissent un bouton de déconnexion.

Identification HTTP avec nom d'utilisateur/mot de passe forcé

```
<?php
 function authenticate() {
   header('WWW-Authenticate: Basic realm="Test Authentication System"');
   header('HTTP/1.0 401 Unauthorized');
   echo "Vous devez entrer un identifiant et un mot de passe valide pour accéder
   à cette ressource.\n";
   exit;
 }

  if ( !isset($_SERVER['PHP_AUTH_USER']) ||
       ($_POST['SeenBefore'] == 1 && $_POST['OldAuth'] == $_SERVER['PHP_AUTH_USER'])) {
  authenticate();
 }
  else {
  echo "<p>Welcome: {$_SERVER['PHP_AUTH_USER']}<br>";
  echo "Old: {$_REQUEST['OldAuth']}";
  echo "<form action='{$_SERVER['PHP_SELF']}' METHOD='POST'>\n";
  echo "<input type='hidden' name='SeenBefore' value='1'>\n";
  echo "<input type='hidden' name='OldAuth' value='{$_SERVER['PHP_AUTH_USER']}'>\n";
  echo "<input type='submit' value='Re Authenticate'>\n";
  echo "</form></p>\n";
 }
?>
```

Ce comportement n'est pas nécessaire par le standard d'identification HTTP Basic . Les tests avec Lynx ont montré qu'il n'affectait pas les informations de session lors de la réception d'un message de type 401. Ce qui fait que presser la touche "retour" ( back ) à un client Lynx précédemment identifié donnera l'accès direct à la ressource. Cependant, l'utilisateur peut utiliser la touche '_' pour détruire les anciennes identifications.

Notez aussi que tout ceci ne fonctionne pas sous Microsoft IIS et que les limitations de PHP en version CGI sont dues aux limitations de IIS.

Note

Si le safe mode est activé, l'uid du script es ajouté automatiquement dans l'en−tête realm , comme faisant partie de l'en−tête WWW−Authenticate .

## 5.2 Cookies

PHP supporte les cookies de manière transparente. Les cookies sont un mécanisme d'enregistrement d'informations sur le client, et de lecture de ces informations. Ce système permet d'authentifier et de suivre les visiteurs. Vous pouvez envoyer un cookie avec la commande setcookie . Les cookies font partie des en−têtes HTTP , ce qui impose que setcookie soit appelée avant tout affichage de texte. Ce sont les mêmes limitations que pour header .

Tous les cookies qui sont envoyés au client seront automatiquement retournés au script PHP, et transformés en variable, exactement comme pour GET et POST. Si vous souhaitez affecter plusieurs valeurs à un seul cookie, ajoutez **[]** au nom du cookie. Pour plus détails, reportez−vous à la fonction setcookie .

En PHP 4.1.0 et plus récent, le tableau superglobal $_COOKIE sera aussi prédéfini avec les cookies transmis par l'internaute. $HTTP_COOKIE_VARS est aussi prédéfini dans les versions de PHP plus anciennes, lorsque la directive track_vars est activée.

Pour plus de détails, y compris des notes sur les bogues des navigateurs, voyez la fonction setcookie .

# 5.3 Gestion des chargements de fichier

## 5.3.1 Chargements de fichiers par méthode POST

PHP est capable de recevoir des fichiers émis par un navigateur conforme à la norme RFC−1867 (c'est−à−dire Netscape Navigator 3 ou supérieur, Microsoft Internet Explorer 3 avec un patch de Microsoft, ou supérieur sans le patch). Cette fonctionnalité permet de charger des fichiers textes ou binaires. Avec l'authentification et les fonctions de manipulation des fichiers, vous avez un contrôle total sur le chargement et la gestion des fichiers chargés.

| Note |
| --- |
| Notes de configuration<br><br>Voir aussi les directives file_uploads , upload_max_filesize , upload_tmp_dir et post_max_size , dans php.ini |

Notez bien que PHP supporte aussi le chargement par la méthode PUT comme dans le navigateur Netscape Composer et les clients Amaya du W3C. Reportez−vous au chapitre sur le support de la méthode PUT .

Un écran de chargement de fichiers peut être constitué en créant un formulaire de la manière suivante :

Formulaire de chargement de fichier

```
 <FORM ENCTYPE="multipart/form−data" ACTION="_URL_" METHOD="POST">
<INPUT TYPE="hidden" name="MAX_FILE_SIZE" value="1000">
Envoyez ce fichier : <INPUT NAME="userfile" TYPE="file">
<INPUT TYPE="submit" VALUE="Send File">
</FORM>
```

Le paramètre _URL_ doit pointer sur un fichier PHP . L'option MAX_FILE_SIZE cachée doit précéder le nom du fichier à charger, et représente la taille maximale du fichier à charger. La valeur est donnée en octets. Dans ce script, les valeurs suivantes doivent être définies pour assurer un chargement correct.

| Attention |
| --- |
| Le champ MAX_FILE_SIZE est là à titre de conseil au navigateur. Il est très facile de contourner cette restriction. Ne comptez pas sur le respect de cette configuration par le navigateur! La configuration de PHP sur la taille maximale à respecter ( upload_max_filesize ) ne peut être contournée, elle. |

Les variables définies après un téléchargement de fichiers diffèrent beaucoup de version en version. La variable $_FILES existe depuis PHP 4.1.0. Le tableau $HTTP_POST_FILES existe depuis PHP 4.0.0. Ces tableaux contiennent toutes les données sur les fichiers téléchargés. L'utilisation de la variable $_FILES est recommandée. Si la directive PHP register_globals vaut **on** , les variables reliées existeront aussi. register_globals vaut par défaut **off** depuis PHP 4.2.0 .

Le contenu du tableau $_FILES est détaillé dans notre exemple ci−dessous. Notez que l'un

suppose que le nom de la variable du fichier téléchargé est **userfile** , tel que défini dans le formulaire ci−dessus.

*$_FILES['userfile']['name']*
: Le nom original du fichier, tel que sur la machine du client web.

*$_FILES['userfile']['type']*
: Le type MIME du fichier, si le navigateur a fourni cette information. Par exemple, cela pourra être "image/gif" .

*$_FILES['userfile']['size']*
: La taille, en octets, du fichier téléchargé.

*$_FILES['userfile']['tmp_name']*
: Le nom temporaire du fichier qui sera chargé sur la machine serveur.

*$_FILES['userfile']['error']*
: Le code d'erreur error code associé au téléchargement de fichier. error code a été introduit en PHP 4.2.0

Note

Dans les versions de PHP antérieure à la version 4.1.0, la variable $_FILES s'appelait $HTTP_POST_FILES et ce n'était pas une variable auto−globale comme l'est $_FILES . PHP 3 ne supporte pas $HTTP_POST_FILES .

Lorsque register_globals vaut **on** dans le php.ini , des variables supplémentaires sont rendues disponibles. Par exemple $userfile_name vaut $_FILES['userfile']['name'] , $userfile_type vaut $_FILES['userfile']['type'] , etc... Gardez à l'esprit que depuis PHP 4.2.0, register_globals vaut **off** par défaut. Il n'est pas prudent de supposer que cette directive vaut **on** .

Le fichier téléchargé sera stocké temporairement dans le dossier temporaire du système, à moins qu'un autre dossier soit fourni avec la directive upload_tmp_dir du php.ini . Le dossier par défaut du serveur peut être changé dans l'environnement via la variable **TMPDIR** . Modifier la valeur de cette variable avec la fonction putenv dans un script PHP sera sans effet. Cette variable d'environnement peut aussi être utilisée pour s'assurer que d'autres opérations fonctionnent aussi sur les fichiers téléchargés.

Validation de téléchargement de fichiers

Les exemples ci−dessous sont valables avec les versions de PHP plus récentes que la 4.0.2. Voyez les fonctions is_uploaded_file et move_uploaded_file .

```
<?php
 // En PHP < 4.1.0, $HTTP_POST_FILES doit être utilisé
 //    à la place de $_FILES.
// En PHP plus ancien que 4.0.3, utilisez copy() et is_uploaded_file()
 //    à la place de move_uploaded_file

$uploaddir = '/var/www/uploads/';

print "<pre>";
if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploaddir . $_FILES['userfile']['name'])) {
    print "Le fichier est valide, et a été téléchargé
            avec succès. Voici plus d'informations :\n";
    print_r($_FILES);
} else {
    echo "Attaque par upload potentielle. Voici plus d'informations :\n";
    print_r($_FILES);
}
?>
```

Le script PHP qui reçoit le fichier chargé doit pouvoir gérer le fichier de manière appropriée. Vous pouvez utiliser la variable $file_size pour recaler tous les fichiers qui sont trop gros ou trop petits. Vous pouvez utiliser la variable $file_type pour recaler les fichiers qui n'ont pas le bon type. Quelques soient les actions, ce script doit pouvoir supprimer le fichier du dossier temporaire, ou le déplacer ailleurs. Depuis PHP 4.2.0, vous pouvez utiliser l'information dans $_FILES['userfile']['error'] et adapter votre politique en fonction des code d'erreur . Quelque soit votre politique, vous devriez soit effacer le fichier du dossier temporaire, soit le déplacer.

Le fichier sera automatiquement effacé du fichier temporaire à la fin du script, s'il n'a pas été déplacé ou renommé.

## 5.3.2 Explication sur les messages d'erreurs de chargement de fichiers

Depuis PHP 4.2.0, PHP retourne un code d'erreur approprié dans le tableau de fichiers. Ce code d'erreur est accessible à l'index **error** du tableau, qui est créé durant le téléchargement, par PHP . En d'autres termes, le message d'erreur est accessible dans la variable $_FILES['userfile']['error'] .

*UPLOAD_ERR_OK*
: Valeur : 0. Aucune erreur, le téléchargement est correct.

*UPLOAD_ERR_INI_SIZE*
: Valeur : 1. Le fichier téléchargé excède la taille de upload_max_filesize , configuré dans le php.ini .

*UPLOAD_ERR_FORM_SIZE*
: Valeur : 2. Le fichier téléchargé excède la taille de **MAX_FILE_SIZE** , qui a été spécifiée dans le formulaire HTML.

*UPLOAD_ERR_PARTIAL*
: Valeur : 3. Le fichier n'a été que partiellement téléchargé.

*UPLOAD_ERR_NO_FILE*
: Valeur : 4. Aucun fichier n'a été téléchargé.

| Note |
| --- |
| Ces constantes sont apparues en PHP 4.3.0. |

## 5.3.3 Erreurs classiques

La variable MAX_FILE_SIZE ne peut pas spécifier une taille de fichier plus grande que la taille qui a été fixée par upload_max_filesize, dans le fichier php3.ini , ou par php3_upload_max_filesize dans les directives Apache. La valeur par défaut est 2 méga−octets.

Ne pas valider les fichiers que vous manipulez peut donner l'accès aux utilisateurs à des fichiers sensibles dans d'autres dossiers!

Attention : il semble que CERN httpd supprime tout ce qui est après le premier caractère dans l'en−tête MIME. Tant que c'est le cas, CERN httpd ne pourra pas effectuer de chargements de fichiers.

## 5.3.4 Télécharger plusieurs fichiers simultanément

Le téléchargement de plusieurs fichiers est possible en utilisant des noms différents dans l'attribut name de la balise input .

Il est aussi possible de télécharger plusieurs fichiers simultanément et d'obtenir les informations sous forme de tableau. Pour cela, vous devez utiliser la syntaxe de tableau dans les noms de balises HTML, comme vous l'avez fait avec les sélections multiples et les boîtes à cocher.

| Note |
| --- |
| Le support des téléchargement multiples a été ajouté en PHP 3.0.10. |

Télécharger plusieurs fichiers simultanément

```
<FORM ACTION="file−upload.php" METHOD="post" ENCTYPE="multipart/form−data">
 Envoyez plusieurs fichiers : <BR>
```

```
  <INPUT NAME="userfile[]" TYPE="file"><BR>
  <INPUT NAME="userfile[]" TYPE="file"><BR>
  <INPUT TYPE="submit" VALUE="Send files">
</FORM>
```

Lorsque le formulaire ci−dessus a été envoyé, les tableaux $_FILES['userfile'] , $_FILES['userfile']['name'] , et $_FILES['userfile']['size'] seront initialisés (tout comme $HTTP_POST_FILES pour les versions de PHP antérieure à la 4.1.0). Lorsque register_globals est activé, les variables globales concernant les fichiers téléchargés sont aussi initialisées. Chacune d'entre elle contiendra un tableau numériquement indexé, avec les valeurs décrivant les fichiers téléchargés.

Par exemple, supposons que les fichiers /home/test/review.html et /home/test/xwp.out ont été téléchargé. Dans ce cas, $_FILES['userfile']['name'][0] contient review.html et $_FILES['userfile']['name'][1] contient xwp.out . Similairement, $_FILES['userfile']['size'][0] va contenir la taille du fichier review.html , etc...

$_FILES['userfile']['name'][0] , $_FILES['userfile']['tmp_name'][0] , $_FILES['userfile']['size'][0] et $_FILES['userfile']['type'][0] sont aussi créées.

## 5.3.5 Chargement par méthode PUT

Le suppor de la méthode PUT a été modifié entre PHP 3 et PHP 4. En PHP 4, il faut utiliser le flux d'entrée standard pour recevoir le contenu du fichier.

Sauver un fichier par méthode PUT en PHP 4

```
 <?php
/* Les données via la méthode PUT arrivent dans stdin */
$putdata = fopen("php://stdin","r");

/* Ouvre un fichier pour écriture */
$fp = fopen("myputfile.ext","w");

/* Lit les données par bloc de 1ko
    et les écrit dans le fichier */
while ($data = fread($putdata,1024))
  fwrite($fp,$data);

/* Ferme les flux */
fclose($fp);
fclose($putdata);
?>
```

Note

Toute la documentation ci−dessous s'applique à PHP 3 uniquement.

PHP supporte la méthode HTTP PUT utilisée par les navigateurs tels que Netscape Composer et W3C Amaya. Les requêtes de type PUT sont beaucoup plus simples que les chargements de fichiers, et elles ressemblent à :

```
PUT /path/filename.html HTTP/1.1
```

Normalement, cela signifie que le client distant va sauver les données qui suivent dans le fichier: /path/filename.html de votre disque. Ce n'est évidemment pas très sécurisé de laisser Apache ou PHP écraser n'importe quel fichier de l'arborescence. Pour éviter ceci, il faut d'abord dire au serveur que vous voulez qu'un script PHP donné gère la requête. Avec Apache, il y a une directive pour cela : **Script** . Elle peut être placée n'importe où dans le fichier de configuration d'Apache. En général, les webmestres la place dans le bloc <Directory> , ou peut être dans le bloc <Virtualhost> . La ligne suivante fera très bien l'affaire :

```
Script PUT /put.php3
```

Elle indique à Apache qu'il doit envoyer les requêtes de chargement par méthode PUT au script put.php3 . Bien entendu, cela présuppose que vous avez activé PHP pour qu'il prenne en charge les fichiers de type .php3, et que PHP est actif.

Dans le fichier put.php3 , vous pouvez mettre ceci :

```
<?php
copy($PHP_UPLOADED_FILE_NAME,$DOCUMENT_ROOT.$REQUEST_URI);
?>
```

Ce script va copier le fichier chargé par le client distant à l'endroit désiré. Vous aurez probablement à effectuer quelques tests et l'identification de l'utilisateur, avant d'effectuer cette copie. Le seul piège est que lorsque PHP reçoit un chargement par méthode PUT, il va enregistrer le fichier dans le dossier temporaire, tout comme avec la méthode POST . A la fin de la requête, le fichier sera effacé. Ce qui fait que ce script doit placer le fichier chargé quelque part. Le nom du fichier temporaire est placé dans la variable globale $PHP_PUT_FILENAME , et la destination prévue est placée dans $REQUEST_URI (ces noms peuvent changer d'une configuration d'Apache à l'autre). Cette destination est celle qui est demandée par le client, et vous n'avez pas à obéir aveuglément au client. Vous pourriez par exemple, déplacer le fichier dans un dossier de chargement.

## 5.4 Utilisation des fichiers à distance

Aussi longtemps que le support des gestionnaires d'URL ("URL fopen wrapper") sont activés dans le php.ini , vous pouvez utiliser des URLs (HTTP et FTP) avec la majorité des fonctions qui utilisent un nom de fichier comme paramètre. Cela inclut notamment include , include_once , require et require_once . Reportez−vous à Liste des protocoles supportés pour plus d'informations sur les protocoles supportés par PHP .

| Note |
| --- |
| En PHP 4.0.3 et plus récent, pour pouvoir utiliser les gestionnaires d'URL, vous devez configurer PHP avec l'option −−enable−url−fopen−wrapper . |

| Note |
| --- |
| Les versions de PHP pour Windows plus anciennes que 4.3 de supportait pas les gestionnaires d'URL avec les fonctions suivantes : include , include_once , require , require_once , et les fonctions de création d'images de l'extension Images . |

Par exemple, vous pouvez suivre l'exemple suivant pour ouvrir un fichier sur un serveur web distant, analyser les résultats pour extraire les informations dont vous avez besoin, et ensuite l'utiliser dans une requête de base de données, ou simplement éditer les informations dans le style de votre site.

Connaître le titre d'une page distante

```
<?php
$file = fopen ("http://www.example.com/", "r");
if (!$file) {
    echo "<p>Impossible de lire la page.\n";
    exit;
}
while (!feof ($file)) {
    $line = fgets ($file, 1024);
    /* Cela ne fonctionne que si les balises Title sont correctement utilisées */
    if (eregi ("<title>(.*)</title>", $line, $out)) {
        $title = $out[1];
        break;
    }
}
fclose($file);
?>
```

Vous pouvez aussi écrire des fichiers sur un serveur FTP aussi longtemps que vous êtes connecté avec un utilisateur ayant les bons droits d'accès, alors que le fichier n'existait pas encore.

Pour vous connecter avec un utilisateur autre qu'anonyme, vous devez spécifier un nom d'utilisateur (et certainement le mot de passe) dans l'URL, comme par exemple 'ftp://user:password@ftp.example.com/path/to/file'. (Vous pouvez utiliser le même type de syntaxe pour accéder aux fichiers via HTTP lorsqu'ils nécessitent une authentification basique).

Stocker des données sur un serveur distant

```
 <?php
$file = fopen ("ftp://ftp.example.com/incoming/outputfile", "w");
if (!$file) {
    echo "<p>Unable to open remote file for writing.\n";
    exit;
}
  /* Ecriture des données. */
fputs ($file, $_SERVER['HTTP_USER_AGENT'] . "\n");
fclose ($file);
?>
```

Note

Remarque: Vous pouvez avoir l'idée,à partir de l'exemple ci−dessus, d'utiliser la même technique pour écrire sur un log distant, mais comme mentionné ci−dessus vous ne pouvez qu'écrire sur un nouveau fichier en utilisant les fonctions fopen avec une URL. Pour faire des log distribués, nous vous conseillons de regarder la partie syslog .

## 5.5 Gestion des connexions

Note

Les informations suivantes ne sont valables qu'à partir de la version 3.0.7.

Le statut des connexions est conservé en interne par PHP. Il y a trois états possibles :

- 0 − NORMAL (normal)
- 1 − ABORTED (annulé)
- 2 − TIMEOUT (périmé)

Lorsqu'un script PHP est en cours d'exécution, son état est NORMAL. Si le client distant se déconnecte, le statut devient ABORTED. En général, une telle déconnexion provient d'un arrêt temporaire. Si la durée maximale d'exécution de PHP est dépassée, (voir set_time_limit ), le script prend le statut TIMEOUT.

Vous pouvez en outre décider si vous voulez que la déconnexion d'un client provoque l'arrêt de votre script. Il est parfois pratique que vos scripts continuent à s'exécuter jusqu'à la fin, même si le client n'est plus là pour recevoir les informations. Cependant, par défaut, le script s'arrêtera dès que le client se déconnecte. Ce comportement peut être modifié avec la directive ignore_user_abort dans le fichier php.ini ou bien avec la directive Apache ignore_user_abort du fichier Apache httpd.conf ou avec la fonction ignore_user_abort . Si vous ne demandez pas à PHP d'ignorer la déconnexion, et que l'utilisateur se déconnecte, le script sera terminé. La seule exception est si vous avez enregistré une fonction de fermeture, avec register_shutdown_function . Avec une telle fonction, lorsque l'utilisateur interrompt sa requête, à la prochaine exécution du script, PHP va s'apercevoir que le dernier script n'a pas été terminé, et il va déclencher la fonction de fermeture. Cette fonction sera aussi appelée à la fin du script, si celui−ci se termine normalement. Pour pouvoir avoir un comportement différent suivant l'état du script lors de sa finalisation, vous pouvez exécutez des commandes spécifiques à la déconnexion grâce à la commande connection_aborted . Cette fonction retournera TRUE si la connexion a été annulée.

Votre script peut aussi être automatiquement interrompu après une certaine durée de temps. Par défaut, le délai est de 30 secondes. Cette valeur peut être changée en utilisant la directive PHP max_execution_time dans le fichier php.ini ou avec la directive php3_max_execution_time , dans le fichier Apache .conf ou encore avec la fonction set_time_limit . Lorsque le délai expire, le script est terminé, et comme pour la déconnexion du client, une fonction de finalisation sera appelée. Dans cette fonction, vous pouvez savoir si c'est le délai d'expiration qui a causé la fin du script, en appelant la fonction connection_timeout . Cette fonction retournera vrai si le délai d'expiration a été dépassé.

Une chose à noter est que les deux cas ABORTED et TIMEOUT peuvent être appelés en même temps. Ceci est possible si vous demandez à PHP d'ignorer les déconnexions des utilisateurs. PHP va quand même noter le fait que l'utilisateur s'est déconnecté, mais le script va continuer. Puis, lorsqu'il atteint la limite de temps, le script va expirer. A ce moment−là, les deux fonctions connection_timeout et connection_aborted vont retourner TRUE . Vous pouvez aussi vérifier les deux états en un seul appel avec la fonction connection_status . Cette fonction va retourner un champ de bits, avec les états. Si les deux états sont actifs, l'état retourné prendra la valeur 3.

## 5.6 Connexions persistantes aux bases de données

Les connexions persistantes aux bases de données SQL sont des connexions qui ne se referment pas à la fin du script. Lorsqu'une connexion persistante est demandée, PHP s'assure qu'il n'y a pas une autre connexion identique (qui serait ouverte précédemment, avec le même nom d'hôte, d'utilisateur et le même mot de passe), et si une telle connexion existe, elle est utilisée. Sinon, elle est créée. Une connexion identique est une connexion qui a ouvert le même hôte, avec le même nom et le même mot de passe (s'ils sont nécessaires).

Ceux qui ne sont pas rompus aux techniques des serveurs web et leur distribution de la charge de travail, se font parfois une fausse idée de ces connexions persistantes. En particulier, les connexions persistantes ne permettent pas l'ouverture de plusieurs sessions avec le même lien, ne permettent pas la réalisation de transactions efficaces et ne font pas le café. En fait, pour être extrêmement clair sur le sujet, les connexions persistantes ne vous donnent aucune fonctionnalité de plus que les connexions non persistantes.

Alors pourquoi les connexions persistantes?

Cela s'explique par la manière avec laquelle les serveurs web fonctionnent. Il y a trois manières d'utiliser PHP pour générer des pages.

La première est d'utiliser PHP comme un CGI (Common Interface Gateway). Lorsque PHP fonctionne de cette manière, une instance de l'interpréteur PHP est créée puis détruite pour chaque page demandée. Etant donné que cet interpreteur est détruit après chaque requête, toutes les ressources acquises (comme une connexion à une base SQL), sont purement et simplement détruites.

La deuxième méthode, et de loin, la plus prisée, est d'exécuter PHP sous la forme d'un module sur un serveur multi−processus, ce qui revient à dire : Apache. Un tel serveur a typiquement un processus parent qui coordonne un ensemble de processus fils, qui servent les fichiers. Lorsque les requêtes parviennent depuis un client, elles sont transmises à un fils disponible. Cela signifie que si un client fait une deuxième requête, il peut être servi par un processus client différent du premier. Les connexions persistantes vous permettent alors de ne vous connecter à une base SQL que la première fois. Lors des connexions ultérieures, les processus fils pourront réutiliser la connexion ouverte précédemment.

La dernière méthode est d'utiliser PHP sous la forme d'un module de serveur multi−threads. Actuellement, PHP 4 supporte ISAPI, WSAPI, et NSAPI (sous Windows), qui permettent tous d'utiliser PHP comme un module sur un serveur multi−threads tel que Netscape FastTrack (iPlanet), Microsoft's Internet Information Server (IIS), et O'Reilly's WebSite Pro. Le comportement est essentiellement le même que pour les serveurs multi−processus décrits précédemment. Notez que SAPI n'est pas disponible avec PHP 3.

Si les connexions persistantes n'ont aucune fonctionnalité de plus, à quoi servent−elles?

La réponse est extrêmement simple : efficacité. Les connexions persistantes sont un bon moyen d'accélérer les accès à une base SQL si le traitement de connexion à la base est long. Ce temps dépend de nombreux facteurs : le type de base de données, cette base est−elle sur le même serveur ou pas, quelle est la charge du serveur de base de données, etc... Si le temps de connexion est long, les connexions persistantes seront bien utiles, car une fois ouverte par un processus fils, la connexion est réutilisable sans avoir à se reconnecter. Si vous avez 20 processus fils, il suffit d'avoir 20 connexions persistantes ouvertes, une par fils.

Notez que les connexions persistantes ont quelques inconvénients si vous hébergez une base de données, dont le nombre maximal de connexion risque d'être atteint par les connexions persistantes. Si votre base de données accepte jusqu'à 16 connexions simultanées et que, 17 processus essaient de se connecter, le dernier restera sur la touche. S'il y a des erreurs dans les scripts qui ne permettent pas de fermer la connexion (par exemple, une boucle infinie), votre serveur sera rapidement engorgé. Vérifiez la documentation de votre base de données pour savoir comment elle traite les connexions inactives ou abandonnées.

Résumons−nous : les connexions persistantes ont été définies pour avoir les mêmes fonctionnalités que les connexions non persistantes. Les deux types de connexions sont parfaitement interchangeables, et n'affecteront pas le comportement de votre script : uniquement son efficacité.

Voir aussi fbsql_pconnect , ibase_pconnect , ifx_pconnect , imap_popen , ingres_pconnect , msql_pconnect , mssql_pconnect , mysql_pconnect , ociplogon , odbc_pconnect , ora_plogon , pfsockopen , pg_pconnect et sybase_pconnect .

# 5.7 Safe mode

Le "Safe Mode" est le mode de sécurité de PHP : une solution au problème de partage de PHP sur un serveur. Ce système pêche au niveau de l'architecture car il n'est pas correct de tenter de résoudre ce problème au niveau de PHP, mais les solutions alternatives basées sur le serveur web et l'OS ne sont pas réalistes. De nombreux intervenants, notamment les fournisseurs d'hébergement, utilisent le "Safe Mode".

## 5.7.1 Securité et Safe Mode

| Nom | Par défaut | Modifiable |
|---|---|---|
| safe_mode | "0" | PHP_INI_SYSTEM |
| safe_mode_gid | "0" | PHP_INI_SYSTEM |
| safe_mode_include_dir | NULL | PHP_INI_SYSTEM |
| safe_mode_exec_dir | "" | PHP_INI_SYSTEM |
| safe_mode_allowed_env_vars | PHP_ | PHP_INI_SYSTEM |
| safe_mode_protected_env_vars | LD_LIBRARY_PATH | PHP_INI_SYSTEM |
| open_basedir | NULL | PHP_INI_SYSTEM |
| disable_functions | "" | PHP_INI_SYSTEM |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*safe_mode bool*
: Active ou non le mode de sécurité de PHP . Lisez le chapitre sécurité pour plus d'informations.

*safe_mode_gid bool*
: Par défaut, le safe mode fait une comparaison des propriétaires, avant d'ouvrir un fichier. Si vous voulez alléger un peu ce niveau de sécurité, vous pouvez réaliser une comparaison de groupe, et activer cette directive. Si cette directive vaut FALSE (sa valeur par défaut), c'est une comparaison sur les UID , et sinon, sur les GID .

*safe_mode_include_dir string*
: Les vérifications basées sur le UID ou GID sont ignorées lorsque les fichiers sont issus du dossier et de ses sous−dossiers, indiqué dans cette directive. Les dossiers peuvent être aussi dans l' include_path ou bien il faut inclure le chemin complet.

Depuis PHP 4.2.0, cette directive utilise le point virgule de la même façon que le fait include_path , pour permettre de configurer plusieurs dossiers.

La restriction spécifiée est en fait un préfixe, plus qu'un nom de dossier. Cela signifie que "safe_mode_include_dir = /dir/incl" autorise aussi bien "/dir/include" que "/dir/incls", si ils existent. Lorsque vous souhaitez restreindre l'accès à un dossier spécifique, il faut terminer cette directive avec un slash /. Par exemple "safe_mode_include_dir = /dir/incl/"

*safe_mode_exec_dir string*
: Si PHP est utilisé en safe mode, les fonctions comme system et toutes celles qui permettent l'exécution en ligne de commande refuseront d'exécuter des programmes qui ne sont pas dans ce dossier.

*safe_mode_allowed_env_vars string*
: Modifier certaines variables d'environnement est un trou de sécurité potentiel. Cette directive contient une liste de noms variables d'environnement séparées par des virgules, ou de préfixes. En Safe mode, l'utilisateur ne pourra modifier que les variables d'environnement dont le nom commence par l'un des préfixes fourni ici. Par défaut, les utilisateurs ne peuvent modifier que les variables d'environnemt qui commencent par PHP_ (e.g. PHP_FOO=BAR).

| Note |
| --- |
| Si cette directive est vide, PHP autorisera la modification de TOUTES les variables d'environnement. |

*safe_mode_protected_env_vars string*
: Cette directive contient une liste de variables d'environnement que le programmeur ne pourra pas modifier en utilisant la fonction putenv . Ces variables seront protégées, même is la directive safe_mode_allowed_env_vars autorise leur modification.

*open_basedir string*
: Limite les fichiers accessibles par PHP dans l'arborescence. Cette directive **n'est pas** affectée par le safe mode.

Lorsqu'un script tente d'ouvrir un fichier, avec les fonctions fopen ou gzopen , la situation du fichier est vérifiée. Si le fichier se situe hors du dossier spécifié dans cette directive, PHP refusera de l'ouvrir. Les liens symboliques sont résolus, ce qui fait que cette restriction ne peut être contournée par un lien symbolique.

La valeur spéciale . indique que le dossier dans lequel le script est stocké, servira de dossier de base.

Sous Windows, séparez les dossiers par des deux points. Sur les autres systèmes, séparez les dossiers avec un point−virgule. Lorsque PHP est utilisé comme module Apache, les chemins de la directive open_basedir des dossiers parents sont automatiques transmis.

La restriction spécifiée par open_basedir est en fait un préfixe, et non un dossier. Cela signifie que "open_basedir = /dir/incl" donne accès au dossier "/dir/include" et aussi au dossier "/dir/incls" si il existe. Lorsque vous souhaitez restreindre l'accès à un dossier spécifique, ajoutez un slash final. Par exemple "open_basedir = /dir/incl/".

| Note |
| --- |
| Le support des dossiers multiples a été ajouté en PHP 3.0.7. |

La valeur par défaut est de permettre l'ouverture de tous les fichiers.

*disable_functions string*

Cette directive vous permet de désactiver certaines fonctions pour des raisons de sécurité . Elle prend une liste de nom de fonctions, séparés par des virgules. disable_functions n'est pas affectée par le safe mode .

Cette directive doit être configurée dans le fichier php.ini . Par exemple, vous ne pourrez pas la configurer dans le fichier httpd.conf .

*disable_classes string*

Cette directive vous permet de désactiver certaines classes pour des raisons de sécurité . Elle prend une liste de nom de fonctions, séparés par des virgules. disable_functions n'est pas affectée par le safe mode .

Cette directive doit être configurée dans le fichier php.ini . Par exemple, vous ne pourrez pas la configurer dans le fichier httpd.conf .

| Note |
| --- |
| Disponibilité<br><br>Cette directive est disponible depuis PHP 4.3.2 |

Voir aussi register_globals , display_errors et log_errors

Lorsque safe mode est actif, PHP vérifie que le propriétaire du script courant est le même que le propriétaire des fichiers qui seront manipulés par ce script. Par exemple, dans a la situation suivante :

```
−rw−rw−r−−    1 rasmus   rasmus       33 Jul  1 19:20 script.php
−rw−r−−r−−    1 root     root       1116 May 26 18:01 /etc/passwd
```

exécuter le script script.php

```
<?php
readfile('/etc/passwd');
?>
```

générera cette erreur, si le Safe Mode est activé :
Warning: SAFE MODE Restriction in effect. The script whose uid is 500 is not allowed to access /etc/passwd owned by uid 0 in /docroot/script.php on line 2

Cependant, il arrive que la vérification faite avec le nom du propriétaire du fichier soit trop restrictive, et qu'une vérification sur le nom du groupe soit suffisante. C'est une autre fonctionnalité supportée par la directive safe_mode_gid . En lui donnant la valeur de On , les vérifications seront faites sur le GID , alors qu'en lui laissant sa valeur par défaut de Off , les vérifications seront faites

sur le UID .

Si vou utilisez la directive open_basedir au lieu du safe mode , alors les manipulations seront limitées aux fichiers situés dans les dossiers spécifiés. Par exemple :

```
<Directory /docroot>
 php_admin_value open_basedir /docroot
</Directory>
```

Si vous exécutez le script script.php ci−dessus avec la configuration d'open_basedir le résultat sera l'affichage suivant :
Warning: open_basedir restriction in effect. File is in wrong directory in /docroot/script.php on line 2

Vous pouvez aussi désactiver individuellement les fonctions. Par exemple, en ajoutant cette ligne dans le fichier php.ini :

```
disable_functions readfile,system
```

toute utilisation des fonctions readfile et system générera l'affichage suivant :
Warning: readfile() has been disabled for security reasons in /docroot/script.php on line 2

## 5.7.2 Fonctions désactivées par le Safe Mode

Voici une liste non−exhaustive des fonctions désactivées par le Safe Mode .

| Fonction | Limitations |
|---|---|
| dbmopen | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| dbase_open | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| filepro | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| filepro_rowcount | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| filepro_retrieve | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| Fonctions ifx_ | restrictions sql_safe_mode, (!= Safe Mode) |
| Fonctions ingres_ | restrictions sql_safe_mode, (!= Safe Mode) |
| Fonctions mysql_ | restrictions sql_safe_mode, (!= Safe Mode) |
| pg_loimport | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| posix_mkfifo | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| putenv | Obéis aux directives safe_mode_protected_env_vars et safe_mode_allowed_env_vars. Voir aussi la documentation de putenv |

| | |
|---|---|
| move_uploaded_file | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| chdir | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| dl | Cette fonction est désactivée par le safe-mode |
| opérateur guillemets obliques | Cette fonction est désactivée par le safe-mode |
| shell_exec (équivalent fonctionnel des guillemets obliques) | Cette fonction est désactivée par le safe-mode |
| exec | Vous ne pouvez exécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier. |
| system | Vous ne pouvez exécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier. |
| passthru | Vous ne pouvez exécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier. |
| popen | Vous ne pouvez exécuter que les programmes qui sont dans le dossier safe_mode_exec_dir. Pour des raisons pratiques, il n'est pas possible d'utiliser des jokers comme .. dans le chemin de ce dossier. |
| fopen | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| mkdir | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| rmdir | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| rename | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| unlink | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| copy | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>(on source and target ) |
| chgrp | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| chown | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| chmod | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |

| | |
|---|---|
| | De plus, vous ne pouvez pas modifier les SUID, SGID et sticky bit |
| touch | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |
| symlink | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>(note: only the target is checked) |
| link | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>(note: seule le fichier de destination est vérifié. |
| getallheaders | En Safe Mode, les en−têtes commencants par 'authorization' (sensible à la casse) ne seront pas retournés. Attention : cette protection est mise en défaut par la fonction getallheaders sur les serveurs AOL! |
| header | Avec le safe mode, le uid du script est ajouté à la partie realm de l'en−tête WWW−Authenticate si vous utilisez cet en−tête pour l'authentification. |
| variables PHP_AUTH | Avec le safe mode, les variables PHP_AUTH_USER , PHP_AUTH_PW et PHP_AUTH_TYPE ne sont pas disponibles dans la variable $_SERVER . Indépendamment, vous pouvez utiliser la variable REMOTE_USER pour connaître le USER. (note : affectée uniquement depuis PHP 4.3.0) |
| highlight_file , show_source | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>(note : affectée uniquement depuis PHP 4.2.1) |
| parse_ini_file | Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>Note<br>Vérifie que le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté.<br>(note : affectée uniquement depuis PHP 4.2.1) |
| Toutes les fonctions qui utilisent les wrappers de fichiers. ( php4/main/fopen_wrappers.c ) | ?? |

# 5.8 Utiliser PHP en ligne de commande

Depuis la version 4.3, PHP supporte un nouveau type de SAPI (Server Application Programming Interface, c'est à dire Interface de Programmation d'Applications Serveur) appelé CLI , ce qui signifie **Command Line Interface** et se traduit par Interface de Ligne de Commande . Comme son nom l'indique, ce type SAPI cible les applications shell (ou desktop), écrites en PHP. Il y a pas mal de différences enter le type CLI SAPI et les autres SAPI , qui sont expliqués dans ce chapitre.

Le CLI SAPI a été publié pour la première fois avec la version PHP 4.2.0 , mais il est resté expérimental, et doit être explicitement activé avec la commande −−enable−cli , lorsque vous exécutez le script ./configure . Depuis PHP 4.3.0, le CLI SAPI n'est plus expérimental, et est donc

**toujours** compilé et installé comme binaire PHP (appelé aussi php.exe sous Windows).

Les différences les plus notables entre le CLI SAPI et les SAPI sont :

- Contrairement au CGI SAPI , aucun en−tête HTTP n'est écrit dans le résultat.

  Bien que le CGI SAPI fournisse un moyen de supprimer les en−têtes HTTP, il n'y a pas moyen d'activer les en−têtes HTTP dans le CLI SAPI .
- Il y a plusieurs directives du php.ini qui sont ignorées par le CLI SAPI , car elles n'ont pas de sens, en environnement shell :

| Directive | Valeur par défaut pour CLI SAPI | Commentaire |
|---|---|---|
| html_errors | FALSE | Il peut être bien difficile de lire les messages d'erreur sur un terminal lorsqu'ils sont noyés dans des balises HTML sans grand intérêts. Par conséquent, cette directive est forcée à FALSE . |
| implicit_flush | TRUE | Il est souhaitable que tout affichage en provenance de print , echo et consorts soit immédiatement affiché dans le terminal, et non pas placé dans un buffer quelconque. Vous pouvez toujours utiliser la bufferisation de sortie si vous voulez retarder un affichage, ou bien en manipuler ultimement le contenu. |
| max_execution_time | 0 (unlimited) | Etant donné les possibilités infinies de PHP en environnement shell, le temps d'exécution maximal d'un script PHP a été rendu illimité. Alors que les scripts destinés au web doivent s'accomplir en une fraction de seconde, il arrive que les scripts shell requièrent bien plus de temps. |
| register_argc_argv | TRUE | En donnant la valeur de TRUE à cette directive, vous aurez toujours accès à la variable **argc** (représentant le nombre d'arguments passés à l'application) et **argv** (le tableau contenant les arguments passés) dans le CLI SAPI .<br><br>Depuis PHP 4.3.0, les variables $argc et $argv sont définies et remplies avec les valeurs appropriées, en utilisant CLI SAPI . Avant cette version, la création de ces variables était liée au comportement des versions CGI et MODULE , qui requièrent l'activation de la directive PHP register_globals . Indépendamment de la version ou de la valeur de register_globals, vous pouvez toujour accéder à $_SERVER et $HTTP_SERVER_VARS . Par exemple: $_SERVER['argv'] |

| Note |
|---|
| Ces directives ne peuvent pas être initialisées avec d'autres valeurs dans le fichier php.ini ou par une autre méthode. C'est une limitation, car ces valeurs par défaut s'appliquent une fois que tous les autres fichiers de configuration ont été analysés. Cependant, ces valeurs peuvent être modifiées durant l'exécution (ce qui n'est pas logique pour certaines directives, comme par exemple register_argc_argv ). |

- Pour faciliter le travail en environnement shell, les constantes suivantes sont définies :

| Constante | Déscription |
|---|---|
| STDIN | Un descripteur de fichier déjà disponible vers stdin . Cela évite de l'ouvrir avec<br><br>`$stdin = fopen('php://stdin', 'r');` |
| STDOUT | Un descripteur de fichier déjà disponible vers stdout . Cela évite de l'ouvrir avec<br><br>`$stdout = fopen('php://stdout', 'w');` |
| STDERR | Un descripteur de fichier déjà disponible vers stderr . Cela évite de l'ouvrir avec<br><br>`$stderr = fopen('php://stderr', 'w');` |

  Etant donné ce qui précède, vous n'avez pas besoin d'ouvrir un fichier vers stderr par vous−même, mais vous pouvez utiliser cette constante directement, comme un descripteur de fichier :

```
php −r 'fwrite(STDERR, "stderr\n");'
```

  Vous n'avez pas non plus à fermer explicitement ces fichiers, PHP s'en chargera automatiquement.

- CLI SAPI **ne transforme pas** le dossier courant en dossier d'exécution du script!

  Exemple de différence entre CGI SAPI et CLI SAPI :

```
<?php
   /* Un test simple : affiche le dossier d'exécution */
   echo getcwd(), "\n";
?>
```

  Lorsque vous utilisez la version CGI , l'affichage pourra être
  $ pwd
  /tmp

  $ php−cgi −f autre_dossier/test.php
  /tmp/autre_dossier
  Cela monter clairement que PHP modifie le dossier courant, et utilise le dossier du script exécuté.

  En utilisant le CLI SAPI , on obtient :
  $ pwd
  /tmp

  $ php −f autre_dossier/test.php
  /tmp
  Cela donne beaucoup plus de souplesse lorsque vous rédigez des scripts shell avec PHP .

| Note |
| --- |
| CGI SAPI supporte le même comportement de CLI SAPI , en lui passant l'option −C , lorsque vous l'invoquez en ligne de commmande. |

La liste des options de ligne de commande fournies par PHP est disponible en exécutant PHP avec l'option −h :
Usage: php [options] [−f] <file> [args...]
php [options] −r <code> [args...]
php [options] [−− args...]
−s Display colour syntax highlighted source.
−w Display source with stripped comments and whitespace.
−f <file> Parse <file>.
−v Version number
−c <path>|<file> Look for php.ini file in this directory
−a Run interactively
−d foo[=bar] Define INI entry foo with value 'bar'
−e Generate extended information for debugger/profiler
−z <file> Load Zend extension <file>.
−l Syntax check only (lint)
−m Show compiled in modules
−i PHP information
−r <code> Run PHP <code> without using script tags <?..?>
−h This help

args... Arguments passed to script. Use −− args when first argument
starts with − or script is read from stdin

Le CLI SAPI dispose de trois moyens pour lire le code du script PHP que vous voulez exécuter :

1. Indiquer à PHP d'exécuter un fichier :

php mon_script.php

php −f mon_script.php
Les deux méthodes (en utilisant −f ou pas) exécute le script contenu dans le fichier mon_script.php . Vous pouvez choisir n'importe quel fichier, et ces fichiers ne sont pas tenus d'utiliser l'extension .php . N'importe quelle extension peut faire l'affaire.

2. Donner du code PHP à exécuter directement en ligne de commande.

   php −r 'print_r(get_defined_constants());'
   Une attention particulière doit alors être apportée aux variables d'environnement, qui seront remplacées, et aux guillements, qui ont des significations spéciales en ligne de commande.

   | Note |
   | --- |
   | Lisez l'exemple attentivement, il n'y a ni balise d'ouverture, ni balise de fermeture! L'option −r fonctionne sans ces balises, et les ajouter conduira alors à une erreur d'analyse. |

3. Donner du code PHP dans l'entrée standard ( stdin ).

   Cela donne la possibilité de créer dynamiquement du code PHP , puis de le fournir à PHP, et enfin, de le traiter à nouveau en shell. Voici un exemple fictif :
   $ some_application | some_filter | php | sort −u >final_output.txt

Il n'est pas possible de combiner ces trois modes d'exécution.

Comme toute application shell, l'exécutable PHP accepte des arguments, et votre script peut aussi les recevoirs. Le nombre d'arguments n'est pas limité par PHP, mais par le shell, qui a une limite en terme de nombre de caractères qui peuvent être passés. Généralement, vous n'atteidrez pas cette limite, mais cela reste possible. Les arguments passés au script seront transmis via la variable tableau $argv . L,index zéro contiendra toujours le nom du script appelé. Si le code a été fourni en ligne de commande, ou via stdin , ce premier élément vaudra − . L'autre variable globale fournie est $argc qui contient le nombre d'éléments dans le tableau $argv : ce nombre est **différent** du nombre d'arguments passés au script.

Tant que les arguments que vous passez à votre script ne commencent pas par le caractère − , il n'y a rien de spécial à surveiller. Si vous passez des arguments à voter script qui commencent par − , cela posera des problèmes car PHP va penser qu'il doit les interpréter. Pour éviter cela, utilisez le séparateur −− . Après cet argument, tous les arguments suivants seront passés à votre script sans être modifié ou analysé par PHP.

```
# Cela ne va pas exécuter le code, mais afficher l'aide de PHP
$ php −r 'var_dump($argv);' −h
Usage: php [options] [−f] <file> [args...]
[...]

# Cela va passer l'argument '−h' à votre script, et éviter que PHP ne le traite
$ php −r 'var_dump($argv);' −− −h
array(2) {
[0]=>
string(1) "−"
[1]=>
string(2) "−h"
}
```

Cependant, il y a une autre méthode pour utiliser PHP en script shell. Vous pouvez aussi utiliser la ligne #!/usr/bin/php en tout début de votre script, suivi de code PHP compris entre balise ouvrantes/fermantes. Vous pouvez aussi ajouter après #!/usr/bin/php les options nécessaires. Vous pouvez aussi le faire en ligne de commande. De cette manière, le script peut être exécuté comme un script shell habituel :

```
 #!/usr/bin/php
<?php
    var_dump($argv);
?>
```

En supposant que ce fichier s'appelle test , dans le dossier courant, nous pouvons alors faire ceci :
$ chmod 755 test
$ ./test −h −− foo
array(4) {
[0]=>
string(6) "./test"
[1]=>
string(2) "−h"
[2]=>
string(2) "−−"
[3]=>
string(3) "foo"
}
Comme vous le voyez, aucune précaution n'est nécessaire pour passer des paramètres qui commencent par − à votre script.

| Option | Description |
|---|---|
| −s | Affiche le code avec la colorisation syntaxique.<br><br>Cette option utilise le mécanisme interne pour analyser le fichier, et produire une version colorisée du code source, au format HTML. Notez que cette option ne fait que générer un bloc HTML , sans les en−têtes. <code> [...] </code><br><br>Note<br>Cette option ne fonctionne pas avec l'option −r . |
| −w | Affiche la source sans les commentaires et les espaces.<br><br>Note<br>Cette option ne fonctionne pas avec l'option −r . |
| −f | Analyse et exécute le fichier donné après l'option −f . Cette optione est optionnelle, et peut être omise. Le seul nom du fichier est suffisant. |
| −v | Affiche les versions de PHP, PHP SAPI, et Zend sur le terminal. Par exemple :<br>$ php −v<br>PHP 4.3.0−dev (cli), Copyright (c) 1997−2002 The PHP Group<br>Zend Engine v1.2.1, Copyright (c) 1998−2002 Zend Technologies |
| −c | Avec cette option, vous pouvez spécifier le nom du dossier dans lequel PHP va recherche le fichier php.ini , ou encore vous pouvez spécifier un fichier de configuration directement (qui n'a pas obligatoirement à être appelé php.ini ) :<br>$ php −c /custom/directory/ mon_script.php<br><br>$ php −c /custom/directory/custom−file.ini mon_script.php |
| −a | Exécute PHP de manière interactive. |
| −d | Cette option permet de modifier n'importe quelle directive de configuration du fichier php.ini . La syntaxe est : |

| | |
|---|---|
| | −d configuration_directive[=value]<br><br>Exemples:<br># L'omission de la valeur conduit à donner la valeur de "1"<br>$ php −d max_execution_time −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(1) "1"<br><br># Passer une valeur vide conduit à donner la valeur de ""<br>php −d max_execution_time= −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(0) ""<br><br># La directive de configuration sera n'importe quelle valeur passée après le caractère '='<br>$ php −d max_execution_time=20 −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(2) "20"<br>$ php −d max_execution_time=doesntmakesense −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(15) "doesntmakesense" |
| −e | Génére des informations étendues pour le profilage et le débogage. |
| −z | Charge une extension Zend. Si seulement un fichier est fourni, PHP essaie de charger cette extension dans le dossier courant par défaut des librairie sur votre système (généralement spécifié avec /etc/ld.so.conf sous Linux). Passer un nom de fichier avec le chemin complet fera que PHP utilisera ce fichier, sans recherche dans les dossiers classiques. Un chemin de dossier relatif indiquera à PHP qu'il doit chercher uniquement les extensions dans ce dossier. |
| −l | Cette option permet de faire une vérification syntaxique sur le code PHP fourni. En cas de réussite, le message No syntax errors detected in <filename> ( Aucune d'erreur de syntaxe n'ont été détectées dans <nom_de_fichier> est affiché sur la sortie standard, et le script shell retourne 0 . En cas d'erreur, le message Errors parsing <filename> ( Erreurs d'analyse dans le fichier <filename> ) est affiché, en plus des messages d'erreurs détectés par l'analyseur lui même. Le script shell retourne le code 255 .<br><br>Cette option ne détecte pas les erreurs fatales (comme par exemple les fonctions non définies). Utilisez −f si vous voulez tester aussi ces erreurs.<br><br>Note<br>Cette option ne fonctionne pas avec l'option −r . |
| −m | Cette option liste les extensions PHP et Zend compilées et chargées :<br>$ php −m<br>[PHP Modules]<br>xml<br>tokenizer<br>standard<br>session<br>posix<br>pcre<br>overload<br>mysql<br>mbstring<br>ctype<br><br>[Zend Modules] |
| −i | Cette option appelle la fonction phpinfo , et affiche le résultat. Si PHP ne fonctionne pas correctement, il est recommandé d'utiliser la commande php −i et de voir si il n'y a pas d'erreurs affichées avant ou après la table d'information. N'oubliez pas que le résutltat de cette option est au format HTML , et donc grand et illisible directement. |
| −r | Cette option permet l'exécution de PHP directement dans la ligne de commande. Les balises |

| | |
|---|---|
| | de PHP ( <?php et ?> ) **ne sont pas** demandées, et coauseront une erreur d'analyse.<br><br>Note<br>De l'attention doit être portée lors de l'utilisation de cette option de PHP , pour qu'il n'y ait pas de collision avec les substitutions de variables en lignes de commande, réalisées par le shell.<br><br>Exemple conduisant à une erreur d'analyse :<br>$ php −r "$foo = get_defined_constants();"<br>Command line code(1) : Parse error − parse error, unexpected '='<br>Le problème ici est que le shell (sh/bash) effectue une substitution de variables, grâce aux guillemets doubles " . puisque la variable $foo n'est probablement pas définie dans le shell, elle est remplacée par rien, ce qui fait que le code passé à PHP pour l'exécution est :<br>$ php −r " = get_defined_constants();"<br>La solution de ce problème est d'utiliser les guillemets simples ' . Les variables de ces chaînes ne seront pas substituées par leurs valeurs par le shell.<br>$ php −r '$foo = get_defined_constants(); var_dump($foo);'<br>array(370) {<br>["E_ERROR"]=><br>int(1)<br>["E_WARNING"]=><br>int(2)<br>["E_PARSE"]=><br>int(4)<br>["E_NOTICE"]=><br>int(8)<br>["E_CORE_ERROR"]=><br>[...]<br>Si vous utilisez un shell différent de sh/bash, vous pouvez rencontrer d'autres problèmes. N'hésitez pas à ouvrir un rapport de bugs, ou d'envoyer un mail à phpdoc@lists.php.net. Il est toujours très facile d'avoir des problèmes lorsque vous essayez d'inclure des variables shell dans le code, ou d'utiliser les anti−slash pour l'échappement. Vous aurez été prévenu. |
| −h | Avec cette option, vous pouvez obtenir des informations sur la liste des options supporté par votre PHP, avec des explications succintes. |

L'exécutable PHP peut être utilisé pour exécuter des scripts indépendants du serveur web. Si vous êtes sur un système Unix, il est recommandé d'ajouter la ligne de configuration en début de scripts, de le rendre exécutable de manière à ce que le système sache quel programme doit exécuter le script. Sous Windows, vous pouvez associer l'exécutable php.exe avec le double−clic sur les fichiers d'extension .php , ou bien vous pouvez faire un fichier batch pour exécuter le script grâce à PHP. La première ligne utilisée dans le monde Unix ne perturbera pas PHP sous Windows, ce qui rend les scripts facilements portables. Un exemple complet est disponible ci−dessous :

Script prévu pour être exécuté en ligne de commande (script.php)

```
 #!/usr/bin/php
<?php

if ($argc != 2 || in_array($argv[1], array('−−help', '−help', '−h', '−?'))) {
?>

C'est une ligne de commande à une option.

  Utilisation :
  <?php echo $argv[0]; ?> <option>

  <option> peut être un mot que vous souhaitez afficher.
  Avec les options −−help, −help, −h,
  et −?, vous obtiendrez cette aide.

<?php
} else {
    echo $argv[1];
}
?>
```

Dans le script ci−dessus, nous utilisons la première ligne pour indiquer que le fichier doit être exécuté par PHP. Nous travaillons avec une version CLI, donc il n'y a pas d'en−têtes HTTP qui soient affichés. Il y a deux variables que vous pouvez utiliser avec les applications de ligne de commande : $argc et $argv . La première est le nombre d'arguments plus un (le nom du script qui est exécuté). La seconde est un tableau contenant les arguments, commençant avec le nom du script en élément 0 ( $argv[0] ).

Dans notre exemple, nous avons vérifié qu'il y a plus ou moins d'un argument. De plus, si cet argument est −−help , −help , −h ou −? , nous avons affiché un message d'aide, ainsi que le nom du script. Nous nous recevons un autre argument, cet argument est affiché dans le terminal.

Pour exécuter le script ci−dessus sous Unix, vous devez le rendre exécuétable, puis l'appeler avec une commande comme : script.php echothis ou script.php −h . Sous Windows, vous pouvez faire un fichier batch pour cela :

| Fichier batch pour exécuter un script PHP en ligne de commande (script.bat) |
|---|
| @c:\php\php.exe script.php %1 %2 %3 %4 |

Si vous avez nommé le programme ci−dessus script.php , et que vous avez votre exécutable php.exe situé à c:\php\php.exe , ce fichier batch l'exécutera avec les options que vous lui passez : script.bat echothis ou script.bat −h .

Voir aussi l'extension Readline , qui dispose de nombreuses fonctions pour améliorer la convivialité de vos applications en ligne de commande.

# 6 Index des fonctions

## 6.1 Apache

### 6.1.1 Introduction

Ces fonctions sont disponibles lorsque PHP est utilisé comme module Apache.

### 6.1.2 Installation

Pour savoir installer PHP sur Apache, reportez−vous à Apache section , dans le chapitre d'installation.

### 6.1.3 Configuration à l'exécution

Le comportement du module PHP d'Apache est configurable dans le fichier php.ini . Les configurations du php.ini peuvent être remplacées par l'option php_flag dans le fichier de configuration du serveur, ou dans les fichiers locaux .htaccess .

| Désactiver l'analyse des fichiers PHP dans un dossier avec .htaccess |
|---|
| php_flag engine off |

| Nom | Défaut | Modifiable | Fonction |
|---|---|---|---|
| engine | On | PHP_INI_ALL | Active ou désactive l'analyse des fichiers PHP |
| child_terminate | Off | PHP_INI_ALL | Spécifie si les scripts PHP peuvent réclamer la fin des processus fils en fin de requête. Reportez−vous à apache_child_terminate |
| last_modified | Off | PHP_INI_ALL | Envoie la date de modification des scripts PHP dans l'en−tête HTTP Last−Modified: |
| xbit_hack | Off | PHP_INI_ALL | Analyse avec PHP les fichiers exécutables, indépendemment de leur extension. |

Voici un éclaircissement sur l'utilisation des directives de configuration.

*engine booléen*

Cette directive est utile uniquement pour le module Apache. Elle est utilisée par les sites qui souhaitent activer ou désactiver PHP, au cas par cas, par dossier ou par dossier virtuel. En utilisant engine off au bon endroit dans le fichier httpd.conf , PHP peut être activé ou désactivé.

### 6.1.4 Types de ressources

Cette extension ne définit aucune ressource.

### 6.1.5 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Installation
- Configuration à l'exécution

- Types de ressources
- Constantes prédefinies
- apache_child_terminate
- apache_get_version
- apache_lookup_uri
- apache_note
- apache_request_headers
- apache_response_headers
- apache_setenv
- ascii2ebcdic
- ebcdic2ascii
- getallheaders
- virtual

## 6.1.7 apache_get_version() : Retourne la version d'Apache

string **apache_get_version** ( void )

apache_get_version retourne une chaîne de caractères représentant la version Apache, ou FALSE en cas d'échec.

Exemple avec apache_get_version

```
<?php
$version = apache_get_version();
echo "$version\n";
?>
```

cela affichera par exemple :

```
Apache/1.3.29 (Unix) PHP/4.3.4
```

Voir aussi phpinfo .

## 6.1.8 apache_lookup_uri() : Effectue une requête partielle pour l'URI spécifiée et renvoie toutes les informations.

object **apache_lookup_uri** ( string filename )

apache_lookup_uri effectue une requête partielle pour l'URI spécifiée. Cette requête permet de récupérer toutes les informations importantes à propos de la ressource concernée. Les propriétés de la classe renvoyée sont les suivantes :

- status
- the_request
- status_line
- method
- content_type
- handler
- uri
- filename
- path_info
- args
- boundary

- no_cache
- no_local_copy
- allowed
- send_bodyct
- bytes_sent
- byterange
- clength
- unparsed_uri
- mtime
- request_time

| Note |
| --- |
| apache_lookup_uri ne fonctionne que lorsque PHP est installé sous la forme d'un module Apache. |

## 6.1.9 apache_note() : Affiche ou affecte le paramètre "apache request notes"

string **apache_note** ( string note_name , *string note_value* )

apache_note est une fonction spécifique au serveur Apache. Cette fonction affecte ou renvoie la valeur de la variable contenue dans la table notes d'Apache. Si la fonction est appelée avec un argument, elle renvoie la valeur courante de la variable note_name . Si apache_note est appelée avec deux arguments, apache_note affecte à la note note_value la valeur note_name et apache_note retournera la valeur précédente de la variable note_name .

## 6.1.10 apache_request_headers() : Lit tous les en−têtes HTTP de la requête

array **apache_request_headers** ( void )

apache_request_headers retourne un tableau associatif avec tous les en−têtes HTTP de la requête courante. Cela n'est possible que lorsque PHP fonctionne comme module Apache .

| Note |
| --- |
| Avant la version PHP 4.3.0, apache_request_headers était appelée getallheaders . Depuis PHP 4.3.0, getallheaders est un alias de apache_request_headers . |

Exemple avec apache_request_headers

```
 <?php
$headers = apache_request_headers();

foreach ($headers as $header => $value) {
    echo "$header: $value <br />\n";
}
?>
```

| Note |
| --- |
| Vous pouvez aussi obtenir les valeurs des variables CGI communes en les lisant dans l'environnement, ce qui fonctionne, que vous soyez ou non en module Apache . Utilisez la fonction phpinfo pour connaître la liste des variables d'environnement disponibles. |

| Note |
| --- |
| A partir de PHP 4.3.3 vous pouvez utiliser cette fonction avec le module serveur NSAPI et avec les serveurs web Netscape, iPlanet et SunONE. |

Voir aussi apache_response_headers .

## 6.1.11 apache_response_headers() : Lit tous les en−têtes de réponse HTTP

array **apache_response_headers** ( void )

apache_response_headers retourne un tableau avec tous les en−têtes de réponse Apache. Cette fonctionnalité n'est disponible qu'en PHP version 4.3.0 et plus récent.

| Note |
| --- |
| A partir de PHP 4.3.3 vous pouvez utiliser cette fonction avec le module serveur NSAPI et avec les serveurs web Netscape, iPlanet et SunONE. |

Voir aussi apache_request_headers et headers_sent .

## 6.1.12 apache_setenv() : Modifie une variable de subprocess_env Apache

int **apache_setenv** ( string variable , string value , *bool walk_to_top* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.1.13 ascii2ebcdic() : Transforme une chaîne ASCII en EBCDIC

int **ascii2ebcdic** ( string ascii_str )

ascii2ebcdic est une fonction spécifique à Apache, qui n'est disponible que sur les systèmes d'exploitation qui gèrent le format EBCDIC (OS/390, BS2000). Elle traduit la chaîne ASCII ascii_str en son équivalent EBCDIC (avec protection des données binaires) et retourne le résultat.

Voir aussi ebcdic2ascii

## 6.1.14 ebcdic2ascii() : Transforme une chaîne EBCDIC en ASCII

int **ebcdic2ascii** ( string ebcdic_str )

ebcdic2ascii est une fonction spécifique à Apache, qui n'est disponible que sur les OS qui gèrent le format EBCDIC (OS/390, BS2000). Elle traduit la chaîne EBCDIC ebcdic_str en son équivalent ASCII (avec protection des données binaires) et retourne le résultat.

Voir aussi ascii2ebcdic

## 6.1.15 getallheaders() : Lit tous les en−têtes de la requête HTTP

array **getallheaders** ( void )

getallheaders est un alias de apache_request_headers . Elle retourne un tableau associatif avec tous les en−têtes HTTP émis par la requête courante. Lisez le manuel de apache_request_headers pour plus de détails.

| Note |
| --- |
| En PHP 4.3.0, getallheaders est devenu un alias de apache_request_headers . En fait, la fonction a été renommée. Cela est dû au fait qu'elle ne fonctionne que lorsque PHP est compilé comme module Apache . |

| Note |
| --- |
| A partir de PHP 4.3.3 vous pouvez utiliser cette fonction avec le module serveur NSAPI et avec les serveurs web Netscape, iPlanet et SunONE. |

Voir aussi apache_request_headers .

## 6.1.16 virtual() : Effectue une sous−requête Apache

int **virtual** ( string filename )

virtual est une fonction spécifique au serveur Apache. Elle est équivalente à la directive " <!−−#include virtual...−−> " lorsque vous utilisez le module include d'Apache. Cette fonction effectue une sous−requête Apache. C'est très utile lorsque vous utilisez des scripts CGI, des fichiers .shtml ou n'importe quel type de fichier qui doit être analysé par le serveur Apache. Il est à noter que lors de l'utilisation avec des scripts CGI, ces derniers doivent générer un en−tête valide, c'est−à−dire, au minimum un en−tête Content−Type . Pour les fichiers PHP, il est conseillé d'utiliser les fonctions include et require . virtual ne peut pas être utilisé pour inclure un fichier qui est lui−même un fichier PHP.

| Note |
| --- |
| A partir de PHP 4.3.3 vous pouvez utiliser cette fonction avec le module serveur NSAPI et avec les serveurs web Netscape, iPlanet et SunONE. |

# 6.2 Tableaux

## 6.2.1 Introduction

Ces fonctions vous permettent de manipuler et de traiter les tableaux de nombreuses façons. Les tableaux sont très efficaces dès qu'il s'agit de stocker, gérer et traiter des données en groupe.

Les tableaux simples et multi−dimensionnels sont supportés et peuvent être créés par l'utilisateur, ou par une fonction. Il y a des fonctions spécifiques qui remplissent des tableaux à partir de résultats de requêtes, et de nombreuses fonctions retournent un tableau.

Reportez−vous à la section tableaux du manuel pour des explications détaillées sur l'implémentation des tableaux en PHP .

## 6.2.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.2.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.2.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.2.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.2.6 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

*CASE_LOWER (entier)*
: CASE_LOWER est utilisé avec array_change_key_case et sert à convertir tous les index d'un tableau en minuscules. C'est aussi le comportement par défaut de array_change_key_case .

*CASE_UPPER (entier)*
: CASE_UPPER est utilisé avec array_change_key_case et sert à convertir tous les index d'un tableau en majuscule.

Constantes d'ordre de tri

*SORT_ASC (entier)*
: SORT_ASC trie en ordre ascendant

*SORT_DESC (entier)*
SORT_DESC trie en ordre descendant

Autres constantes d'ordre de tri

*SORT_REGULAR (entier)*
SORT_REGULAR compare normalement les valeurs d'un tri
*SORT_NUMERIC (entier)*
SORT_NUMERIC compare numériquement les valeurs d'un tri
*SORT_STRING (entier)*
SORT_STRING compare alphabétiquement les valeurs d'un tri.

*COUNT_NORMAL (entier)*
*COUNT_RECURSIVE (entier)*
*EXTR_OVERWRITE (entier)*
*EXTR_SKIP (entier)*
*EXTR_PREFIX_SAME (entier)*
*EXTR_PREFIX_ALL (entier)*
*EXTR_PREFIX_INVALID (entier)*
*EXTR_PREFIX_IF_EXISTS (entier)*
*EXTR_IF_EXISTS (entier)*
*EXTR_REFS (entier)*

## 6.2.7 Voir aussi

Voir aussi is_array , explode , implode , split , preg_split et unset .


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Voir aussi
- array_change_key_case
- array_chunk
- array_combine
- array_count_values
- array_diff_assoc
- array_diff_uassoc
- array_diff
- array_fill
- array_filter
- array_flip
- array_intersect_assoc
- array_intersect
- array_key_exists
- array_keys
- array_map
- array_merge_recursive
- array_merge
- array_multisort
- array_pad

- array_pop
- array_push
- array_rand
- array_reduce
- array_reverse
- array_search
- array_shift
- array_slice
- array_splice
- array_sum
- array_udiff_assoc
- array_udiff_uassoc
- array_udiff
- array_unique
- array_unshift
- array_values
- array_walk
- array
- arsort
- asort
- compact
- count
- current
- each
- end
- extract
- in_array
- key
- krsort
- ksort
- list
- natcasesort
- natsort
- next
- pos
- prev
- range
- reset
- rsort
- shuffle
- sizeof
- sort
- uasort
- uksort
- usort

### 6.2.9 array_chunk() : Sépare un tableau en tableaux de taille inférieure

array **array_chunk** ( array input , int size , *bool* *preserve_keys* )

array_chunk sépare le tableau input en plusieurs tableaux de taille size . Il est aussi possible que le dernier tableau ait moins de valeurs. Le résultat est un tableau multidimensionnel, indexé numériquement.

En passant la valeur TRUE au paramètre optionnel preserve_keys , vous pouvez forcer PHP à préserver les clés originales du tableau input . Si vous utilisez la valeur par défaut ( FALSE ), de nouveaux index numériques seront fournis, commençant à 0.

Exemple avec array_chunk

```
<?php
 $input_array = array('a', 'b', 'c', 'd', 'e');
 print_r(array_chunk($input_array, 2));
 print_r(array_chunk($input_array, 2, TRUE));
?>
```

L'exemple ci−dessus va afficher :

```
Array
(
[0] => Array
(
[0] => a
[1] => b
)

[1] => Array
(
[0] => c
[1] => d
)

[2] => Array
(
[0] => e
)

)
Array
(
[0] => Array
(
[0] => a
[1] => b
)

[1] => Array
(
[2] => c
[3] => d
)

[2] => Array
(
[4] => e
)

)
```

## 6.2.10 array_combine() : Crée un tableau à partir de deux autres tableaux

array **array_combine** ( array keys , array values )

array_combine retourne un array , dont les clés sont les valeurs de keys , et les valeurs sont les valeurs de values .

array_combine retourne FALSE si le nombre d'éléments de chaque tableau n'est pas égal, ou si les tableaux sont vides.

Exemple avec array_combine

```
<?php
$a = array('vert','rouge','jaune');
$b = array('avocat','pomme','banane');
$c = array_combine($a, $b);

print_r($c);
?>
```

Le script ci−dessus affichera :

```
Array
(
    [vert] => avocat
    [rouge] => pomme
    [jaune] => banane
)
```

Voir aussi array_merge , array_walk et array_values .

## 6.2.11 array_count_values() : Compte le nombre de valeurs dans un tableau

array **array_count_values** ( array input )

array_count_values retourne un tableau contenant les valeurs du tableau input comme clés et leur fréquence comme valeur.

Exemple avec array_count_values

```
<?php
 $array = array(1, "bonjour", 1, "monde", "bonjour");
 print_r(array_count_values($array));
?>
```

Le résultat du programme ci−dessus est :

```
Array
(
[1] => 2
[bonjour] => 2
[monde] => 1
)
```

Note

array_count_values a été ajoutée en PHP 4.0.

Voir aussi count , array_unique , array_values et count_chars .

## 6.2.12 array_diff_assoc() : Calcule la différence de deux tableaux, en prenant en compte les clés

array **array_diff_assoc** ( array array1 , array array2 , *array* ... )

array_diff_assoc retourne un tableau contenant les valeurs du tableau array1 qui ne sont présente dans aucun autre argument. Notez que les clés sont utilisées durant la comparaison, contrairement à array_diff .

Exemple avec array_diff_assoc

```
<?php

$array1 = array ("a" => "vert", "b" => "brun", "c" => "bleu", "rouge");
$array2 = array ("a" => "vert", "jaune", "rouge");
$result = array_diff_assoc ($array1, $array2);

/* Le résultat est :

Array
(
    [b] => brun
    [c] => bleu
    [0] => rouge
)

*/

?>
```

Dans notre exemple ci−dessus, vous pouvez voir que la paire "a" => "vert" est présente dans les deux tableaux, et donc, n'est pas présente dans le résultat de la fonction. Au contraire, la paire 0 => "rouge" est présente dans le résultat, car le second argument "rouge" possède une clé qui est 1 .

Les deux valeurs de la paire clé => valeur sont considérées égales uniquement si (string) $elem1 === (string) $elem2 . En d'autre termes, une comparaison stricte est faite sur les représentation des index, avec le type chaîne.

Note

Notez bien que cette fonction ne travaille que sur une dimension de tableau. Bien sur, vous pouvez utiliser des sous dimensions de tableau comme ceci : array_diff_assoc($array1[0], $array2[0]); .

Voir aussi array_diff , array_intersect et array_intersect_assoc .

## 6.2.13 array_diff_uassoc() : Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction utilisateur

array **array_diff_assoc** ( array array1 , array array2 , *array* ... , callback key_compare_func )

array_diff_uassoc retourne un array contenant toutes les valeurs de array1 qui ne sont pas présentes dans l'un des autres arguments. Notez que les clefs sont utilisées dans les comparaisons contrairement à array_diff . Cette comparaison est effectuée par une fonction callback de l'utilisateur. Cette fonction doit retourner un entier plus petit que, égal à, ou plus grand que zéro si le premier argument est respectivement plus petit que, égal au, ou plus grand que le second. Ce comportement est différent de celui de array_diff_assoc où une fonction interne de comparaison des index est utilisée.

Exemple avec array_diff_uassoc

```
<?php
function key_compare_func($a, $b) {
    if ($a === $b) return 0;
    return ($a > $b)? 1:-1;
}
```

```
$array1 = array("a" => "green", "b" => "brown", "c" => "blue", "red");
$array2 = array("a" => "green", "yellow", "red");
$result = array_diff_uassoc($array1, $array2, "key_compare_func");

?>
```

Le résultat est :

```
Array
(
[b] => brown
[c] => blue
[0] => red
)
```

Dans notre exemple, vous voyez que la paire "a" => "green" est présente dans les deux tableaux et donc ne figure pas dans le résultat de la fonction. Contrairement à cela, la paire 0 => "red" est dans le résultat car dans le second argument, la clef de "red" est 1 .

L'égalité de deux indices est vérifiée par la fonction utilisateur.

| Note |
| --- |
| Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi−dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple, array_diff_uassoc($array1[1], $array2[1], "key_compare_func"); . |

Voir aussi array_diff , array_diff_assoc , array_udiff , array_udiff_assoc , array_udiff_uassoc , array_intersect , array_intersect_assoc , array_uintersect , array_uintersect_assoc et array_uintersect_uassoc .

## 6.2.14 array_diff() : Calcule la différence entre deux tableaux

array **array_diff** ( array array1 , array array2 , *array* ... )

array_diff retourne un tableau qui contient toutes les valeurs du tableau array1 qui sont absentes de tous les autres arguments. Notez que les clés sont préservées.

Exemple avec array_diff

```
<?php
 $array1 = array ("a" => "vert", "rouge", "bleu", "rouge");
 $array2 = array ("b" => "vert", "jaune", "rouge");
 $result = array_diff ($array1, $array2);
 print_r($result);
?>
```

Les valeurs multiples dans $array1 seront toutes traitées de la même façon. L'exemple va afficher :

```
Array
(
[1] => bleu
)
```

| Note |
| --- |
| Deux éléments sont considérés comme égaux si et seulement si (string) $elem1 === (string) $elem2 . En clair : lorsque la représentation en chaîne de caractères est identique. |

| Attention |
| --- |
| Cette fonction était inutilisable en PHP 4.0.4! |

Voir aussi array_diff_assoc , array_intersect et array_intersect_assoc .

## 6.2.15 array_fill() : Remplis un tableau avec une même valeur

array **array_fill** ( int start_index , int num , mixed value )

array_fill crée un tableau avec num entrées de la valeur value . Les index commençant à la valeur start_index .

Exemple avec array_fill

```
<?php
 $a = array_fill(5, 6, 'banane');
 print_r($a);
?>
```

$a vaut :
Array
(
[5] => banane
[6] => banane
[7] => banane
[8] => banane
[9] => banane
[10] => banane
)

Voir aussi str_repeat et range .

## 6.2.16 array_filter() : Filtre les éléments d'un tableau

array **array_filter** ( array input , *callback* *callback* )

array_filter retourne un tableau contenant les éléments du tableau input , filtrés grâce à la fonction callback . Si input est un tableau associatif, les clés sont préservées.

Exemple avec array_filter

```
<?php
function impair($var) {
    return ($var % 2 == 1);
}

function pair($var) {
    return ($var % 2 == 0);
}

$array1 = array ("a"=>1, "b"=>2, "c"=>3, "d"=>4, "e"=>5);
$array2 = array (6, 7, 8, 9, 10, 11, 12);

echo "Impairs :\n";
print_r(array_filter($array1, "impair"));
echo "Pairs :\n";
print_r(array_filter($array2, "pair"));
?>
```

Le programme ci−dessus va afficher :
Impairs :
Array
(
[a] => 1
[c] => 3
[e] => 5
)
Pairs :
Array

```
(
[0] => 6
[2] => 8
[4] => 10
[6] => 12
)
```

La fonction appelée ne doit pas modifier le tableau lui−même, c'est à dire ajouter ou supprimer une valeur. Si des éléments sont modifiés, le comportement de array_filter est indéfini.

Si la fonction callback n'est pas fournie, array_filter va supprimer toutes les entrées de input qui sont égales à FALSE . Voyez conversion en booléen pour plus d'informations.

array_filter sans callback

```
<?php

$entry = array(
             0 => 'foo',
             1 => false,
             2 => −1,
             3 => null,
             4 => ''
          );

print_r(array_filter($entry));
?>
```

Cela va afficher :

```
Array
(
[0] => foo
[2] => −1
)
```

Voir aussi array_map , array_reduce et array_walk .

## 6.2.17 array_flip() : Remplace les clés par les valeurs, et les valeurs par les clés

array **array_flip** ( array trans )

array_flip retourne un tableau dont les clés sont les valeurs du précédent tableau, et les valeurs sont les clés. array_flip ne fonctionne que sur des entiers et des chaînes, et affichera une erreur s'il détecte une clé ou une valeur de type invalide (tableau, objet, booléen, nombre à virgule flottante).

Notez bien que les valeurs de trans doivent être des clés valides, c'est à dire qu'elles doivent être des entiers ( integer ) ou des chaînes de caractères ( string ). Une alerte sera émise si une valeur a un type qui ne convient pas et la paire en question **ne sera pas inversée** .

Si une valeur n'est pas unique, seule la dernière clé sera utilisée comme valeur, et toutes les autres seront perdues.

array_flip retourne FALSE en cas d'échec.

Exemple avec array_flip

```
<?php
 $trans = array_flip ($trans);
 $original = strtr ($str, $trans);
?>
```

Exemple avec array_flip : collision

```
<?php
 $trans = array ("a" => 1, "b" => 1, "c" => 2);
 $trans = array_flip ($trans);
 print_r($trans);
?>
```

maintenant $trans vaut :
Array
(
[1] => b
[2] => c
)

Voir aussi array_values , array_keys et array_reverse .

## 6.2.18 array_intersect_assoc() : Calcule l'intersection de deux tableaux avec des tests sur les index

array **array_intersect_assoc** ( array array1 , array array2 , *array* ... )

array_intersect_assoc retourne un tableau contenant toutes les valeurs de array1 qui sont aussi présentes dans tous les autres arguments. Notez que les clés sont utilisées durant la comparaison, contrairement à array_intersect .

Exemple avec array_intersect_assoc

```
<?php
$array1 = array ("a" => "vert", "b" => "brun", "c" => "bleu", "rouge");
$array2 = array ("a" => "vert", "jaune", "rouge");
$result_array = array_intersect_assoc ($array1, $array2);

/* $result_array ressemble à ceci :

 Array
(
    [a] => vert
)

*/
?>
```

Dans notre exemple, vous pouvez voir que la paire "a" => "vert" est présente dans les deux tableaux, et donc placée dans le dernier tableau. La valeur rouge n'est pas retournée car dans $array1 son index est 2 tandis que dans le tableau $array2 , son index est 1 .

Les deux valeurs de la paire clé => valeur sont considérées égales uniquement si (string) $elem1 === (string) $elem2 . En d'autre termes, une comparaison stricte est faite sur les représentation des index, avec le type chaîne.

Voir aussi array_diff et array_intersect .

## 6.2.19 array_intersect() : Calcule l'intersection de tableaux

array **array_intersect** ( array array1 , array array2 , *array* ... )

array_intersect retourne un tableau contenant toutes les valeurs de array1 qui sont présentes dans tous les autres arguments. Notez que les clés sont préservées.

Exemple avec array_intersect

```
<?php
```

```
  $array1 = array ("a" => "vert", "rouge", "bleu");
  $array2 = array ("b" => "vert", "jaune", "rouge");
  $result = array_intersect ($array1, $array2);
  print_r($result);
?>
```

$result vaut :
Array
(
[a] => vert
[0] => rouge
)

Note

Deux éléments sont considérés comme égaux si et seulement si (string) $elem1 === (string) $elem2 . En clair : lorsque la représentation en chaîne de caractères est identique.

Attention

Cette fonction était boguée en PHP 4.0.4!

Voir aussi array_intersect_assoc , array_diff et array_diff_assoc .

## 6.2.20 array_key_exists() : Vérifie si une clé existe dans un tableau

bool **array_key_exists** ( mixed key , array search )

array_key_exists retourne TRUE s'il existe une clé du nom de key dans le tableau search . key peut être n'importe quelle valeur valide d'index de tableau.

Exemple avec array_key_exists

```
<?php
 $search_array = array("premier" => 1, "second" => 4);
 if (array_key_exists("premier", $search_array)) {
   echo "L'élément 'premier' existe dans le tableau";
 }
?>
```

Note

Cette fonction s'appelait key_exists en PHP version 4.0.6.

Voir aussi isset , array_keys et in_array .

## 6.2.21 array_keys() : Retourne toutes les clés d'un tableau

array **array_keys** ( array input , *mixed search_value* )

array_keys retourne les clés numériques et littérales du tableau input .

Si l'option search_value est spécifiée, seules les clés ayant cette valeur seront retournées. Sinon, toutes les clés de input sont retournées.

Exemple avec array_keys

```
<?php
 $array = array (0 => 100, "couleur" => "rouge");
 print_r(array_keys ($array));

 $array = array ("bleu", "rouge", "vert", "bleu", "bleu");
 print_r(array_keys ($array, "bleu"));

 $array = array ("couleur" => array("bleu", "rouge", "vert"), "taille" => array("petit", "moyen", "grand"));
```

```
  print_r(array_keys ($array));
?>
```

Le résultat du script d'exemple est :
Array
(
[0] => 0
[1] => couleur
)
Array
(
[0] => 0
[1] => 3
[2] => 4
)
Array
(
[0] => couleur
[1] => taille
)

Note

array_keys a été ajoutée en PHP 4. Ci−dessous, voici une implémentation qui fonctionnera sous PHP 3 :

Implémentation de array_keys pour les utilisateurs de PHP 3

```
 <?php
function array_keys ($arr, $term="") {
    $t = array();
    while (list($k,$v) = each($arr)) {
        if ($term && $v != $term) {
            continue;
        }
        $t[] = $k;
    }
    return $t;
}
?>
```

Voir aussi array_values et array_key_exists .

## 6.2.22 array_map() : Applique sur fonction sur des tableaux

array **array_map** ( mixed callback , array arr1 , *array* ... )

array_map retourne un tableau contenant tous les éléments du tableau arr1 , après leur avoir appliqué la fonction callback . Le nombre de paramètres de la fonction callback doit être égal au nombre de tableaux passés dans la fonction array_map .

Exemple avec array_map

```
 <?php
  function cube($n) {
      return $n*$n*$n;
  }
  $a = array(1, 2, 3, 4, 5);
  $b = array_map("cube", $a);
?>
```

Cela conduit à ce que la variable $b contienne :
Array
(
[0] => 1

[1] => 8
[2] => 27
[3] => 64
[4] => 125
)

array_map : utilisation de plusieurs tableaux

```
<?php
 function parle_espagnol($n, $m) {
     return "Le nombre $n se dit $m en espagnol";
 }

 function map_espagnol($n, $m) {
  return array($n => $m);
 }

 $a = array(1, 2, 3, 4, 5);
 $b = array("uno", "dos", "tres", "cuatro", "cinco");
 $c = array_map("parle_espagnol", $a, $b);
 print_r($c);
?>
```

Cela va afficher :
// Contenu de $c
Array
(
[0] => Le nombre 1 se dit uno en espagnol
[1] => Le nombre 2 se dit dos en espagnol
[2] => Le nombre 3 se dit tres en espagnol
[3] => Le nombre 4 se dit cuatro en espagnol
[4] => Le nombre 5 se dit cinco en espagnol
)

// Contenu de $d
Array
(
[0] => Array
(
[1] => uno
)

[1] => Array
(
[2] => dos
)

[2] => Array
(
[3] => tres
)

[3] => Array
(
[4] => cuatro
)

[4] => Array
(
[5] => cinco
)

)

Généralement, lors de l'utilisation de plusieurs tableaux, ils doivent être d'égale longueur, car la fonction de callback est appliquée de manière similaire à tous les tableaux. Si les tableaux sont de tailles inégales, les plus petits seront complétés avec des éléments vides.

Une utilisation intéressante de cette fonction est la construction de tableaux de tableaux, facilement réalisée en passant la valeur NULL comme nom de fonction de callback.

Création d'un tableau de tableaux

```
<?php
$a = array(1, 2, 3, 4, 5);
$b = array("one", "two", "three", "four", "five");
$c = array("uno", "dos", "tres", "cuatro", "cinco");
$d = array("un", "deux","trois","quatre","cinq");

$d = array_map(null, $a, $b, $c, $d);
print_r($d);
?>
```

Le programme ci−dessus va afficher :
Array
(
[0] => Array
(
[0] => 1
[1] => one
[2] => uno
[3] => un
)

[1] => Array
(
[0] => 2
[1] => two
[2] => dos
[3] => deux
)

[2] => Array
(
[0] => 3
[1] => three
[2] => tres
[3] => trois
)

[3] => Array
(
[0] => 4
[1] => four
[2] => cuatro
[3] => quatre
)

[4] => Array
(

[0] => 5
[1] => five
[2] => cinco
[3] => cinq
)

)

Voir aussi array_filter , array_reduce et array_walk .

## 6.2.23 array_merge_recursive() : Combine plusieurs tableaux ensembles, récursivement

array **array_merge_recursive** ( array array1 , array array2 , *array* ... )

array_merge_recursive rassemble tous les éléments de plusieurs tableaux ensembles, en ajoutant les éléments de l'un à la suite des éléments du précédent. array_merge_recursive retourne le tableau résultant.

Si les tableaux passés en arguments ont les mêmes clés (chaînes de caractères), les valeurs sont alors rassemblées dans un tableau, de manière récursive, de façon à ce que, si l'une de ces valeurs est un tableau elle−même, la fonction la rassemblera avec les valeurs de l'entrée courante. Cependant, si deux tableaux ont la même clé numérique, la dernière valeur n'écrasera pas la précédente, mais sera ajoutée à la fin du tableau.

Exemple avec array_merge_recursive

```
<?php
 $ar1 = array("couleur" => array("favorie" => "rouge"), 5);
 $ar2 = array(10, "couleur" => array("favorie" => "vert", "rouge"));
 $result = array_merge_recursive($ar1, $ar2);
 print_r($result);
?>
```

Le résultat sera

```
 Array
(
    [couleur] => Array
        (
            [favorie] => Array
                (
                    [0] => rouge
                    [1] => vert
                )

            [0] => rouge
        )

    [0] => 5
    [1] => 10
)
```

Voir aussi array_merge .

## 6.2.24 array_merge() : Fusionne plusieurs tableaux

array **array_merge** ( array array1 , array array2 , *array* ... )

array_merge rassemble les éléments de plusieurs tableaux ensembles, en ajoutant les valeurs de l'un à la fin de l'autre. Le résultat est un tableau.

Si les tableaux ont des clés en commun, la dernière valeur rencontrée écrasera l'ancienne. Pour les valeurs numériques, cela n'arrive pas, car alors, les valeurs sont ajoutées en fin de tableau.

Exemple avec array_merge

```
<?php
 $array1 = array ("couleur" => "rouge", 2, 4);
 $array2 = array ("a", "b", "couleur" => "vert", "forme" => "trapézoÔde");
 $result = array_merge ($array1, $array2);
 print_r($result);
?>
```

La $result vaut :
Array
(
[couleur] => vert
[0] => 2
[1] => 4
[2] => a
[3] => b
[forme] => trapézoÔde
)

Exemple avec array_merge

```
<?php
 $array1 = array();
 $array2 = array(1 => "data");
 $result = array_merge($array1, $array2);
?>
```

N'oubliez pas que les index numériques seront renumérotés.
Array
(
[0] => data
)

Si vous voulez préserver complètement les tableaux, et que vous souhaitez simplement les ajouter les uns aux autres, utilisez l'opérateur + :

```
<?php
 $array1 = array();
 $array2 = array(1 => "data");
 $result = $array1 + $array2;
?>
```

Les index numériques seront préservés. D'où le résultat :
Array
(
[1] => data
)

Note

Les clés communes seront écrasées successivement par les clés identiques suivantes.

Voir aussi array_merge_recursive et array_combine .

## 6.2.25 array_multisort() : Tri multi−dimensionnel

bool **array_multisort** ( array ar1 , *mixed arg* , *mixed ...* , *array ...* )

array_multisort sert à trier simultanément plusieurs tableaux, ou bien à trier un tableau multi−dimensionnel, suivant l'une ou l'autre de ses dimensions. Les clés sont préservées.

Les tableaux passés en arguments sont traités comme les colonnes d'une table, triées par lignes (un peu comme la clause SQL ORDER BY ). Le premier tableau est la clé primaire de tri. Les valeurs du premier tableau qui sont égales, sont triées grâce au tableau suivant, et ainsi de suite...

La structure des arguments de array_multisort est un peu inhabituelle, mais elle est plus souple. Le premier argument DOIT être un tableau, mais les arguments suivants peuvent être des tableaux ou une ou deux options de tri, prises dans les valeurs suivantes :

Options de tri :

- SORT_ASC : Tri en ordre ascendant
- SORT_DESC : Tri en ordre descendant

Options de type de tri:

- SORT_REGULAR : Comparaison normale des valeurs
- SORT_NUMERIC : Comparaison numérique des valeurs
- SORT_STRING : Comparaison alphabétique des valeurs

Une seule option de tri de chaque type peut être appliquée après un tableau. Une option ne s'applique qu'au tableau précédent. Tous les autres sont mis par défaut à SORT_ASC et SORT_REGULAR .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Trier plusieurs tableaux

```
<?php
 $ar1 = array ("10", 100, 100, "a");
 $ar2 = array (1, 3, "2", 1);
 array_multisort ($ar1, $ar2);
?>
```

Dans cet exemple, après le tri, le premier tableau contient 10, "a", 100, 100; Le deuxième tableau contient 1, 1, "2", 3. Les entrées du second tableau correspondant aux valeurs jumelles du premier tableau (100 et 100), sont aussi triées.

Classer un tableau multi-dimensionnel

```
<?php
 $ar = array (array ("10", 100, 100, "a"), array (1, 3, "2", 1));
 array_multisort($ar[0], SORT_ASC, SORT_STRING,
                 $ar[1], SORT_NUMERIC, SORT_DESC);
?>
```

Dans cet exemple, après le tri, le premier tableau contient 10, 100, 100, "a" (tri alphabétique, ordre croissant); Le deuxième tableau contient 1, 3, "2", 1 (tri numérique, ordre décroissant).

## 6.2.26 array_pad() : Complète un tableau jusqu'à la longueur spécifiée, avec une valeur.

array **array_pad** ( array input , int pad_size , mixed pad_value )

array_pad retourne une copie du tableau input complété jusqu'à la taille de pad_size avec la valeur pad_value . Si pad_size est positif, alors le tableau est complété à droite, s'il est négatif, il est complété à gauche. Si la valeur absolue de pad_size est plus petite que la taille du tableau input , alors le tableau n'est pas complété.

Exemple avec array_pad

```
<?php
  $input = array(12, 10, 9);
  $result = array_pad($input, 5, 0);
// Le résultat est array (12, 10, 9, 0, 0)
  $result = array_pad($input, -7, -1);
// Le résultat est array (-1, -1, -1, -1, 12, 10, 9)
  $result = array_pad($input, 2, "noop");
// pas complété
?>
```

Voir aussi array_fill et range .

## 6.2.27 array_pop() : Dépile un élément de la fin d'un tableau

mixed **array_pop** ( array array )

array_pop dépile et retourne le dernier élément du tableau array , le raccourcissant d'un élément. Si array est vide, ou n'est pas un tableau, array_pop retourne NULL .

| Note |
| --- |
| Cette fonction remet le pointeur au début du tableau (équivalent de reset ). |

Exemple avec array_pop

```
<?php
  $stack = array("orange", "banane", "pomme", "framboise");
  $fruit = array_pop($stack);
  print_r($stack);
  echo $fruit . "\n";
?>
```

Après ceci, $stack n'a plus que 3 éléments: "orange" , "banane" et "pomme" :
Array
(
[0] => orange
[1] => banane
[2] => pomme
)
framboise
tandis que $fruit contient "framboise" .

| Attention |
| --- |
| Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur exacte de cette fonction. |

Voir aussi array_push , array_shift et array_unshift .

## 6.2.28 array_push() : Empile un ou plusieurs éléments à la fin d'un tableau

int **array_push** ( array array , mixed var , *mixed* ... )

array_push considère array comme une pile, et empile les variables passées en paramètres à la fin de array . La longueur du tableau array augmente d'autant. Cela a le même effet que

```
<?php
  $array[] = $var;
?>
```

répété pour chaque var .

array_push retourne le nouveau nombre d'éléments du tableau.

Exemple avec array_push

```
<?php
 $stack = array ("orange", "banane");
 array_push ($stack, "pomme", "bleuet");
 print_r($stack);
?>
```

$stack vaut alors :
Array
(
[0] => orange
[1] => banane
[2] => pomme
[3] => bleuet
)

Voir aussi array_pop , array_shift et array_unshift .

## 6.2.29 array_rand() : Prend une ou plusieurs valeurs, au hasard dans un tableau

mixed **array_rand** ( array input , *int num_req* )

array_rand est pratique lorsque vous voulez sélectionner une ou plusieurs valeurs au hasard dans un tableau. Le paramètre input est un tableau, et num_req spécifie le nombre de valeurs que vous voulez obtenir (par défaut, c'est 1).

Si vous ne demandez qu'une entrée, array_rand retourne l'index de la valeur. Sinon, elle retourne un tableau d'index. Cela vous permet de faire une sélection au hasard de clés, ou bien de valeurs.

Note

Depuis PHP 4.2.0, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec srand ou mt_srand car c'est fait automatiquement.

Exemple avec array_rand

```
<?php
 srand ((double) microtime() * 10000000);
 $input = array ("Neo", "Morpheus", "Trinitée", "Cypher", "Tank");
 $rand_keys = array_rand ($input, 2);
 print $input[$rand_keys[0]]."\n";
 print $input[$rand_keys[1]]."\n";
?>
```

Voir aussi shuffle .

## 6.2.30 array_reduce() : Réduit itérativement un tableau

mixed **array_reduce** ( array input , callback function , *int initial* )

array_reduce applique itérativement la fonction callback aux éléments du tableau input , de manière à réduire le tableau à une valeur simple. Si l'argument optionnel intial est disponible, il sera utilisé pour initialiser le processus, ou bien comme valeur finale si le tableau est vide.

Exemple avec array_reduce

```
<?php
 function rsum($v, $w) {
   $v += $w;
   return $v;
 }
 function rmul($v, $w) {
   $v *= $w;
   return $v;
 }
 $a = array(1, 2, 3, 4, 5);
 $x = array();
 $b = array_reduce($a, "rsum");
 $c = array_reduce($a, "rmul", 10);
 $d = array_reduce($x, "rsum", 1);
?>
```

Dans cet exemple, $b contiendra 15 , $c contiendra 1200 (= 1*2*3*4*5*10), et $d contiendra 1 .

Voir aussi array_filter , array_map , array_unique et array_count_values .

## 6.2.31 array_reverse() : Inverse l'ordre des éléments d'un tableau

array **array_reverse** ( array array , *bool* *preserve_keys* )

array_reverse prend le tableau array et retourne un nouveau tableau qui contient les mêmes éléments mais dans l'ordre inverse, en préservant les clés si le paramètre preserve_keys vaut TRUE .

Exemple avec array_reverse

```
<?php
 $input = array ("php", 4.0, array ("rouge", "vert"));
 $result = array_reverse ($input);
 $result_keyed = array_reverse ($input, TRUE);
 print_r($result);
 print_r($result_keyed);
?>
```

Ce code fait que $result et $result_keyed contiennent les mêmes éléments, mais qu'ils diffèrent au niveau des clés. $result et $result_keyed contiennent :

```
Array
(
[0] => Array
(
[0] => rouge
[1] => vert
)

[1] => 4
[2] => php
)
Array
(
[2] => Array
(
[0] => rouge
[1] => vert
)

[1] => 4
[0] => php
)
```

| Note |
| --- |
| Le second paramètre preserve_keys a été ajouté en PHP 4.0.3. |

Voir aussi array_flip .

## 6.2.32 array_search() : Recherche dans un tableau la clé associée à une valeur

mixed **array_search** ( mixed needle , array haystack , *bool strict* )

array_search recherche needle dans haystack et retourne la clé associée s'il la trouve, ou FALSE sinon.

| Note |
| --- |
| Si needle est une chaîne de caractères, la comparaison est effectuée du manière sensible à la casse. |

| Note |
| --- |
| Avant PHP 4.2.0, array_search retournait NULL en cas d'échec, au lieu de FALSE . |

Si le troisième paramètre strict vaut TRUE , alors array_search s'assurera aussi que le type de needle est le même que celui de la valeur trouvée dans haystack .

Si needle est trouvé plus d'une fois dans haystack , la première clef concordante est retournée. Pour trouver toutes les clefs correspondantes, utilisez plutôt la fonction array_keys avec le paramètre optionnel search_value .

Exemple avec array_search

```
 <?php
$array = array(0 => "blue", 1 => "red", 2 => "green", 3 => "red");

$key = array_search($array, "green"); // $key = 2;
$key = array_search($array, "red");   // $key = 1;
?>
```

| Attention |
| --- |
| Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur exacte de cette fonction. |

Voir aussi array_keys , array_values , array_key_exists et in_array .

## 6.2.33 array_shift() : Dépile un élément au début d'un tableau

mixed **array_shift** ( array array )

array_shift extrait la première valeur d'un tableau et la retourne, en raccourcissant le tableau d'un élément, et en déplaçant tous les éléments vers le bas. Si array est vide, ou n'est pas un tableau, array_shift retourne NULL .

| Note |
| --- |
| Cette fonction remet le pointeur au début du tableau (équivalent de reset ). |

Exemple avec array_shift

```
<?php
 $stack = array ("orange", "banane", "pomme", "bleuet");
 $fruit = array_shift ($stack);
 print_r($stack);
?>
```

$stack contient encore trois éléments :
Array
(
[0] => banane
[1] => pomme
[2] => bleuet
)
et orange a été placé dans $fruit .

Voir aussi array_unshift , array_push et array_pop .

## 6.2.34 array_slice() : Extrait une portion de tableau

array **array_slice** ( array array , int offset , *int length* )

array_slice retourne une série d' élément du tableau array commençant à l'offset offset et représentant length éléments.

Si offset est positif, la série commencera à cet offset dans le tableau array . Si offset est négatif, cette série commencera à l'offset offset mais en commençant à la fin du tableau array .

Si length est fourni et positif, alors la série retournée aura autant d'éléments. Si length est fourni et négatif, alors la série contiendra les éléments depuis l'offset offset jusqu'à length éléments en partant de la fin. Si length est omis, la séquence lira tous les éléments du tableau, depuis l' offset précisé jusqu'à la fin du tableau.

Exemples avec array_slice

```
<?php
$input = array ("a", "b", "c", "d", "e");

$output = array_slice ($input, 2);      // retourne "c", "d" et "e"
$output = array_slice ($input, 2, -1);  // retourne "c", "d"
$output = array_slice ($input, -2, 1);  // retourne "d"
$output = array_slice ($input, 0, 3);   // retourne "a", "b" et "c"
?>
```

Voir aussi array_splice et unset .

## 6.2.35 array_splice() : Efface et remplace une portion de tableau

array **array_splice** ( array input , int offset , *int length* , *array replacement* )

array_splice supprime les éléments désignés par offset et length du tableau input et les remplace par les éléments du tableau replacement , si ce dernier est présent.

Si offset est positif, la série commencera à cet offset dans le tableau input . Si offset est négatif, cette série commencera à l'offset offset mais en commençant à la fin du tableau input .

Si length est donné et positif, alors la série aura autant d'éléments. Si length est donné et négatif, les éléments seront pris dans l'ordre inverse. Si length est omis, la séquence lira tous les éléments du tableau, depuis l'offset offset jusqu'à la fin du tableau. Conseil : pour supprimer tous les

éléments du tableau depuis offset jusqu'à la fin, même si un tableau de remplacement replacement est spécifié, utilisez count($input) à la place de length .

Si replacement est précisé, alors les éléments supprimés sont remplacés par les éléments de ce tableau. Si l' offset et length sont tels que la taille du tableau ne change pas, alors les éléments du tableau de remplacement replacement sont insérés à partir de l'offset offset .

Conseil : si le tableau de remplacement ne contient qu'un seul élément, il n'est pas obligatoire de forcer le type en tableau avec array , à moins que cette variable ne soit elle−même un tableau.

Les codes suivants sont équivalents :

| | |
|---|---|
| array_push($input, $x, $y) | array_splice($input, count($input), 0, array($x, $y)) |
| array_pop($input) | array_splice($input, −1) |
| array_shift($input) | array_splice($input, −1) |
| array_unshift($input, $x, $y) | array_splice($input, 0, 0, array($x, $y)) |
| $a[$x] = $y | array_splice($input, $x, 1, $y) |

array_splice retourne le tableau des éléments supprimés.

Exemples avec array_splice

```
<?php
$input = array ("rouge", "vert", "bleu", "jaune");
array_splice ($input, 2);
// $input vaut array ("rouge", "vert")

$input = array ("rouge", "vert", "bleu", "jaune");
array_splice ($input, 1, −1);
// $input vaut array ("rouge", "jaune")

$input = array ("rouge", "vert", "bleu", "jaune");
array_splice ($input, 1, count($input), "orange");
// $input vaut array ("rouge", "orange")

$input = array ("rouge", "vert", "bleu", "jaune");
array_splice ($input, −1, 1, array("black", "marron"));
// $input vaut array ("rouge", "vert",
//          "bleu", "black", "marron")
?>
```

Voir aussi array_slice , unset et array_merge .

## 6.2.36 array_sum() : Calcule la somme des valeurs du tableau

mixed **array_sum** ( array array )

array_sum retourne la somme des valeurs du tableau, sous forme d'un entier ou d'un nombre à virgule flottante.

Exemple avec array_sum

```
$a = array(2, 4, 6, 8);
echo "sum(a) = ".array_sum($a)."\n";

$b = array("a"=>1.2,"b"=>2.3,"c"=>3.4);
echo "sum(b) = ".array_sum($b)."\n";
```

Le résultat de ce script sera :
sum(a) = 20
sum(b) = 6.9

Note

Dans les versions de PHP antérieure à la version 4.0.6, cette fonction modifiez le tableau passé en argument et convertissez ses arguments en chaînes de caractères (ce qui conduisait généralement à 0).

## 6.2.37 array_udiff_assoc() : Calcule la différence entre des tableaux avec vérification sur les index. Les données sont comparées en utilisant une fonction callback

array **array_udiff_assoc** ( array array1 , array array2 , *array* ... , callback data_compare_func )

array_udiff_assoc retourne un tableau contenant toutes les valeurs de array1 qui ne sont présentes dans aucun autre des arguments. Notez que les clefs sont utilisées dans les comparaisons contrairement à array_diff et array_udiff . La comparaison des données est effectuée en utilisant une fonction callback fournie par l'utilisateur. Ce comportement est différent de celui de array_diff_assoc qui utilise une fonction de comparaison interne.

Exemple avec array_udiff_assoc

```
<?php
class cr {
    private $priv_member;
    function cr($val) {
        $this->priv_member = $val;
    }

    static function comp_func_cr($a, $b) {
        if ($a->priv_member === $b->priv_member) return 0;
        return ($a->priv_member > $b->priv_member)? 1:-1;
    }
}
$a = array("0.1" => new cr(9), "0.5" => new cr(12), 0 => new cr(23), 1=> new cr(4), 2 => new cr(-15),);
$b = array("0.2" => new cr(9), "0.5" => new cr(22), 0 => new cr( 3), 1=> new cr(4), 2 => new cr(-15),);

$result = array_udiff_assoc($a, $b, array("cr", "comp_func_cr"));
print_r($result);
?>
```

Le résultat est :

```
Array
(
[0.1] => cr Object
(
[priv_member:private] => 9
)

[0.5] => cr Object
(
[priv_member:private] => 12
)

[0] => cr Object
(
[priv_member:private] => 23
)
)
```

Dans notre exemple, vous voyez que la paire "1" => new cr(4) est présente dans les deux tableaux et donc absente du tableau résultant.

Pour la comparaison, la fonction callback de l'utilisateur est utilisée. Cette fonction doit retourner un entier inférieur, égal ou supérieur à zéro si le premier argument est respectivement plus petit, égal ou plus grand que le second.

Note

Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi−dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple, array_udiff_assoc($array1[1], $array2[1], "compare_func"); .

Voir aussi array_diff , array_diff_assoc , array_diff_uassoc , array_udiff , array_udiff_uassoc , array_intersect , array_intersect_assoc , array_uintersect , array_uintersect_assoc et array_uintersect_uassoc .

## 6.2.38 array_udiff_uassoc() : Calcule la différence de deux tableaux associatifs, avec une fonction utilisateur

array **array_udiff_uassoc** ( array array1 , array array2 , *array* ... , callback data_compare_func , callback key_compare_func )

array_diff_uassoc retourne un array contenant toutes les valeurs de array1 qui ne sont pas présentes dans l'un des autres arguments. Notez que les clefs sont utilisées dans les comparaisons contrairement à array_diff et array_udiff . Cette comparaison est effectuée par une fonction callback de l'utilisateur : data_compare_func . Ce comportement est différent de celui de array_diff_assoc qui utilise une fonction interne pour les comparaisons. La comparaisons des clefs (indices) est effectuée par la fonction callback key_compare_func . Ce comportement est différent de celui de array_udiff_assoc , vu que cette dernière utilise une fonction interne pour comparer les indices.

Exemple avec array_udiff_uassoc

```
<?php
class cr {
    private $priv_member;
        function cr($val) {
        $this->priv_member = $val;
    }
    static function comp_func_cr($a, $b) {
        if ($a->priv_member === $b->priv_member) return 0;
        return ($a->priv_member > $b->priv_member)? 1:-1;
    }
    static function comp_func_key($a, $b) {
        if ($a === $b) return 0;
        return ($a > $b)? 1:-1;
    }
}
$a = array("0.1" => new cr(9), "0.5" => new cr(12), 0 => new cr(23), 1=> new cr(4), 2 => new cr(-15),);
$b = array("0.2" => new cr(9), "0.5" => new cr(22), 0 => new cr( 3), 1=> new cr(4), 2 => new cr(-15),);

$result = array_udiff_uassoc($a, $b, array("cr", "comp_func_cr"), array("cr","comp_func_key"));
print_r($result);
?>
```

Le résultat est :

```
Array
(
[0.1] => cr Object
(
[priv_member:private] => 9
)

[0.5] => cr Object
(
[priv_member:private] => 12
)

[0] => cr Object
(
[priv_member:private] => 23
)
)
```

Dans notre exemple, vous voyez que la paire "1" => new cr(4) est présente dans les deux tableaux et donc, absente du tableau résultant. Gardez en tête que vous devez fournir deux fonctions callback.

Pour la comparaison, la fonction callback de l'utilisateur est utilisée. Cette fonction doit retourner un entier inférieur, égal ou supérieur à zéro si le premier argument est respectivement plus petit, égal ou plus grand que le second.

Note

Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi-dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple array_udiff_uassoc($array1[0], $array2[0], "data_compare_func", "key_compare_func"); .

Voir aussi array_diff , array_diff_assoc , array_diff_uassoc , array_udiff , array_udiff_assoc , array_intersect , array_intersect_assoc , array_uintersect , array_uintersect_assoc et array_uintersect_uassoc .

## 6.2.39 array_udiff() : Calcule la différence entre deux tableaux en utilisant une fonction callback pour comparer les données

array **array_udiff** ( array array1 , array array2 , *array* ... , callback data_compare_func )

array_udiff retourne un tableau contenant toutes les valeurs de array1 qui ne sont pas présents dans les autres arguments. Notez que les clefs sont préservées. Pour la comparaison des données, la fonction data_compare_function est utilisée. Cette fonction doit retourner un entier plus petit que, égal à, ou plus grand que zéro si le premier argument est respectivement plus petit que, égal au, ou plus grand que le second. Ce comportement est différent de celui de array_diff où une fonction interne de comparaison des données est utilisée.

Exemple avec array_udiff

```
 <?php
class cr {
    private $priv_member;
        function cr($val) {
        $this->priv_member = $val;
    }
    static function comp_func_cr($a, $b) {
        if ($a->priv_member === $b->priv_member) return 0;
        return ($a->priv_member > $b->priv_member)? 1:-1;
    }
}
$a = array("0.1" => new cr(9), "0.5" => new cr(12), 0 => new cr(23), 1=> new cr(4), 2 => new cr(-15),);
$b = array("0.2" => new cr(9), "0.5" => new cr(22), 0 => new cr( 3), 1=> new cr(4), 2 => new cr(-15),);

$result = array_udiff($a, $b, array("cr", "comp_func_cr"));
print_r($result);
?>
```

Le résultat est :

```
Array
(
[0.5] => cr Object
(
[priv_member:private] => 12
)

[0] => cr Object
(
[priv_member:private] => 23
)

)
```

Note

Deux éléments sont considérés égaux si et seulement si (string) $elem1 === (string) $elem2 . En clair : lorsque la représentation en chaîne est la même.

Note

Notez que cette fonction ne vérifie qu'une seule dimension d'un tableau multi-dimensionnel. Vous pouvez, bien sûr, tester une dimension particulière en utilisant par exemple array_udiff($array1[0], $array2[0], "data_compare_func"); .

Voir aussi array_diff , array_diff_assoc , array_diff_uassoc , array_udiff_assoc , array_udiff_uassoc , array_intersect , array_intersect_assoc , array_uintersect , array_uintersect_assoc et array_uintersect_uassoc .

## 6.2.40 array_unique() : Dédoublonne un tableau

array **array_unique** ( array array )

array_unique prend le tableau array et retourne un nouveau tableau, complètement dédoublonné.

| Note |
| --- |
| Deux éléments sont considérés comme égaux si et seulement si (string) $elem1 === (string) $elem2 . En clair : lorsque la représentation en chaîne de caractères est identique. |

| Attention |
| --- |
| Cette fonction était inutilisable en PHP 4.0.4! |

Notez que les clés sont préservées. array_unique conserve la clé de la première valeur rencontrée, et ignore toutes les suivantes.

Exemple avec array_unique

```
<?php
 $input = array ("a" => "vert", "rouge", "b" => "vert", "bleu", "rouge");
 $result = array_unique ($input);
 print_r($result);
?>
```

Cela va afficher :
Array
(
[a] => vert
[0] => rouge
[1] => bleu
)

array_unique et les types de valeurs

```
<?php
 $input = array (4,"4","3",4,3,"3");
 $result = array_unique ($input);
 var_dump($result);
?>
```

Le résultat du programme sera ( PHP 4.0.6) :
array(2) {
[3]=>
int(4)
[4]=>
int(3)
}

## 6.2.41 array_unshift() : Empile un ou plusieurs éléments au début d'un tableau

int **array_unshift** ( array array , mixed var , *mixed* ... )

array_unshift ajoute les éléments passés en argument au début du tableau array . Notez que les éléments sont ajoutés comme un tout, et qu'ils restent dans le même ordre.

array_unshift retourne le nouveau nombre d'éléments du tableau array .

Exemples avec array_unshift

```
<?php
 $queue = array ("orange", "banane");
 array_unshift ($queue, "pomme", "framboise");
 print_r($queue);
?>
```

$queue vaut :
Array
(
[0] => pomme
[1] => framboise
[2] => orange
[3] => banane
)

Voir aussi array_shift , array_push et array_pop .

## 6.2.42 array_values() : Retourne les valeurs d'un tableau

array **array_values** ( array input )

array_values retourne les valeurs du tableau input et indexe numériquement le tableau.

Exemples avec array_values

```
<?php
 $array = array ("taille" => "XL", "couleur" => "vert");
 print_r(array_values ($array));
?>
```

Cet exemple affiche :
Array
(
[0] => XL
[1] => vert
)

Note

array_values a été ajoutée en PHP 4. Ci−dessous, voici une implémentation pour ceux qui utilisent toujours PHP 3.

array_values pour les utilisateurs PHP 3

```
<?php
function array_values ($arr) {
    $t = array();
    while (list($k, $v) = each ($arr)) {
        $t[] = $v;
    }
    return $t;
}
?>
```

Voir aussi array_keys .

## 6.2.43 array_walk() : Exécute une fonction sur chacun des membres d'un tableau.

bool **array_walk** ( array array , callback function , *mixed* *userdata* )

array_walk exécute la fonction func avec chaque élément du tableau arr . Les éléments sont passés en tant que premier argument de la fonction func . func doit être une fonction définie par l'utilisateur, et non pas une fonction native PHP . Vous ne pouvez pas utiliser array_walk directement avec str2lower , il faut absolument passer par une fonction utilisateur. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Si func a besoin de plus d'un argument, une alerte sera générée pour chaque appel de func . Ces alertes sont supprimées en ajoutant le suffixe '@' avant l'appel de array_walk ou simplement en utilisant error_reporting .

Note

Si func doit travailler avec les véritables valeurs du tableau, spécifiez que le premier paramètre de func doit être passé par référence. Alors, les éléments seront directement modifiés dans le tableau.

Note

Passer les clés et userdata à func a été ajouté en PHP 4.0.

En PHP 4, reset doit être appelé si nécessaire, car array_walk ne reinitialise pas automatiquement le tableau.

Exemple avec array_walk

```
 <?
<font face="courrier">PHP</font>

  $fruits = array ("d"=>"citron", "a"=>"orange", "b"=>"banane", "c"=>"pomme");

  function test_alter (&$item1, $key, $prefix) {
    $item1 = "$prefix: $item1";
  }

  function test_print ($item2, $key) {
    echo "$key. $item2<br>\n";
  }

  array_walk ($fruits, 'test_print');
  reset ($fruits);

  array_walk ($fruits, 'test_alter', 'fruit');
  reset ($fruits);

  array_walk ($fruits, 'test_print');
?>
```

L'affichage de ce script sera :

```
d. citron<br>
a. orange<br>
b. banane<br>
c. pomme<br>
d. fruit: citron<br>
a. fruit: orange<br>
b. fruit: banane<br>
c. fruit: pomme<br>
```

Voir aussi each et list .

## 6.2.44 array() : Crée un tableau

array **array** ( *mixed* ... )

array retourne un tableau créé avec les paramètres passés. On peut attribuer un index particulier à une valeur avec l'opérateur =?> .

Note

array est un élément de langage utilisé pour représenter des tableaux littéraux, et non pas une fonction au sens strict du terme.

La syntaxe "index => valeur", séparée par des virgules, définit les index et leur valeur. Un index peut être une chaîne ou un nombre. Si l'index est omis, un index numérique sera automatiquement généré (commençant à 0). Si l'index est un entier, le prochain index généré prendra la valeur d'index la plus grande + 1. Notez que si deux index identiques sont définis, le dernier remplacera le premier.

L'exemple suivant montre comment créer un tableau à deux dimensions, comment spécifier les index d'un tableau associatif, et comment générer automatiquement des index numériques.

Exemple avec array

```
<?php
$fruits = array (
    "fruits"  => array ("a"=>"orange", "b"=>"banane", "c"=>"pomme"),
    "nombres" => array (1, 2, 3, 4, 5, 6),
    "trous"   => array ("premier", 5 => "second", "troisième")
);
```

Index automatique d'un tableau avec array

```
<?php
 $array = array( 1, 1, 1, 1,  1, 8=>1,  4=>1, 19, 3=>13);
 print_r($array);
?>
```

qui affichera :

```
Array
(
    [0] => 1
    [1] => 1
    [2] => 1
    [3] => 13
    [4] => 1
    [8] => 1
    [9] => 19
)
```

Notez bien que l'index '3' est défini deux fois, et conserve finalement sa dernière valeur de 13. L'index '4' est défini après l'index '8', et l'index généré suivant (valeur 19) est 9, puisque le plus grand index est alors 8.

Cet exemple crée un tableau dont les index commencent à 1.

Tableau d'index commençant à 1

```
<?php
 $firstquarter  = array(1 => 'Janvier', 'Février', 'Mars');
 print_r($firstquarter);
?>
```

qui affichera :

```
Array
(
[1] => Janvier
[2] => Février
[3] => Mars
)
```

Voir aussi array_pad , list et range .

## 6.2.45 arsort() : Trie un tableau en ordre inverse

bool **arsort** ( array array , *int* *sort_flags* )

arsort trie un tableau de telle manière que la corrélation entre les index et les valeurs soit conservée. L'usage principal est lors de tri de tableaux associatifs où l'ordre des éléments est important.

Exemple avec arsort

```
<?php
 $fruits = array("d"=>"papaye","a"=>"orange","b"=>"banane","c"=>"ananas");
 arsort ($fruits);
 for (reset ($fruits); $key = key($fruits); next($fruits)) {
   echo "fruits[$key] = ".$fruits[$key]."\n";
 }
?>
```

Cet exemple va afficher:
fruits[d] = papaye
fruits[a] = orange
fruits[b] = banane
fruits[c] = ananas
Les fruits ont été triés en ordre alphabétique inverse, et leurs index respectifs ont été conservés.

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel sort_flags . Pour plus de détails, voyez le manuel pour la fonction sort .

Voir aussi asort , rsort , ksort et sort .

## 6.2.46 asort() : Trie un tableau en ordre

bool **asort** ( array array , *int* *sort_flags* )

asort trie un tableau de telle manière que la corrélation entre les index et les valeurs soit conservée. L'usage principal est lors de tri de tableaux associatifs où l'ordre des éléments est important.

Exemple avec asort

```
<?php
 $fruits = array(  "d"=>"papaye",
                   "a"=>"orange",
                   "b"=>"banane",
                   "c"=>"ananas");
 asort($fruits);
 for(reset($fruits); $key = key($fruits); next($fruits)) {
   echo "fruits[$key] = ".$fruits[$key]."\n";
 }
?>
```

Cet exemple va afficher:
fruits[c] = ananas
fruits[b] = banane
fruits[a] = orange
fruits[d] = papaye
Les fruits ont été triés par ordre alphabétique, et leurs index respectifs ont été conservés.

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel sort_flags . Pour plus de détails, voyez le manuel pour la fonction sort .

Voir aussi arsort , rsort , ksort et sort .

## 6.2.47 compact() : Crée un tableau contenant les variables et leur valeur

array **compact** ( mixed varname , *mixed* ... )

compact accepte différents paramètres. Les paramètres peuvent être des variables contenant des chaînes, ou un tableau de chaînes, qui peut contenir d'autres tableaux de noms, que compact traitera récursivement.

Pour chacun des arguments, compact recherche une variable avec une variable de même nom dans la table courante des symboles, et l'ajoute dans le tableau, de manière à avoir la relation nom => 'valeur de variable'. En bref, c'est le contraire de la fonction extract . compact retourne le tableau ainsi créé.

Exemple avec compact

```
<?php
  $ville = "Montréal";
  $province = "Québec";
  $evenement = "Conférence PHP Québec";
  $location_vars = array("ville", "province");
  $result = compact("evenement", $location_vars);
?>
```

Après ce programme, $result contiendra :
Array
(
[evenement] => Conférence PHP Québec
[ville] => Montréal
[province] => Québec
)

Voir aussi extract .

## 6.2.48 count() : Compte le nombre d'éléments d'une variable

int **count** ( mixed var , *int* *mode* )

count retourne le nombre d'éléments dans var , qui est généralement un tableau (et tout le reste n'aura qu'un élément).

Si var n'est pas un tableau, 1 sera retourné (exception : count( NULL ) égale 0 ).

Note

Le paramètre optionnel mode est disponible depuis PHP 4.2.0.

Si le paramètre optionnel mode vaut COUNT_RECURSIVE (ou 1), count va compter récursivement les tableaux. C'est particulièrement pratique pour compter le nombre d'éléments d'un tableau. La valeur par défaut pour mode est 0 .

Attention

count peut retourner 0 pour une variable qui n'a pas été affectée, ou pour un tableau vide. Utilisez plutôt isset pour tester si la variable existe.

Reportez vous à la section sur les Tableaux du manuel, pour plus de détails sur le fonctionnement des tableaux en PHP .

Exemple avec count

```
<?php
 $a[0] = 1;
 $a[1] = 3;
 $a[2] = 5;
 $result = count ($a);
// $result == 3

 $b[0] = 7;
 $b[5] = 9;
 $b[10] = 11;
 $result = count ($b);
// $result == 3;
?>
```

Exemple récursif avec count (PHP >= 4.2.0)

```
<?php
$food = array( 'fruits'  => array('orange', 'banane', 'pommes'),
               'legumes'  => array('carotte', 'chou','pois'));

// count récursif
echo count($food, COUNT_RECURSIVE); // affiche 8

// count normal
echo count($food);                  // affiche 2

?>
```

Note

sizeof est un alias de count .

Voir aussi is_array , isset et strlen .

## 6.2.49 current() : Transforme une variable en tableau

mixed **current** ( array array )

Chaque tableau entretient un pointeur interne, qui est initialisé lorsque le premier élément est inséré dans le tableau.

current ne fait que retourner l'élément courant pointé par le pointeur interne du tableau array . current ne déplace pas le pointeur. Si le pointeur est au−delà du dernier élément de la liste, current retourne FALSE .

Attention

Si le tableau des éléments vides ou des zéros (0 ou "", la chaîne vide) alors current retournera FALSE pour ces éléments. Il est donc impossible de déterminer si vous êtes réellement à la fin de la liste en utilisant la fonction current . Pour passer en revue proprement un tableau qui peut contenir des éléments vides ou des zéros, utilisez la fonction each .

Exemple avec current

```
<?php
$transport = array('pied', 'velo', 'voiture', 'avion');
$mode = current($transport); // $mode = 'pied';
$mode = next($transport);    // $mode = 'velo';
$mode = current($transport); // $mode = 'velo';
$mode = prev($transport);    // $mode = 'pied';
$mode = end($transport);     // $mode = 'avion';
$mode = current($transport); // $mode = 'avion';
?>
```

Voir aussi end , key , next , prev et reset .

## 6.2.50 each() : Retourne chaque paire clé/valeur d'un tableau

array **each** ( array array )

each retourne la paire clé/valeur courante du tableau array et avance le pointeur de tableau. Cette paire est retournée dans un tableau de 4 éléments, avec les clés 0 , 1 , key , et value . Les éléments 0 et key contiennent le nom de la clé et, et 1 et value contiennent la valeur.

Si le pointeur interne de tableau est au−delà de la fin du tableau, each retourne FALSE .

Exemples avec each

```
<?php
 $foo = array("ghislain", "damien", "didou", "guillaume", "jim", "christophe");
 $bar = each($foo);
 print_r($bar);
?>
```

$bar contient maintenant les paires suivantes :
Array
(
[1] => ghislain
[value] => ghislain
[0] => 0
[key] => 0
)

```
<?php
 $foo = array( "Mehdi" => "Didou","Damien" => "Dams");
 $bar = each($foo);
 print_r($bar);
?>
```

$bar contient maintenant les paires suivantes :
Array
(
[1] => Didou
[value] => Didou
[0] => Mehdi
[key] => Mehdi
)

each est typiquement utilisé en conjonction avec list pour passer en revue un tableau. Par exemple, avec $_POST :

Passer en revue $_POST avec each

```
 echo "Valeurs transmises via la méthode POST :<br />\n";
reset ($_POST);
while (list ($key, $val) = each ($_POST)) {
    echo "$key => $val<br />\n";
}
```

Après chaque each , le pointeur de tableau est déplacé au prochain élément, ou sur le dernier élément, lorsqu'on arrive à la fin.

Voir aussi key , list , current , reset , next et prev .

## 6.2.51 end() : Positionne le pointeur de tableau en fin de tableau

mixed **end** ( array array )

end déplace le pointeur interne du tableau array jusqu'au dernier élément.

Exemple avec end

```
<?php

 $fruits = array('pomme','banane','myrtille');

 print end($fruits); // myrtille

?>
```

Voir aussi current , each , prev , next et reset .

## 6.2.52 extract() : Importe les variables dans la table des symboles

int **extract** ( array var_array , *int* *extract_type* , *string* *prefix* )

extract sert à exporter un tableau vers la table des symboles. Elle prend un tableau associatif var_array , crée les variables dont les noms sont les index de ce tableau, et leur affecte la valeur associée. Pour chaque paire clé/valeur, extract crée une variable, avec les paramètres extract_type et prefix .

| Note |
| --- |
| Depuis la version 4.0.5, extract retourne le nombre de variables extraites. |

| Note |
| --- |
| EXTR_IF_EXISTS et EXTR_PREFIX_IF_EXISTS ont été introduites en PHP version 4.2.0. |

| Note |
| --- |
| EXTR_REFS a été introduite en PHP 4.3.0. |

extract vérifie l'existence de la variable avant de la créer. Le traitement des collisions est déterminé par extract_type . Ce paramètre peut prendre une des valeurs suivantes :

*EXTR_OVERWRITE*
: Lors d'une collision, réécrire la variable existante.

*EXTR_SKIP*
: Lors d'une collision, ne pas réécrire la variable existante.

*EXTR_PREFIX_SAME*
: Lors d'une collision, ajouter le préfixe prefix , et créer une nouvelle variable.

*EXTR_PREFIX_ALL*
: Ajouter le préfixe prefix , et créer une nouvelle variable.

*EXTR_PREFIX_INVALID*
: Préfixer uniquement les variables aux noms invalides ou numériques avec le préfixe prefix . Ceci a été ajouté en PHP 4.0.5.

*EXTR_IF_EXISTS*
: Ecrase la variable uniquement si elle existe déjà dans la table des symboles, et sinon, ne rien faire. Ceci est pratique pour définir une liste de variables valides, puis d'extraire du tableau les valeurs que vous avez déjà définies grâce à $_REQUEST , par exemple. Cette option a été ajoutée en PHP 4.2.0.

*EXTR_PREFIX_IF_EXISTS*
: Ne crée que des variables préfixées, si la version non préfixée de la même variable existe dans la table des symboles courante. Cette option a été ajoutée en PHP 4.2.0.

*EXTR_REFS*
: Extrait les variables sous forme de références. Cela signifie que les valeurs des variables importées font toujours référence aux valeurs des variables d'origine dans le paramètre var_array . Vous pouvez utiliser cette option seule, ou bien la combiner avec d'autres option avec l'opérateur OR, dans le paramètre extract_type . Cette option a été ajoutée en PHP 4.3.0.

Si extract_type est omis, extract utilise EXTR_OVERWRITE par défaut.

Notez que prefix n'est nécessaire que pour les valeurs de extract_type suivantes : EXTR_PREFIX_SAME , EXTR_PREFIX_ALL , EXTR_PREFIX_INVALID ou EXTR_PREFIX_IF_EXISTS . Le résultat préfixé n'est pas un nom de variable valide, il ne sera pas importé dans la table des symboles.

extract retourne le nombre de variables réellement importées dans la table des symboles.

Une utilisation possible de la fonction extract est l'exportation vers la table des symboles de tableaux de variables retournés par wddx_deserialize .

Exemple avec extract

```
<?php
/* Supposons que $var_array est un tableau retourné
    par wddx_deserialize() */
  $taille = "grand";
  $var_array = array("couleur" => "bleu",
                     "taille"  => "moyen",
                     "forme" => "sphere");
  extract($var_array, EXTR_PREFIX_SAME, "wddx");
  print "$couleur, $taille, $forme, $wddx_taille\n";
?>
```

L'exemple ci−dessus affiche :

```
bleu, grand, sphere, moyen
```

La variable $taille n'a pas été réécrite, car on avait spécifié le paramètre EXTR_PREFIX_SAME , qui a permis la création $wddx_size . Si EXTR_SKIP avait été utilisé, alors $wddx_size n'aurait pas été créé. Avec EXTR_OVERWRITE , $taille aurait pris la valeur "moyen", et avec EXTR_PREFIX_ALL , les variables créées seraient $wddx_couleur , $wddx_taille et $wddx_forme .

Vous devez utiliser un tableau associatif. Un tableau indexé numériquement ne produira aucun résultat, à moins que vous n'utilisiez l'option EXTR_PREFIX_ALL ou EXTR_PREFIX_INVALID .

Voir aussi compact .

## 6.2.53 in_array() : Indique si une valeur appartient à un tableau

bool **in_array** ( mixed needle , array haystack , *bool strict* )

in_array recherche needle dans haystack et retourne TRUE s'il s'y trouve, ou FALSE sinon.

Le troisième paramètre strict est optionnel. S'il vaut TRUE alors in_array vérifiera aussi que le types du paramètre needle correspond au type de la valeur trouvée dans haystack .

Note

Si needle est une chaîne, la comparaison est faîte en tenant compte de la casse.

Note

Dans les versions antérieure à 4.2.0, needle ne pouvait pas être un tableau.

Exemple avec in_array

```
<?php
 $os = array ("Mac", "NT", "Irix", "Linux");
 if (in_array ("Linux", $os)) {
   print "Trouvé Linux";
 }
 if (in_array ("mac", $os)) {
   print "Trouvé mac";
```

```
  }
?>
```

La seconde condition échoue, car in_array est sensible à la casse. Le script retourne :
Trouvé Linux

Exemple avec in_array et le mode strict

```
 <?php
$a = array('1.10', 12.4, 1.13);

if (in_array('12.4', $a, TRUE))
    echo "'12.4'  a été trouvé avec recherche stricte\n";
if (in_array(1.13, $a, TRUE))
    echo "1.13  a été trouvé avec recherche stricte\n";
?>
```

Ce script va afficher :
1.13 a été trouvé avec recherche stricte

Exemple avec in_array et deux tableaux en paramètres

```
 <?php
$a = array(array('p', 'h'), array('p', 'r'), 'o');

if (in_array(array ('p', 'h'), $a))
    echo "'ph' a été trouvé\n";
if (in_array(array ('f', 'i'), $a))
    echo "'fi' a été trouvé\n";
if (in_array('o', $a))
    echo "'o' a été trouvé\n";
?>
```

Ce script va afficher :
'ph' a été trouvé
'o' a été trouvé

Voir aussi array_search , array_key_exists et isset .

## 6.2.54 key() : Retourne une clé d'un tableau associatif

mixed **key** ( array  array )

key retourne l'index de la clé courante dans un tableau.

Voir aussi current et next .

## 6.2.55 krsort() :  Trie un tableau en sens inverse et suivant les clés

bool **krsort** ( array  array , *int*  *sort_flags* )

krsort trie un tableau en ordre inverse et suivant les clés, en maintenant la correspondance entre les clés et les valeurs. Cette fonction est pratique pour les tableaux associatifs.

Exemple avec krsort

```
 <?php
  $fruits = array("d"=>"papaye","a"=>"orange","b"=>"banane","c"=>"ananas");
  ksort($fruits);
  for(reset($fruits); $key = key($fruits); next($fruits)) {
    echo "fruits[$key] = ".$fruits[$key]."\n";
  }
?>
```

Cet exemple va afficher :
fruits[a] = orange

fruits[b] = banane
fruits[c] = ananas
fruits[d] = papaye

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel sort_flags . Pour plus de détails, voyez le manuel pour la fonction sort .

Voir aussi array−multisort , arsort , asort , ksort , natsort , natcasesort , rsort , sort , uasort , uksort et usort .

## 6.2.56 ksort() : Trie un tableau suivant les clés

bool **ksort** ( array array , *int* *sort_flags* )

ksort trie un tableau suivant les clés, en maintenant la correspondance entre les clés et les valeurs. Cette fonction est pratique pour les tableaux associatifs.

Exemple avec ksort

```
<?php
 $fruits = array("d"=>"papaye","a"=>"orange","b"=>"banane","c"=>"ananas");
 ksort($fruits);
 reset($fruits);
 while (list ($key, $val) = each ($fruits)) {
   echo "$key => $val\n";
 }
?>
```

Cet exemple va afficher :
a => orange
b => banane
c => ananas
d => papaye

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel sort_flags . Pour plus de détails, voyez le manuel pour la fonction sort .

Voir aussi array−multisort , arsort , asort , krsort , natsort , natcasesort , rsort , sort , uasort , uksort et usort .

Note

Le second paramètre a été ajouté en PHP 4.0.

## 6.2.57 list() : Transforme une liste de variables en tableau

void **list** ( mixed ... )

Tout comme array , list n'est pas une véritable fonction, mais un élément de langage, qui permet d'assigner une série de variables en une seule ligne.

Note

list fonctionne uniquement avec des tableaux à indexation numérique, et suppose que l'indexation commence à 0.

Exemple avec list

```
<?php

$info = array('café', 'noir', 'cafeine');
```

```
// Liste de toutes les variables
list($drink, $color, $power) = $info;
echo "$drink est $color et la $power le rend particulier.\n";

// Liste de certaines variables
list($drink, , $power) = $info;
echo "$drink contient de la $power.\n";

// Ou bien, n'utilisons que le troisième
list( , , $power) = $info;
echo "J'ai besoin de $power!\n";

?>
```

Exemple avec list et MySQL

```
 <table>
 <tr>
  <th>Nom de l'employé</th>
  <th>Salaire</th>
 </tr>

<?php

$resultat = mysql_query ("SELECT id, nom, salaire FROM employes",$conn);
while (list ($id, $nom, $salaire) = mysql_fetch_row ($resultat)) {
    echo " <tr>\n".
           "  <td><a href=\"info.php?id=$id\">$nom</a></td>\n".
           "  <td>$salaire</td>\n".
           " </tr>\n";
}

?>
</table>
```

Attention

list assigne les valeurs en commençant par la valeur la plus à droite. Si vous utiliser ces variables, ce ne sera pas un problème. Mais si vous utilisez des tableaux, vous serez surpris de voir que list les affecte de droite à gauche.

Exemple avec list en tenant compte de l'ordre

```
 <?php

$info = array('café', 'noir', 'cafeine');

list($a[0], $a[1], $a[2]) = $info;

var_dump($a);

?>
```

Cet exemple donne le résultat suivant (notez l'ordre d'affichage des éléments, en comparaison avec l'ordre dans lequel ils ont été écrits dans la syntaxe de list ) :
array(3) {
[2]=>
string(8) "cafeine"
[1]=>
string(5) "noir"
[0]=>
string(6) "café"
}

Voir aussi each , array et extract .

## 6.2.58 natcasesort() : Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse

void **natcasesort** ( array array )

natcasesort implémente un algorithme de tri qui traite les chaînes alpha−numériques comme un être humain : c'est ce qui est appelé l'"ordre naturel".

natcasesort est la version insensible à la casse de natsort . Voir aussi natsort pour un exemple illustré.

Pour plus de détails, rendez−vous sur le site de Martin Pool : Natural Order String Comparison .

Voir aussi sort , natsort , strnatcmp et strnatcasecmp .

## 6.2.59 natsort() : Tri d'un tableau avec l'algorithme à "ordre naturel"

void **natsort** ( array array )

natsort implémente un algorithme de tri qui traite les chaînes alpha−numériques comme un être humain : c'est ce qui est appelé l'"ordre naturel". Un exemple de la différence de traitement entre un tel algorithme et un algorithme de tri de chaînes (comme lorsqu'on utilise sort ) est illustré ci−dessous :

Exemple avec natsort

```
<?php
$array1 = $array2 = array ("img12.png", "img10.png", "img2.png", "img1.png");

sort($array1);
echo "Tri standard\n";
print_r($array1);

natsort($array2);
echo "Tri en ordre naturel\n";
print_r($array2);
?>
```

Le code ci−dessus va afficher :

```
Tri standard
Array
(
[0] => img1.png
[1] => img10.png
[2] => img12.png
[3] => img2.png
)
Tri en ordre naturel
Array
(
[3] => img1.png
[2] => img2.png
[1] => img10.png
[0] => img12.png
)
```

Pour plus de détails, rendez−vous sur le site de Martin Pool : Natural Order String Comparison .

Note

Si vous souhaitez conserver la relation clé/valeur, utilisez plutôt uasort($arr, 'strnatcmp') .

Voir aussi natcasesort , strnatcmp et strnatcasecmp .

## 6.2.60 next() : Avance le pointeur interne d'un tableau

mixed **next** ( array array )

next retourne l'élément suivant du tableau array ou FALSE s'il n'y a plus d'éléments. Le pointeur de interne de tableau est avancé d'un élément.

next se comporte comme current , mais avec une différence : il avance le pointeur interne de tableau d'un élément avant de retourner la valeur qu'il pointe. Lorsque le pointeur dépasse le dernier élément, next retourne FALSE .

| Attention |
| --- |
| Si le tableau contient des éléments vides ou des zéros, next retournera FALSE pour ces éléments. Pour passer proprement en revue un tableau, il faut utiliser each . |

Exemples avec next

```
<?php
$transport = array('pied', 'velo', 'voiture', 'avion');
$mode = current($transport); // $mode = 'pied';
$mode = next($transport);    // $mode = 'velo';
$mode = next($transport);    // $mode = 'voiture';
$mode = prev($transport);    // $mode = 'velo';
$mode = end($transport);     // $mode = 'avion';
?>
```

Voir aussi current , end , prev et reset .

## 6.2.61 pos() : Retourne l'élément courant d'un tableau

mixed **pos** ( array array )

pos est une fonction alias de current .

Voir aussi end , next , prev et reset .

## 6.2.62 prev() : Recule le pointeur courant de tableau

mixed **prev** ( array array )

prev repositionne le pointeur interne du tableau array à la dernière place qu'il occupait, ou bien retourne FALSE s'il ne reste plus d'éléments.

| Attention |
| --- |
| Si le tableau contient des éléments vides, prev retournera FALSE pour ces éléments aussi. Pour passer en revue tous les éléments, utilisez plutôt each . |

prev se comporte exactement comme next , mais il fait reculer le pointeur plutôt que de l'avancer.

Exemples avec next

```
<?php
$transport = array('pied', 'velo', 'voiture', 'avion');
$mode = current($transport); // $mode = 'pied';
$mode = next($transport);    // $mode = 'velo';
$mode = next($transport);    // $mode = 'voiture';
$mode = prev($transport);    // $mode = 'velo';
$mode = end($transport);     // $mode = 'avion';
?>
```

Voir aussi current , end , next et reset .

## 6.2.63 range() : Crée un tableau contenant un intervalle d'éléments

array **range** ( int low , int high , *int step* )

range retourne un tableau contenant tous les entiers depuis low jusqu'à high , inclus. Si low est supérieur à high , la séquence sera en ordre décroissant.

| Note |
| --- |
| Nouveau paramètre<br><br>Le paramètre optionnel step a été ajouté en PHP 5.0. |

Si une valeur est donnée au paramètre step , il sera utilisé comme valeur incrémentale entre les éléments de la séquence. step doit être exprimé comme un nombre entier positif. Si il n'est pas spécifié, step vaut par défaut 1.

Exemple avec range

```
<?php
// array(0,1,2,3,4,5,6,7,8,9)
foreach(range(0, 9) as $number) {
    echo $number;
}

// La paramètre de pas (step) a été introduit en version 4.4.0
// array(0,10,20,30,40,50,60,70,80,90,100)
foreach(range(0, 100, 10) as $number) {
    echo $number;
}

// L'utilisation des caractères a été introduit en version 4.1.0
// array('a','b','c','d','e','f','g','h','i');
foreach(range('a', 'i') as $letter) {
    echo $letter;
}
// array('c','b','a');
foreach(range('c', 'a') as $letter) {
    echo $letter;
}
?>
```

| Note |
| --- |
| Avant la version PHP 4.1.0, la fonction range ne générait que des séquences d'entiers. Le support des séquences de caractères a été ajouté en PHP 4.1.0. |

| Attention |
| --- |
| Dans les versions de PHP de 4.1.0 à 4.3.2, range voyait les chaînes numériques comme des chaînes et non comme des entiers. Au lieu de cela, ils étaient utilisés comme séquence de caractères. Par exemple, "4242" était traité comme "4" . |

Voir aussi shuffle , array_fill et foreach .

## 6.2.64 reset() : Remet le pointeur interne de tableau au début

mixed **reset** ( array array )

reset replace le pointeur de tableau array au premier élément.

reset retourne la valeur du premier élément.

Voir aussi current , each , next et prev .

## 6.2.65 rsort() : Trie en ordre inverse

bool **rsort** ( array array , *int* *sort_flags* )

rsort effectue un tri en ordre décroissant (du plus grand au plus petit).

Exemple avec rsort

```
<?php
 $fruits = array("papaye","orange","banane","ananas");
 rsort($fruits);
 for (reset($fruits); list($key,$value) = each($fruits); ) {
   echo "fruits[$key] = ", $value, "\n";
}
?>
```

Cet exemple va afficher:
fruits[0] = papaye
fruits[1] = orange
fruits[2] = banane
fruits[3] = ananas
Les fruits ont été classés dans l'ordre alphabétique inverse.

Vous pouvez modifier le comportement de cette fonction en utilisant le paramètre optionnel sort_flags . Pour plus de détails, voyez le manuel pour la fonction sort .

Voir aussi arsort , asort , ksort , sort et usort .

## 6.2.66 shuffle() : Mélange les éléments d'un tableau

void **shuffle** ( array array )

shuffle mélange les éléments du tableau array .

Note

Depuis PHP 4.2.0, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec srand ou mt_srand car c'est fait automatiquement.

Exemple avec shuffle

```
<?php
$numbers = range (1,20);
srand ((float)microtime()*1000000);
shuffle ($numbers);
while (list (,$number) = each ($numbers)) {
    echo "$number ";
}
?>
```

Voir aussi arsort , asort , ksort , rsort , sort et usort .

## 6.2.67 sizeof() : Alias de count

sizeof est un alias de count .

## 6.2.68 sort() : Trie un tableau

bool **sort** ( array array , *int* *sort_flags* )

sort trie le tableau array . Les éléments seront triés du plus petit au plus grand.

Exemple avec sort

```
<?php
 $fruits = array("papaye","orange","banane","ananas");
 sort ($fruits);
 reset ($fruits);
 while (list ($key, $val) = each ($fruits)) {
   echo "fruits[".$key."] = ".$val."\n";
 }
?>
```

Cet exemple va afficher :
fruits[0] = ananas
fruits[1] = banane
fruits[2] = orange
fruits[3] = papaye
Les fruits ont été classés dans l'ordre alphabétique.

Le paramètre optionnel sort_flags peut être utilisé pour modifier le comportement de trie en utilisant ces valeurs :

Constantes de type de tri :

- SORT_REGULAR : compare les éléments normalement
- SORT_NUMERIC : compare les éléments numériquement
- SORT_STRING : compare les éléments comme des chaînes de caractères

Note

Le second paramètre a été ajouté en PHP 4.

Voir aussi arsort , asort , ksort , natsort , natcasesort , rsort , usort , array_multisort et uksort .

## 6.2.69 uasort() : Trie un tableau en utilisant une fonction de comparaison personnalisée

bool **uasort** ( array array , callback cmp_function )

uasort trie le tableau array en conservant la correspondance entre les index et leurs valeurs. uasort sert essentiellement lors de tri de tableaux associatifs où l'ordre des éléments est significatif. La fonction de comparaison utilisée cmp_function est définie par l'utilisateur.

Note

Voyez les fonctions usort et uksort pour des exemples de tris avec utilisation de fonction personnalisée.

Voir aussi usort , uksort , sort , asort , arsort , ksort et rsort .

## 6.2.70 uksort() : Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur

bool **uksort** ( array array , callback cmp_function )

uksort trie les clés du tableau array en utilisant la fonction utilisateur cmp_function . Si un tableau doit être trié avec un critère complexe, il est préférable d'utiliser uksort .

Exemple avec uksort

```
<?php
function compare ($a, $b) {
    if ($a == $b) return 0;
    return ($a > $b) ? -1 : 1;
}

$a = array (4 => "quatre", 3 => "trois", 20 => "vingt", 10 => "dix");

uksort ($a, "compare");

while (list ($key, $value) = each ($a)) {
    echo "$key: $value\n";
}
?>
```

Cet exemple affichera:
20: vingt
10: dix
4: quatre
3: trois

Voir aussi usort , uasort , sort , asort , arsort , ksort , natsort et rsort .

## 6.2.71 usort() : Trie un tableau en utilisant une fonction de comparaison personnalisée

bool **usort** ( array array , callback cmp_function )

usort va trier le tableau array avec ses valeurs, en utilisant une fonction définie par l'utilisateur. Si un tableau doit être trié avec un critère complexe, il est préférable d'utiliser cette méthode.

La fonction de comparaison cmp_function doit retourner un entier, qui sera inférieur, égal ou supérieur à zéro suivant que le premier argument est considéré comme plus petit, égal ou plus grand que le second argument. Si les deux arguments sont égaux, leur ordre est indéfini.

Note

Si deux éléments sont égaux, au sens de la fonction cmp_function , leur ordre sera indéfini. Jusqu'en PHP 4.0.6, la fonction de tri leur conserverait leur ordre original, mais le nouvel algorithme introduit en PHP 4.1.0 fait que ce n'est plus le cas, car cela serait trop coûteux.

Exemple avec usort

```
<?php
function compare($a, $b)
 {
    if ($a == $b) {
      return 0;
    }
    return ($a > $b) ? -1 : 1;
}

$a = array (3, 2, 5, 6, 1);

usort ($a, "compare");

while (list ($key, $value) = each ($a)) {
    echo "$key: $value\n";
}
?>
```

Cet exemple va afficher :

```
0: 6
1: 5
2: 3
3: 2
4: 1
```

Note

Evidemment dans ce cas trivial, rsort serait plus approprié.

Tri avec usort sur un tableau multi−dimensionnel

```
 <?php
function compare($a, $b)
 {
    return strcmp($a["fruit"], $b["fruit"]);
}

$fruits[0]["fruit"] = "citron";
$fruits[1]["fruit"] = "pommes";
$fruits[2]["fruit"] = "raisins";

usort($fruits, "compare");

while (list ($key, $value) = each ($fruits)) {
    echo "\$fruits[$key]: " . $value["fruit"] . "\n";
}
?>
```

Lors du tri de tableau multi−dimensionnel, $a et $b contiennent des références sur le premier élément du tableau.

Cet exemple va afficher :

```
$fruits[0]: raisins
$fruits[1]: pommes
$fruits[2]: citron
```

Tri avec usort sur un objet

```
 <?php
class TestObj {
    var $name;

    function TestObj($name)
     {
        $this−>name = $name;
    }

    /* Ceci est une fonction de comparaison statique */
    function cmp_obj($a, $b)
     {
        $al = strtolower($a−>name);
        $bl = strtolower($b−>name);
        if ($al == $bl) {
          return 0;
        }
        return ($al > $bl) ? +1 : −1;
    }
}

$a[] = new TestObj("c");
$a[] = new TestObj("b");
$a[] = new TestObj("d");

uasort($a, array ("TestObj", "cmp_obj"));

foreach ($a as $item) {
    print $item−>name."\n";
}
?>
```

Cet exemple va afficher :

```
b
c
d
```

Voir aussi uasort , uksort , sort , asort , arsort , ksort , natsort et rsort .

# 6.3 Calendrier

## 6.3.1 Introduction

L'extension de calendrier propose une série de fonctions qui simplifient les conversions entre les différents formats de calendrier. La référence est le nombre de jours du calendrier Julien. C'est le nombre de jours depuis une date qui commence bien au−delà des dates les plus reculées dont on a besoin (située en 4000 avant J.C.). Pour convertir une date d'un calendrier à un autre, il faut d'abord la convertir dans ce calendrier, puis convertir le résultat dans le calendrier désiré. Attention, le nombre de jours du calendrier Julien est un système très différent du calendrier Julien! Pour plus d'informations, rendez−vous sur le site de http://www.hermetic.ch/cal_stud/jdn.htm . Pour plus d'informations (en anglais), reportez−vous à http://www.boogle.com/info/cal−overview.html . Les traductions issues de ces pages seront mises entre guillemets.

## 6.3.2 Installation

Pour faire fonctionner ces fonctions avec PHP , vous devez le compiler avec l'option −−enable−calendar .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

## 6.3.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.3.4 Types de ressources

Cette extension ne définit aucune ressource.

## 6.3.5 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*CAL_GREGORIAN (entier)*
*CAL_JULIAN (entier)*
*CAL_JEWISH (entier)*
*CAL_FRENCH (entier)*
*CAL_NUM_CALS (entier)*
*CAL_DOW_DAYNO (entier)*
*CAL_DOW_SHORT (entier)*
*CAL_DOW_LONG (entier)*
*CAL_MONTH_GREGORIAN_SHORT (entier)*
*CAL_MONTH_GREGORIAN_LONG (entier)*
*CAL_MONTH_JULIAN_SHORT (entier)*
*CAL_MONTH_JULIAN_LONG (entier)*
*CAL_MONTH_JEWISH (entier)*

*CAL_MONTH_FRENCH (entier)*

Les constantes suivantes sont disponibles depuis PHP 4.3.0 :

*CAL_EASTER_DEFAULT (entier)*
*CAL_EASTER_ROMAN (entier)*
*CAL_EASTER_ALWAYS_GREGORIAN (entier)*
*CAL_EASTER_ALWAYS_JULIAN (entier)*


- Introduction
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- cal_days_in_month
- cal_from_jd
- cal_info
- cal_to_jd
- easter_date
- easter_days
- frenchtojd
- gregoriantojd
- jddayofweek
- jdmonthname
- jdtofrench
- jdtogregorian
- jdtojewish
- jdtojulian
- jdtounix
- jewishtojd
- juliantojd
- unixtojd


## 6.3.7 cal_from_jd() : Convertit le nombre de jours Julien en un calendrier et retourne des détails

array **cal_from_jd** ( int jd , int calendar )

cal_from_jd convertit le nombre de jours Julien jd en une date du calendrier calendar . Les valeurs possibles pour calendar sont CAL_GREGORIAN , CAL_JULIAN , CAL_JEWISH et CAL_FRENCH .

Voir aussi cal_to_jd .

## 6.3.8 cal_info() : Retourne des détails sur un calendrier

array **cal_info** ( *int calendar* )

cal_info retourne des informations sur le calendrier calendar spécifié, ou sur tous les calendriers supportés si l'argument calendar est omis.

Les informations de calendriers sont retournés dans un tableau qui contient les éléments calname , calsymbol , month , abbrevmonth et maxdaysinmonth .

Si le paramètre calendar n'est pas spécifié, un tableau représentant tous les calendriers est retourné. Cette fonctionnalité sera disponible en PHP 5.

## 6.3.9 cal_to_jd() : Convertit un calendrier en nombre de jours Julien

int **cal_to_jd** ( int calendar , int month , int day , int year )

cal_to_jd calcule le nombre de jours Juliens pour une date, dans le calendrier calendar . Les valeurs possibles pour calendar sont CAL_GREGORIAN , CAL_JULIAN , CAL_JEWISH et CAL_FRENCH .

Voir aussi cal_to_jd .

## 6.3.10 easter_date() : Retourne un timestamp UNIX pour Pâques, à minuit

int **easter_date** ( *int* *year* )

easter_date retourne un timestamp UNIX pour Pâques, à minuit, pour une année donnée. Si l'année n'est pas précisée, c'est l'année en cours qui est utilisée.

Depuis PHP 4.3.0, le paramètre year est optionnel, et prend la valeur de l'année en cours par défaut.

Exemples avec easter_date

```
<?php
echo date( "M-d-Y", easter_date(1999) );        /* "Apr-04-1999" */
echo date( "M-d-Y", easter_date(2000) );        /* "Apr-23-2000" */
echo date( "M-d-Y", easter_date(2001) );        /* "Apr-15-2001" */
?>
```

Attention

easter_date génère une alerte (Warning) si la date tombe hors de la zone de validité des timestamp UNIX (i.e. avant 1970 ou après 2037).

La date de Pâques a été fixée par le concile de Nicée, en 325 de notre ère, comme étant le dimanche après la première pleine lune qui suit l'équinoxe de printemps. L'équinoxe de printemps est considéré comme étant toujours le 21 mars, ce qui réduit le problème au calcul de la date de la lune pleine qui suit, et le dimanche suivant. L'algorithme fut introduit vers 532, par Dionysius Exiguus. Avec le calendrier Julien, (pour les années avant 1753), un cycle de 19 ans suffit pour connaître les date des phases de la lune. Avec le calendrier grégorien, (à partir des années 1753, conçu par Clavius et Lilius, puis introduit par le pape Grégoire XIII en octobre 1582, et en Grande Bretagne et ses colonies en septembre 1752), deux facteurs de corrections ont été ajoutés pour rendre le cycle plus précis.

(Ce code est basé sur le programme en C de Simon Kershaw, <webmaster@ely.anglican.org> )

Voir easter_days pour les calculs de date de Pâques avant 1970 et après 2037.

## 6.3.11 easter_days() : Retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée.

int **easter_days** ( *int year* , *int method* )

easter_days retourne le nombre de jours entre le 21 Mars et Pâques, pour une année donnée. Si l'année n'est pas précisée, l'année en cours est utilisée par défaut.

Depuis PHP 4.3.0, le paramètre year est optionnel, et prend la valeur de l'année en cours par défaut.

Le paramètre method a été introduit en PHP 4.3.0 et permet de calculer les dates de Pâques dans le calendrier Grégorien, entre les années 1582 − 1752, lorsque ce paramètre prend la valeur de CAL_EASTER_ROMAN . Voyez les constantes calendaires pour plus de constantes valides.

easter_days peut être utilisée à la place de easter_date pour calculer la date de Pâques, pour les années qui tombent hors de l'intervalle de validité des timestamps UNIX (i.e. avant 1970 ou après 2037).

Exemple avec easter_days

```
<?php
 echo easter_days(1999);        /* 14, i.e. 4 Avril   */
 echo easter_days(1492);        /* 32, i.e. 22 Avril  */
 echo easter_days(1913);        /*  2, i.e. 23 Mars   */
?>
```

La date de Pâques a été fixée par le concile de Nicée, en 325 de notre ère, comme étant le dimanche après la première pleine lune qui suit l'équinoxe de printemps. L'équinoxe de printemps est considéré comme étant toujours le 21 mars, ce qui réduit le problème au calcul de la date de la lune pleine qui suit, et le dimanche suivant. L'algorithme fut introduit vers 532, par Dionysius Exiguus. Avec le calendrier Julien, (pour les années avant 1753), un cycle de 19 ans suffit pour connaître les date des phases de la lune. Avec le calendrier grégorien, (à partir des années 1753, conçu par Clavius et Lilius, puis introduit par le pape Grégoire XIII en octobre 1582, et en Grande Bretagne et ses colonies en septembre 1752), deux facteurs de corrections ont été ajoutés pour rendre le cycle plus précis.

(Ce code est basé sur le programme en C de Simon Kershaw, <webmaster@ely.anglican.org> )

Voir aussi easter_date .

## 6.3.12 frenchtojd() : Convertit une date du calendrier français républicain en nombre de jours du calendrier Julien.

int **frenchtojd** ( int month , int day , int year )

frenchtojd convertit une date du calendrier français républicain en nombre de jours du calendrier Julien.

Ces fonctions convertissent les dates comprises entre l'an 1 et l'an 14 (22 Septembre 1792 à 22 Septembre 1806 en calendrier grégorien). Cela couvre plus que la durée d'existence de ce calendrier.

## 6.3.13 gregoriantojd() : Convertit une date grégorienne en nombre de jours du calendrier Julien

int **gregoriantojd** ( int month , int day , int year )

Intervalle de validité pour le calendrier grégorien : 4714 avant JC à 9999 après JC.A.D.

Bien qu'il soit possible de manipuler des dates jusqu'en 4714 avant JC, une telle utilisation n'est pas significative. En effet, ce calendrier fut créé le 18 octobre 1582 après J.C. (ou 5 octobre 1582 en calendrier grec). Certains pays ne l'acceptèrent que bien plus tard. Par exemple, les britanniques n'y passèrent en 1752, les Russes en 1918 et les Grecs en 1923. La plupart des pays européens utilisaient le calendrier Julien avant le Grégorien.

| Fonctions calendrier |
|---|
| `<?php`<br>`  $jd = gregoriantojd(10,11,1970);`<br>`  echo("$jd\n");`<br>`  $gregorian = jdtogregorian($jd);`<br>`  echo("$gregorian\n");`<br>`?>` |

## 6.3.14 jddayofweek() : Retourne le numéro du jour de la semaine.

mixed **jddayofweek** ( int julianday , int mode )

jddayofweek retourne le numéro du jour de la semaine. Peut retourner une chaîne ou un entier, en fonction du mode.

| Mode | Signification |
|---|---|
| 0 | Retourne le numéro du jour comme un entier (0=dimanche, 1=lundi, etc.) |
| 1 | Retourne une chaîne contenant le nom du jour (anglais grégorien) |
| 2 | Retourne une chaîne contenant le nom abrégé du jour de la semaine (anglais grégorien). |

## 6.3.15 jdmonthname() : Retourne le nom du mois.

string **jdmonthname** ( int julianday , int mode )

jdmonthname retourne une chaîne contenant le nom du mois. mode indique de quel calendrier dépend ce mois, et quel type de nom doit être retourné.

| Mode | Signification |
|---|---|
| 0 | Grégorien – abrégé |
| 1 | Grégorien |
| 2 | Julien – abrégé |
| 3 | Julien |
| 4 | Juif |
| 5 | Républicain français |

## 6.3.16 jdtofrench() : Convertit le nombre de jours du calendrier Julien en date du calendrier français républicain

string **jdtofrench** ( int juliandaycount )

jdtofrench convertit le nombre de jours du calendrier julien en date du calendrier français républicain.

## 6.3.17 jdtogregorian() : Convertit le nombre de jours du calendrier Julien en date grégorienne.

string **jdtogregorian** ( int julianday )

jdtogregorian convertit le nombre de jours du calendrier Julien en une chaîne contenant une date du calendrier grégorien, au format "mois/jour/année".

## 6.3.18 jdtojewish() : Convertit le nombre de jours du calendrier Julien en date du calendrier juif.

string **jdtojewish** ( int juliandaycount , *bool hebrew* , *int fl* )

jdtojewish convertit le nombre de jours du calendrier Julien en date du calendrier juif.

Les paramètres optionnels hebrew et fl sont disponibles depuis PHP 5.0.0.

Si le paramètre hebrew vaut TRUE , le paramètre fl sera utilisé pour générer une chaîne au format hébreux. Les formats disponibles sont : CAL_JEWISH_ADD_ALAFIM_GERESH , CAL_JEWISH_ADD_ALAFIM et CAL_JEWISH_ADD_GERESHAYIM .

Exemple avec jdtojewish

```
<?php
 echo jdtojewish(gregoriantojd(10,8,2002), true,
      CAL_JEWISH_ADD_GERESHAYIM + CAL_JEWISH_ADD_ALAFIM + CAL_JEWISH_ADD_ALAFIM_GERESH);
?>
```

## 6.3.19 jdtojulian() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien.

string **jdtojulian** ( int julianday )

jdtojulian convertit le nombre de jours du calendrier Julien en une chaîne contenant la date du calendrier Julien, au format "mois/jour/année".

## 6.3.20 jdtounix() : Convertit un nombre de jours Julien en timestamp UNIX

int **jdtounix** ( int jday )

jdtounix retourne un timestamp UNIX correspondant au nombre de jours Julien jday ou FALSE si jday n'est pas dans l'intervalle de validité de l'époque UNIX. (années grégoriennes entre 1970 et 2037 ou 2440588 <= jday <= 2465342 ).

Voir aussi unixtojd .

## 6.3.21 jewishtojd() : Convertit une date du calendrier juif en nombre de jours du calendrier Julien.

int **jewishtojd** ( int month , int day , int year )

Bien qu'il soit possible de manipuler des dates à partir de l'an 1 (3761 avant J.C.), une telle utilisation a peu de sens.

Le calendrier juif a été utilisé depuis plusieurs dizaines de siècles, mais dans les premiers temps, il n'y avait pas de formule pour déterminer le début du mois. Un nouveau mois commençait quand une nouvelle lune était observée.

## 6.3.22 juliantojd() : Convertit le nombre de jours du calendrier Julien en date du calendrier Julien.

int **juliantojd** ( int month , int day , int year )

Intervalle de validité du calendrier Julien : 4713 avant JC à 9999 après J.C..

Bien qu'il soit possible de manipuler des dates jusqu'en 4713 avant JC, une telle utilisation n'est pas significative. En effet, ce calendrier fut créé en 46 avant J.C., et ses détails ne furent finalisés qu'au plus tôt en 8 après JC, et probablement pas avant le 4ième siècle après JC. De plus, le début de l'année variait suivant les peuples, certains n'acceptant pas janvier comme premier mois de l'année.

## 6.3.23 unixtojd() : Convertit un timestamp UNIX en nombre de jours Julien

int **unixtojd** ( *int timestamp* )

unixtojd retourne le nombre de jours Julien du timestamp UNIX timestamp (nombre de secondes depuis le 1/1/1970), ou pour le jour courant si timestamp est omis.

Voir aussi jdtounix .

# 6.4 Classes et Objets

## 6.4.1 Introduction

Ces fonctions vous donnent des informations sur les objets et les classes. Vous pouvez lire le nom de la classe qui est instanciée par un objet, ainsi que la liste des membres et des méthodes. En utilisant ces fonctions, vous pouvez non seulement connaitre la classe d'un objet, mais aussi tous ses ascendants (i.e. la classe qui est étendue par une autre classe).

## 6.4.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.4.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.4.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.4.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.4.6 Constantes prédefinies

Cette extension ne définit aucune constante.

## 6.4.7 Exemples

Dans cet exemple, on définit une classe de base, et une extension. La classe de base définit un légume, s'il est mangeable ou pas et sa couleur. La sous−classe epinard ajoute une méthode pour le cuisiner, et une autre pour savoir s'il est cuisiné.

Fonctions d'objets : classes.inc

```
<?php

// classe de base, avec ses membres et ses méthodes
class Legume {

    var $edible;
    var $color;

    function Legume( $edible, $color="green" ) {
        $this->edible = $edible;
        $this->color = $color;
    }

    function is_edible() {
        return $this->edible;
    }
```

```
    function what_color() {
        return $this->color;
    }

} // fin de la classe Legume

// Extension la classe de base
class Epinard extends Legume {
    var $cuit = FALSE;
    function Epinard() {
        $this->Legume( TRUE, "green" );
    }
    function cuisine() {
        $this->cuit = TRUE;
    }

    function is_cooked() {
        return $this->cooked;
    }

} // fin de la classe Epinard

?>
```

Lorsqu'on instantie deux objets de ces classes et que l'on affiche leurs informations, on affiche aussi leur héritage. On définit ici des utilitaires qui servent essentiellement à afficher ces informations proprement.

Fonctions d'objets : test_script.php

```
 <pre>
<?php

include "classes.inc";

// Fonctions utilitaires
function print_vars($obj) {
    $arr = get_object_vars($obj);
    while (list($prop, $val) = each($arr))
        echo "\t$prop = $val\n";
}

function print_methods($obj) {
    $arr = get_class_methods(get_class($obj));
    foreach ($arr as $method)
        echo "\tfonction $method()\n";
}

function class_parentage($obj, $class) {
    if (is_subclass_of($GLOBALS[$obj], $class)) {
        echo "L'objet $obj appartient à la classe ".get_class($$obj);
        echo " est une sous-classe de $class\n";
    } else {
        echo "L'objet $obj n'est pas une sous-classe de $class\n";
    }
}

// instantie 2 objets

$veggie = new Legume(true,"blue");
$leafy = new Epinard();

// affiche les informations sur ces objets
echo "légume : CLASS ".get_class($veggie)."\n";
echo "feuillues : CLASS ".get_class($leafy);
echo ", PARENT ".get_parent_class($leafy)."\n";

// affiche les propriétés du légume
echo "\nlégumes : Propriétés\n";
print_vars($veggie);

// et les méthodes des feuillus
echo "\nfeuillus : Méthodes\n";
print_methods($leafy);

  echo "\nParentée:\n";
class_parentage("leafy", "Epinard");
class_parentage("leafy", "Legume");
?>
</pre>
```

Il est important de noter que dans les exemples ci-dessus, les objets $feuilles sont une instance de Epinard qui est une sous-classe de Legume , donc la dernière partie du script va afficher :

```
[...]
Parentée:
L'objet feuilles n'est pas une sous classe epinard
L'objet feuilles est une sous-classe de Legume
```

- Introduction
- Pré-requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- call_user_method_array
- call_user_method
- class_exists
- get_class_methods
- get_class_vars
- get_class
- get_declared_classes
- get_object_vars
- get_parent_class
- is_a
- is_subclass_of
- method_exists

## 6.4.9 call_user_method() : Appelle une méthode utilisateur d'un objet

mixed **call_user_method** ( string method_name , object obj , *mixed parameter* , *mixed ...* )

call_user_method appelle la méthode method_name de l'objet obj . Un exemple d'utilisation de cet objet est présenté ci-dessous, où une classe est définie, puis instantiée. On utilise alors call_user_method pour appeler indirectement les méthodes print_info .

Exemple avec call_user_method

```php
<?php
class Country {
    var $NAME;
    var $TLD;

    function Country($name, $tld) {
        $this->NAME = $name;
        $this->TLD = $tld;
    }

    function print_info($prestr="") {
        echo $prestr."Pays : ".$this->NAME."\n";
        echo $prestr."Top Level Domain : ".$this->TLD."\n";
    }
}

$cntry = new Country("Perou","pe");

echo "* Appel de la méthode de l'objet directement\n";
$cntry->print_info();

echo "\n* Appel de la méthode de l'objet indirectement\n";
call_user_method ("print_info", $cntry, "\t");
?>
```

Voir aussi call_user_func_array . call_user_func et call_user_method_array .

## 6.4.10 class_exists() : Vérifie qu'une classe a été définie

bool **class_exists** ( string class_name )

class_exists retourne TRUE si la classe class_name a été définie, et FALSE sinon.

## 6.4.11 get_class_methods() : Retourne les noms des méthodes d'une classe

array **get_class_methods** ( mixed class_name )

get_class_methods retourne un tableau contenant les noms des méthodes de la classe class_name .

Note

A partir de PHP 4.0.6, vous pouvez spécifier l'objet lui−même, au lieu de sa classe class_name . Par exemple :

```
<?php
 $class_methods = get_class_methods($my_class);
?>
```

Exemple avec get_class_methods

```
<?php
class maclasse {
    // constructeur
    function maclasse() {
        return(TRUE);
    }
    // méthode 1
    function myfunc1() {
        return(TRUE);
    }
    // méthode 2
    function mafunc2() {
        return(TRUE);
    }
}
$ma_classe = new maclasse();

$class_methods = get_class_methods(get_class($ma_classe));

foreach ($class_methods as $method_name) {
    echo "$method_name\n";
}
?>
```

Va afficher :

```
maclasse
myfunc1
mafunc2
```

Voir aussi get_class_vars et get_object_vars

## 6.4.12 get_class_vars() : Retourne les valeurs par défaut des attributs d'une classe.

array **get_class_vars** ( string class_name )

get_class_vars retourne un tableau contenant les valeurs par défaut des attributs de la classe class_name .

Note

Avant PHP 4.2.0, les variables de classe non initialisées n'étaient pas rapportées par get_class_vars .

Exemple avec get_class_vars

```
<?php

class myclass {

    var $var1; // this has no default value...
    var $var2 = "xyz";
    var $var3 = 100;

    // constructeur
    function myclass() {
        return(TRUE);
    }

}

$my_class = new myclass();

$class_vars = get_class_vars(get_class($my_class));

foreach ($class_vars as $name => $value) {
    echo "$name : $value\n";
}

?>
```

va afficher :

```
// Avant PHP 4.2.0
var2 : xyz
var3 : 100

// Depuis PHP 4.2.0
var1 :
var2 : xyz
var3 : 100
```

## 6.4.13 get_class() : Retourne la classe d'un objet

string **get_class** ( object obj )

get_class retourne la classe de l'objet obj . get_class retourne FALSE si obj n'est pas un objet.

Note

get_class retourne un nom de classe utilisateur, en minuscules. Une classe définie par une extension PHP est retournée dans sa notation originale.

Exemple avec get_class

```
<?php

class foo {
    function foo() {
    // un peu de code
    }

    function name() {
        echo "Mon nom est " , get_class($this) , "\n";
    }
}

// create an object
$bar = new foo();

// external call
echo "Son nom est " , get_class($bar) , "\n";

// internal call
$bar->name();

?>
```

Le résultat sera :

```
Son nom est foo
Mon nom est foo
```

Voir aussi gettype , get_parent_class et is_subclass_of

Voir aussi get_parent_class , gettype et is_subclass_of .

## 6.4.14 get_declared_classes() : Liste toutes les classes définies

array **get_declared_classes** ( void )

get_declared_classes retourne un tableau contenant la liste des fonctions déclarées dans le script courant.

Note

En PHP 4.0.1pl2, trois classes supplémentaires sont retournées, au début de ce tableau : stdClass (définie dans Zend/zend.c ), OverloadedTestClass (définie dans ext/standard/basic_functions.c ) et Directory (définie dans ext/standard/dir.c ).

## 6.4.15 get_object_vars() : Retourne un tableau associatif des propriétés d'un objet

array **get_object_vars** ( object obj )

get_object_vars retourne un tableau associatif contenant les propriétés de l'objet obj . Les clés du tableau sont les noms des propriétés de l'objet.

Note

Dans les versions antérieures à la PHP 4.2.0, si les variables déclarées dans la classe de l'objet obj , n'avaient pas été assignées, elles n'apparaissaient pas dans le tableau retourné. Depuis PHP 4.2.0, elles apparaissent, avec la valeur NULL .

Exemple avec get_object_vars

```php
<?php
class Point2D {
    var $x, $y;
    var $label;

    function Point2D($x, $y) {
        $this->x = $x;
        $this->y = $y;
    }

    function setLabel($label) {
        $this->label = $label;
    }

    function getPoint() {
        return array("x" => $this->x,
                     "y" => $this->y,
                     "label" => $this->label);
    }
}

// "$label" est déclaré mais non défini
$p1 = new Point2D(1.233, 3.445);
print_r(get_object_vars($p1));

$p1->setLabel("point #1");
print_r(get_object_vars($p1));

?>
```

Le script ci−dessus va afficher :
Array
(
[x] => 1.233
[y] => 3.445
[label] =>
)

Array
(
[x] => 1.233
[y] => 3.445
[label] => point #1
)

Voir aussi get_class_methods et get_class_vars

## 6.4.16 get_parent_class() : Retourne le nom de la classe d'un objet

string **get_parent_class** ( mixed obj )

get_parent_class retourne le nom de la classe de l'objet obj .

Si le paramètre obj est une chaîne, get_parent_class retourne le nom de la classe parente. Cette fonctionnalité a été ajoutée en PHP 4.0.5.

Exemple avec get_parent_class

```
<?php

class papa {
    function papa() {
    // un peu de code
    }
}

class child extends papa {
    function child() {
        echo "je suis le fils de " , get_parent_class($this) , "\n";
    }
}

class child2 extends papa {
    function child2() {
        echo "Je suis aussi le papa de " , get_parent_class('child2') , "\n";
    }
}

$foo = new child();
$bar = new child2();

?>
```

The output is:

```
je suis le fils de papa
Je suis aussi le papa de papa
```

Voir aussi get_class et is_subclass_of .

Voir aussi get_class et is_subclass_of .

### 6.4.17 is_a() : Retourne TRUE si un objet a pour parent une classe donnée

bool **is_a** ( object object , string class_name )

is_a retourne TRUE si l'objet object a pour classe class_name , ou bien si class_name est un de ses parents. is_a retourne FALSE sinon.

Voir aussi get_class , get_parent_class et is_subclass_of .

### 6.4.18 is_subclass_of() : Détermine si un objet est une sous−classe

bool **is_subclass_of** ( object object , string class_name )

is_subclass_of retourne TRUE si l'objet object est une sous−classe de class_name , FALSE sinon.

Voir aussi get_class et get_parent_class

### 6.4.19 method_exists() : Vérifie que la méthode existe pour une classe.

bool **method_exists** ( object object , string method_name )

method_exists retourne TRUE si la méthode method_name a été définie pour la classe object , et sinon, retourne FALSE .

# 6.5 CURL

## 6.5.1 Introduction

PHP supporte libcurl, une librairie créée par Daniel Stenberg, qui vous permet de vous connecter, de communiquer avec de nombreux serveurs, grâce à de nombreux protocoles. libcurl supporte actuellement les protocoles suivants : HTTP , HTTPS , FTP , gopher , telnet , dict , file , et LDAP . libcurl supporte aussi les certificats HTTPS, les POST HTTP, PUT HTTP, le chargement par FTP (ce qui peut être fait par l'extension FTP), les chargements par formulaire HTTP, les proxies, les cookies et l'authentification par mot de passe et nom de compte.

Ces fonctions ont été ajoutées en PHP 4.0.2.

## 6.5.2 Pré−requis

Pour pouvoir utiliser les fonctions CURL, vous devez installer le package CURL . PHP requiert la version CURL 7.0.2−beta ou plus récente. PHP ne fonctionnera pas avec une version inférieure à la version 7.0.2−beta. A partir de PHP 4.2.3, vous aurez besoin de la version 7.9.0 ou plus récente.

## 6.5.3 Installation

Pour utiliser CURL depuis les scripts PHP, vous devez aussi compiler PHP avec l'option −−with−curl[=DIR] où DIR est le chemin jusqu'au dossier contenant les dossiers lib et include . Dans le dossier include il doit se trouver un dossier appelé curl , qui contient notamment les fichiers easy.h et curl.h . Il doit aussi se trouver un fichier nommé libcurl.a dans le dossier lib . A partir de PHP 4.3.0, vous pouvez aussi configurer PHP pour qu'il utilise curl comme gestionnaire d'URL avec l'option −−with−curlwrappers .

| Note |
| --- |
| Note aux utilisateurs Win32<br><br>Afin d'activer ce module dans l'environnement Windows, vous devez copier les librairies libeay32.dll et ssleay32.dll depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : C:\WINNT\SYSTEM32 ou C:\WINDOWS\SYSTEM ). |

## 6.5.4 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*CURLOPT_PORT (entier)*
*CURLOPT_FILE (entier)*
*CURLOPT_INFILE (entier)*
*CURLOPT_INFILESIZE (entier)*
*CURLOPT_URL (entier)*
*CURLOPT_PROXY (entier)*
*CURLOPT_VERBOSE (entier)*
*CURLOPT_HEADER (entier)*
*CURLOPT_HTTPHEADER (entier)*

*CURLOPT_NOPROGRESS (entier)*
*CURLOPT_NOBODY (entier)*
*CURLOPT_FAILONERROR (entier)*
*CURLOPT_UPLOAD (entier)*
*CURLOPT_POST (entier)*
*CURLOPT_FTPLISTONLY (entier)*
*CURLOPT_FTPAPPEND (entier)*
*CURLOPT_NETRC (entier)*
*CURLOPT_FOLLOWLOCATION (entier)*
*CURLOPT_FTPASCII (entier)*
*CURLOPT_PUT (entier)*
*CURLOPT_MUTE (entier)*
*CURLOPT_USERPWD (entier)*
*CURLOPT_PROXYUSERPWD (entier)*
*CURLOPT_RANGE (entier)*
*CURLOPT_TIMEOUT (entier)*
*CURLOPT_POSTFIELDS (entier)*
*CURLOPT_REFERER (entier)*
*CURLOPT_USERAGENT (entier)*
*CURLOPT_FTPPORT (entier)*
*CURLOPT_LOW_SPEED_LIMIT (entier)*
*CURLOPT_LOW_SPEED_TIME (entier)*
*CURLOPT_RESUME_FROM (entier)*
*CURLOPT_COOKIE (entier)*
*CURLOPT_SSLCERT (entier)*
*CURLOPT_SSLCERTPASSWD (entier)*
*CURLOPT_WRITEHEADER (entier)*
*CURLOPT_SSL_VERIFYHOST (entier)*
*CURLOPT_COOKIEFILE (entier)*
*CURLOPT_SSLVERSION (entier)*
*CURLOPT_TIMECONDITION (entier)*
*CURLOPT_TIMEVALUE (entier)*
*CURLOPT_CUSTOMREQUEST (entier)*
*CURLOPT_STDERR (entier)*
*CURLOPT_TRANSFERTEXT (entier)*
*CURLOPT_RETURNTRANSFER (entier)*
*CURLOPT_QUOTE (entier)*
*CURLOPT_POSTQUOTE (entier)*
*CURLOPT_INTERFACE (entier)*
*CURLOPT_KRB4LEVEL (entier)*
*CURLOPT_HTTPPROXYTUNNEL (entier)*
*CURLOPT_FILETIME (entier)*
*CURLOPT_WRITEFUNCTION (entier)*
*CURLOPT_READFUNCTION (entier)*
*CURLOPT_PASSWDFUNCTION (entier)*
*CURLOPT_HEADERFUNCTION (entier)*
*CURLOPT_MAXREDIRS (entier)*
*CURLOPT_MAXCONNECTS (entier)*
*CURLOPT_CLOSEPOLICY (entier)*
*CURLOPT_FRESH_CONNECT (entier)*
*CURLOPT_FORBID_REUSE (entier)*
*CURLOPT_RANDOM_FILE (entier)*
*CURLOPT_EGDSOCKET (entier)*
*CURLOPT_CONNECTTIMEOUT (entier)*

*CURLOPT_SSL_VERIFYPEER (entier)*
*CURLOPT_CAINFO (entier)*
*CURLOPT_COOKIEJAR (entier)*
*CURLOPT_SSL_CIPHER_LIST (entier)*
*CURLOPT_BINARYTRANSFER (entier)*
*CURLCLOSEPOLICY_LEAST_RECENTLY_USED (entier)*
*CURLCLOSEPOLICY_LEAST_TRAFFIC (entier)*
*CURLCLOSEPOLICY_SLOWEST (entier)*
*CURLCLOSEPOLICY_CALLBACK (entier)*
*CURLCLOSEPOLICY_OLDEST (entier)*
*CURLINFO_EFFECTIVE_URL (entier)*
*CURLINFO_HTTP_CODE (entier)*
*CURLINFO_HEADER_SIZE (entier)*
*CURLINFO_REQUEST_SIZE (entier)*
*CURLINFO_TOTAL_TIME (entier)*
*CURLINFO_NAMELOOKUP_TIME (entier)*
*CURLINFO_CONNECT_TIME (entier)*
*CURLINFO_PRETRANSFER_TIME (entier)*
*CURLINFO_SIZE_UPLOAD (entier)*
*CURLINFO_SIZE_DOWNLOAD (entier)*
*CURLINFO_SPEED_DOWNLOAD (entier)*
*CURLINFO_SPEED_UPLOAD (entier)*
*CURLINFO_FILETIME (entier)*
*CURLINFO_SSL_VERIFYRESULT (entier)*
*CURLINFO_CONTENT_LENGTH_DOWNLOAD (entier)*
*CURLINFO_CONTENT_LENGTH_UPLOAD (entier)*
*CURLE_OK (entier)*
*CURLE_UNSUPPORTED_PROTOCOL (entier)*
*CURLE_FAILED_INIT (entier)*
*CURLE_URL_MALFORMAT (entier)*
*CURLE_URL_MALFORMAT_USER (entier)*
*CURLE_COULDNT_RESOLVE_PROXY (entier)*
*CURLE_COULDNT_RESOLVE_HOST (entier)*
*CURLE_COULDNT_CONNECT (entier)*
*CURLE_FTP_WEIRD_SERVER_REPLY (entier)*
*CURLE_FTP_ACCESS_DENIED (entier)*
*CURLE_FTP_USER_PASSWORD_INCORRECT (entier)*
*CURLE_FTP_WEIRD_PASS_REPLY (entier)*
*CURLE_FTP_WEIRD_USER_REPLY (entier)*
*CURLE_FTP_WEIRD_PASV_REPLY (entier)*
*CURLE_FTP_WEIRD_227_FORMAT (entier)*
*CURLE_FTP_CANT_GET_HOST (entier)*
*CURLE_FTP_CANT_RECONNECT (entier)*
*CURLE_FTP_COULDNT_SET_BINARY (entier)*
*CURLE_PARTIAL_FILE (entier)*
*CURLE_FTP_COULDNT_RETR_FILE (entier)*
*CURLE_FTP_WRITE_ERROR (entier)*
*CURLE_FTP_QUOTE_ERROR (entier)*
*CURLE_HTTP_NOT_FOUND (entier)*
*CURLE_WRITE_ERROR (entier)*
*CURLE_MALFORMAT_USER (entier)*
*CURLE_FTP_COULDNT_STOR_FILE (entier)*
*CURLE_READ_ERROR (entier)*
*CURLE_OUT_OF_MEMORY (entier)*

*CURLE_OPERATION_TIMEOUTED (entier)*
*CURLE_FTP_COULDNT_SET_ASCII (entier)*
*CURLE_FTP_PORT_FAILED (entier)*
*CURLE_FTP_COULDNT_USE_REST (entier)*
*CURLE_FTP_COULDNT_GET_SIZE (entier)*
*CURLE_HTTP_RANGE_ERROR (entier)*
*CURLE_HTTP_POST_ERROR (entier)*
*CURLE_SSL_CONNECT_ERROR (entier)*
*CURLE_FTP_BAD_DOWNLOAD_RESUME (entier)*
*CURLE_FILE_COULDNT_READ_FILE (entier)*
*CURLE_LDAP_CANNOT_BIND (entier)*
*CURLE_LDAP_SEARCH_FAILED (entier)*
*CURLE_LIBRARY_NOT_FOUND (entier)*
*CURLE_FUNCTION_NOT_FOUND (entier)*
*CURLE_ABORTED_BY_CALLBACK (entier)*
*CURLE_BAD_FUNCTION_ARGUMENT (entier)*
*CURLE_BAD_CALLING_ORDER (entier)*
*CURLE_HTTP_PORT_FAILED (entier)*
*CURLE_BAD_PASSWORD_ENTERED (entier)*
*CURLE_TOO_MANY_REDIRECTS (entier)*
*CURLE_UNKNOWN_TELNET_OPTION (entier)*
*CURLE_TELNET_OPTION_SYNTAX (entier)*
*CURLE_OBSOLETE (entier)*
*CURLE_SSL_PEER_CERTIFICATE (entier)*

## 6.5.5 Exemples

Une fois que vous avez compilé PHP avec le support CURL, vous pouvez commencer à l'exploiter avec vos scripts PHP. Le principe de fonctionnement est d'initialiser une session CURL avec curl_init , puis de choisir toutes vos options de transfert avec curl_exec et de finir votre session avec curl_close . Voici un exemple d'utilisation des fonctions CURL, qui récupère la page principale de PHP :

Utilisation de CURL et PHP pour récupérer une page

```
<?php

$ch = curl_init ("http://www.nexen.net/");
$fp = fopen ("example_homepage.txt", "w");

curl_setopt ($ch, CURLOPT_FILE, $fp);
curl_setopt ($ch, CURLOPT_HEADER, 0);

curl_exec ($ch);
curl_close ($ch);
fclose ($fp);
?>
```


- Introduction
- Pré−requis
- Installation
- Constantes prédefinies
- Exemples
- curl_close
- curl_errno
- curl_error
- curl_exec
- curl_getinfo
- curl_init

- curl_multi_add_handle
- curl_multi_close
- curl_multi_exec
- curl_multi_getcontent
- curl_multi_info_read
- curl_multi_init
- curl_multi_remove_handle
- curl_multi_select
- curl_setopt
- curl_version

### 6.5.7 curl_errno() : Retourne un entier représentant le dernier message d'erreur CURL

int **curl_errno** ( resource ch )

curl_errno retourne le numéro d'erreur de la dernière opération cURL, avec la ressource ch , ou 0 (zéro) si aucune erreur n'est survenue.

Voir aussi curl_error .

### 6.5.8 curl_error() : Retourne une chaîne contenant le dernier message d'erreur CURL

string **curl_error** ( resource ch )

curl_error retourne un message clair représentant la dernière erreur qui est survenue avec la ressource ch ou '' (la chaîne vide) si aucune erreur n'est survenue.

Voir aussi curl_errno .

### 6.5.9 curl_exec() : Exécute une session CURL

bool **curl_exec** ( resource ch )

curl_exec doit être appelée après l'initialisation et le paramétrage d'une session CURL. Son but est simplement d'exécuter la session ch .

Array

### 6.5.10 curl_getinfo() : Lit les informations détaillant un transfert CURL

string **curl_getinfo** ( resource ch , *int opt* )

curl_getinfo lit les informations détaillant le transfert identifié par la ressource ch . Le paramètre opt peut prendre l'une des valeurs suivantes :

- CURLINFO_EFFECTIVE_URL − Dernière URL réelle
- CURLINFO_HTTP_CODE − Dernier code HTTP reçu

- CURLINFO_FILETIME − Date distante du document, et −1 si la date du document distant est inconnue.
- CURLINFO_TOTAL_TIME − Durée de la transaction en secondes pour le dernier transfert
- CURLINFO_NAMELOOKUP_TIME − Durée de résolution du nom de domaine en secondes
- CURLINFO_CONNECT_TIME − Durée d'établissement de la connexion en secondes
- CURLINFO_PRETRANSFER_TIME − Durée en secondes, entre le début de la transaction et de début du transfert de fichiers
- CURLINFO_STARTTRANSFER_TIME − Durée en secondes jusqu'à ce que le premier octet soit sur le point d'être transféré
- CURLINFO_REDIRECT_TIME − Durée en secondes de toutes les étapes de redirection avant que la transaction finale ne soit débutée
- CURLINFO_SIZE_UPLOAD − Nombre total d'octets envoyés
- CURLINFO_SIZE_DOWNLOAD − Nombre total d'octets téléchargés
- CURLINFO_SPEED_DOWNLOAD − Vitesse moyenne de téléchargement
- CURLINFO_SPEED_UPLOAD − Vitesse moyenne d'envoi
- CURLINFO_HEADER_SIZE − Taille des en−têtes reçus
- CURLINFO_REQUEST_SIZE − Taille totale des requêtes envoyées. Actuellement, uniquement pour les requêtes HTTP
- CURLINFO_SSL_VERIFYRESULT − Résultat de la vérification de la certification SSL demandée par CURLOPT_SSL_VERIFYPEER
- CURLINFO_CONTENT_LENGTH_DOWNLOAD − Taille du corps du téléchargement, lu dans l'en−tête Content−Length:
- CURLINFO_CONTENT_LENGTH_UPLOAD − Taille spécifiée de l'envoi.
- CURLINFO_CONTENT_TYPE − Content−Type: du fichier téléchargé. NULL indique que le serveur n'a pas envoyé d'en−tête Content−Type:

Lorsqu'elle est appelée sans le paramètre opt , un tableau associatif est retourné avec les éléments suivants :

- url
- content_type
- http_encode
- header_size
- request_size
- filetime
- ssl_verify_result
- redirect_count
- total_time
- namelookup_time
- connect_time
- pretransfer_time
- size_upload
- size_download
- speed_download
- speed_upload
- download_content_length
- upload_content_length
- starttransfer_time
- redirect_time

### 6.5.7 curl_errno() : Retourne un entier représentant le dernier message d'erreur CURL

## 6.5.11 curl_init() : Initialise une session CURL

resource **curl_init** ( *string* *url* )

curl_init initialise une nouvelle session et retourne un identifiant de session CURL à utiliser avec les fonctions curl_setopt , curl_exec et curl_close . Si le paramètre optionnel url est fourni, alors CURLOPT_URL prendra cette valeur. Vous pouvez manuellement fixer cette valeur avec la fonction curl_setopt .

| Initialiser une session CURL et récupérer une page web |
|---|

```
 <?php
$ch = curl_init();

curl_setopt ($ch, CURLOPT_URL, "http://www.exemple.com/");
curl_setopt ($ch, CURLOPT_HEADER, 0);

curl_exec ($ch);

curl_close ($ch);
?>
```

Voir aussi curl_close et curl_setopt .

## 6.5.12 curl_multi_add_handle() : Ajoute une ressource cURL à un cURL multiple

int **curl_multi_add_handle** ( resource mh , resource ch )

| Attention |
|---|
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_multi_init , curl_init , et curl_multi_remove_handle .

## 6.5.13 curl_multi_close() : Termine un jeu de transferts cURL

void **curl_multi_close** ( resource mh )

| Attention |
|---|
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_multi_init et curl_close .

## 6.5.14 curl_multi_exec() : Exécute les sous−requêtes d'un cURL

int **curl_multi_exec** ( resource mh )

| Attention |
|---|
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_multi_init et curl_exec .

## 6.5.15 curl_multi_getcontent() : Retourne le contenu obtenu avec l'option CURLOPT_RETURNTRANSFER

string **curl_multi_getcontent** ( resource ch )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_multi_init .

## 6.5.16 curl_multi_info_read() : Lit les informations sur les transferts actuels

array **curl_multi_info_read** ( resource mh )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_multi_init .

## 6.5.17 curl_multi_init() : Retourne un nouveau cURL multiple

resource **curl_multi_init** ( )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_init et curl_multi_close .

## 6.5.18 curl_multi_remove_handle() : Retire un cURL multiple d'un jeu de cURL

int **curl_multi_remove_handle** ( resource mh , resource ch )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_multi_init , curl_init et curl_multi_add_handle .

## 6.5.19 curl_multi_select() : Retourne les sockets associées à cURL, pour utilisation

int **curl_multi_select** ( resource mh , *float* *timeout* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi curl_multi_init .

## 6.5.20 curl_setopt() : Modifie une option de transfert CURL

bool **curl_setopt** ( resource ch , string option , mixed value )

curl_setopt fixe les options de transfert de la session CURL identifiée par ch . option est le nom de l'option à fixer, et value est sa valeur.

value doit être de type "long" pour les options suivantes (spécifiée par option ) :

- curlopt_infilesize : Lorsque vous téléchargez un fichier sur un site distant, cette option sert à indiquer à PHP la taille maximale du fichier attendu.
- curlopt_verbose : Choisissez une valeur non nulle pour que CURL vous affiche tous les événements.
- curlopt_header : Choisissez une valeur non nulle pour que CURL inclut l'en−tête dans la valeur de retour.
- curlopt_noprogress : Choisissez une valeur non nulle pour que PHP n'affiche pas l'état des transferts CURL.

  | Note |
  | --- |
  | PHP choisit automatiquement une valeur non nulle. Ne changez cette valeur que le temps du débogage. |

- curlopt_nobody : Choisissez une valeur non nulle pour que le corps du transfert ne soit pas inclus dans la valeur de retour.
- curlopt_failonerror : Choisissez une valeur non nulle pour que PHP traite silencieusement les codes HTTP supérieurs à 300. Le comportement par défaut est de retourner la page normalement, en ignorant ce code.
- curlopt_upload : Choisissez une valeur non nulle pour que PHP prépare un chargement.
- curlopt_post : Choisissez une valeur non nulle pour que PHP fasse un HTTP POST. Un POST est un encodage normal "application/x−www−from−url", utilisé couramment par les formulaires HTML.
- curlopt_ftplistonly : Choisissez une valeur non nulle pour que PHP ne fasse que lister les noms d'un dossier FTP.
- curlopt_ftpappend : Choisissez une valeur non nulle pour que PHP concatène le fichier distant, plutôt que de l'écraser.
- curlopt_netrc : Choisissez une valeur non nulle pour que PHP analyse votre fichier ~./netrc et utilise votre nom de compte et mot de passe sur le site distant que vous souhaitez contacter.
- curlopt_followlocation : Choisissez une valeur non nulle pour suivre toutes les en−têtes "Location: " que le serveur envoie dans les en−têtes HTTP (notez que cette fonction est récursive et que PHP suivra toutes les en−têtes "Location: " qu'il trouvera).
- curlopt_put : Choisissez une valeur non nulle pour que pour chargement se fasse par HTTP PUT. Le fichier à charger doit être fixé avec les options CURLOPT_INFILE et CURLOPT_INFILESIZE .
- curlopt_mute : Choisissez une valeur non nulle pour que PHP soit totalement silencieux concernant toutes les fonctions CURL.
- curlopt_timeout : Passez un entier "long" comme paramètre qui représente le temps maximum d'exécution de la fonction CURL.
- curlopt_low_speed_limit : Passez un entier long qui représente la vitesse minimale en octets par secondes en dessous de laquelle, et pendant CURLOPT_LOW_SPEED secondes, PHP considérera qu'elle est trop lente, et annulera le transfert.
- curlopt_low_speed_time : Passez un entier "long" qui représente le temps en secondes, qui, si la vitesse de transfert reste en dessous de CURLOPT_LOW_SPEED_LIMIT , PHP considèrera que la connexion est trop lente, et l'annulera.
- curlopt_resume_from : Passez un entier "long", qui représente l'offset, en octets, à partir duquel vous voulez commencer le transfert.

- curlopt_sslversion : Passez un entier "long" qui contient la version de SSL (2 ou 3) à utiliser. Par défaut, PHP essaiera de le déterminer lui−même, bien que dans certains cas, il vous faudra le faire manuellement.
- curlopt_timecondition : Passez un entier "long" qui définit comment CURLOPT_TIMEVALUE est utilisé. Vous pouvez choisir entre les valeurs TIMECOND_IFMODSINCE ou TIMECOND_ISUNMODSINCE . C'est une fonctionnalité HTTP.
- curlopt_timevalue : Passez un entier "long" qui représente le temps en secondes depuis le 1er janvier 1970. Cette valeur sera utilisée comme spécifié dans l'option CURLOPT_TIMEVALUE . Par défaut, TIMECOND_IFMODSINCE sera utilisée.

value doit être une chaîne de caractères pour les valeurs suivantes de option

- curlopt_url : L'URL que PHP va récupérer. Vous pouvez aussi choisir cette valeur lors de l'appel à curl_init .
- curlopt_userpwd : Passez une chaîne de caractères au format [nom]:[mot de passe], pour que PHP l'utilise lors de la connexion.
- curlopt_proxyuserpwd : Passez une chaîne de caractères au format [nom]:[mot de passe ], pour que PHP l'utilise lors de la connexion à un proxy HTTP.
- curlopt_range : Passez une chaîne de caractères qui représente la plage de valeurs que vous désirez. Elle est au format "X−Y", où les valeurs de X ou Y peuvent être omises. Le transfert HTTP supporte aussi plusieurs intervalles, séparés par des virgules : X−Y,N−M.
- curlopt_postfields : Passez une chaîne de caractères qui contient toutes les données à passer lors d'une opération de HTTP POST.
- curlopt_referer : Passez une chaîne de caractères qui contient l'en−tête de "REFERER", utilisé lors d'une requête HTTP.
- curlopt_useragent : Passez une chaîne de caractères qui contient l'en−tête "user−agent" utilisé dans une requête HTTP.
- curlopt_ftpport : Passez une chaîne de caractères qui désignera l'adresse IP utilisée pour l'instruction FTP "PORT". L'instruction POST indique au serveur distant de se connecter à cette adresse IP. La chaîne peut être une adresse IP, un nom d'hôte, un nom d'interface réseau (sous UNIX), ou juste '−', pour utiliser les IP par défaut du système.
- curlopt_cookie : Passez une chaîne de caractères qui contiendra le contenu du cookie, à transmettre dans l'en−tête HTTP.
- curlopt_sslcert : Passez une chaîne de caractères qui contiendra le nom de fichier du certificat, au format PEM.
- curlopt_sslcertpasswd : Passez une chaîne de caractères qui contient le mot de passe nécessaire pour utiliser le certificat CURLOPT_SSLCERT .
- curlopt_cookiefile : Passez une chaîne de caractères qui contiendra le nom du fichier contenant les données de cookie. Le fichier de cookie peut être au format Netscape, ou simplement des en−têtes HTTP écrits dans un fichier.
- curlopt_customrequest : Passez une chaîne de caractères qui sera utilisée à la place de GET ou HEAD lors des requêtes HTTP. Cette commande est pratique pour effectuer un DELETE ou une autre commande HTTP exotique.

| Note |
| --- |
| N'utilisez pas cette commande sans vous assurer que le serveur l'accepte. |

Les options suivantes requièrent un pointeur de fichier, qui est obtenu avec la fonction fopen :

- curlopt_file : Le fichier de sortie de votre transfert. Par défaut, STDOUT.
- curlopt_infile : Le fichier d'entrée de votre transfert.
- curlopt_writeheader : Le fichier de destination de l'en−tête de la sortie du transfert.
- curlopt_stderr : Le fichier d'erreurs.

## 6.5.21 curl_version() : Retourne la version courante de CURL

string **curl_version** ( void )

curl_version retourne une chaîne avec la version courante de la librairie CURL.

# 6.6 Dates et heures

## 6.6.1 Introduction

Ces fonctions vous permettent de manipuler la date et l'heure sur le serveur qui exécute PHP. Vous pouvez utiliser ces fonctions pour formater la date et l'heure de nombreuses façons.

| Note |
| --- |
| N'oubliez pas que ces fonctions dépendent de la configuration locale de votre serveur. Faites notamment attention aux changements d'heure d'hiver et d'été, ainsi qu'aux années bissextiles. |

## 6.6.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.6.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.6.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.6.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.6.6 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- checkdate
- date
- getdate
- gettimeofday
- gmdate
- gmmktime
- gmstrftime
- localtime
- microtime

- mktime
- strftime
- strtotime
- time

## 6.6.8 date() : Formate une date/heure locale

string **date** ( string format , *int timestamp* )

date retourne une date sous forme d'une chaîne, au format donné par la chaîne format . La date est fournie par le paramètre timestamp , sous la forme d'un timestamp. Par défaut, la date courante est utilisée.

| Note |
|---|
| L'intervalle de validité d'un timestamp va généralement du Vendredi 13 Décembre 1901 20:45:54 GMT au Mardi 19 Janvier 2038 03:14:07 GMT. (Ces dates correspondent aux valeurs minimales et maximales des entiers 32 bits non−signés).<br><br>Pour générer un timestamp à partir d'une représentation de date, pour pouvez utiliser la fonction strtotime . De plus, certaines bases de données disposent de fonctions pour convertir leurs propres formats de date en timestamps (comme par exemple, MySQL et sa fonction UNIX_TIMESTAMP() ). |

Les caractères suivants sont utilisés pour spécifier le format :

| Caractère | Description | Exemple de valeur |
|---|---|---|
| a | Ante meridiem et Post meridiem en minuscules | am ou pm |
| A | Ante meridiem et Post meridiem en majuscules | AM ou PM |
| B | Heure Internet Swatch | 000 à 999 |
| d | Jour du mois, sur deux chiffres (avec un zéro initial) | 01 à 31 |
| D | Jour de la semaine, en trois lettres (et en anglais) | Mon à Sun |
| F | Mois, textuel, version longue; en anglais, comme January ou December | January à December |
| g | Heure, au format 12h, sans les zéros initiaux | 1 à 12 |
| G | Heure, au format 24h, sans les zéros initiaux | 0 à 23 |
| h | Heure, au format 12h, avec les zéros initiaux | 01 à 12 |
| H | Heure, au format 24h, avec les zéros initiaux | 00 à 23 |
| i | Minutes avec les zéros initiaux | 00 à 59 |
| I (i majuscule) | L'heure d'été est activée ou as | 1 si oui, 0 sinon. |
| j | Jour du mois sans les zéros initiaux | 1 à 31 |
| l (lowercase 'L') | Jour de la semaine, textuel, version longue, en anglais | Sunday à Saturday |
| L | Est ce que l'année est bissextile | 1 si bissextile, 0 sinon. |
| m | Mois au format numérique, avec zéros initiaux | 01 à 12 |
| M | Mois, en trois lettres, en anglais | Jan à Dec |
| n | Mois sans les zéros initiaux | 1 à 12 |
| O | Différence d'heures avec l'heure de Greenwich (GMT), exprimée en heures | Exemple : +0200 |
| r | Format de date RFC 822 | Exemple : Thu, 21 Dec 2000 16:01:07 +0200 |
| s | Secondes, avec zéros initiaux | 00 à 59 |
| S | Suffixe ordinal d'un nombre pour le jour du mois, en anglais, sur deux lettres | st , nd , rd ou th . Fonctionne bien avec j |
| t | Nombre de jours dans le mois | 28 à 31 |
| T | Fuseau horaire du serveur | Exemples : EST , MDT ... |
| U | Secondes depuis l'époque Unix (1er Janvier 1970, 0h00 00s GMT) | Voir aussi time |
| w | Jour de la semaine au format numérique | 0 (pour dimanche) à 6 (pour samedi) |
| W | Numéro de semaine dans l'année ISO−8601, les semaines commencent le lundi (ajouté en PHP 4.1.0) | Exemple : 42 (la 42ème semaine de l'année) |
| Y | Année, 4 chiffres | Exemples : 1999 et 2003 |
| y | Année, 2 chiffres | Exemples : 99 et 03 |
| z | Jour de l'année | 0 Ë 366 |
| Z | Décalage horaire en secondes. Le décalage des zones à l'ouest de la zone UTC est négative, et à l'est, il est positif. | −43200 à 43200 |

Les caractères non reconnus seront imprimés tels quel. "Z" retournera toujours "0" lorsqu'il est utilisé avec gmdate .

Exemple avec date

```
<?php
 echo (date("l dS of F Y h:i:s A"));
 echo ("Le 14 Juillet 2002 tombe un " . date("l", mktime(0,0,0,7,14,2002)));
?>
```

Vous pouvez faire afficher un caractère spécial dans la chaîne de format en le protégeant par un anti−slash. Si le caractère est lui−même une séquence incluant un anti−slash, vous devrez protéger aussi l'anti−slash.

Protection des caractères avec date

```
<?php
echo date("l \\t\h\e jS");
 // affiche 'Saturday the 8th'
// \t représente une tabulation
?>
```

Il est possible d'utiliser date et mktime ensemble pour générer des dates dans le futur ou dans le passé.

Exemples avec date et mktime

```
<?php
 $demain          = mktime(0,0,0,date("m")  ,date("d") + 1,date("Y")    );
 $le_mois_dernier = mktime(0,0,0,date("m")−1,date("d")    ,date("Y")    );
 $l_an_prochain   = mktime(0,0,0,date("m"),  date("d")    ,date("Y") + 1);
?>
```

Note

Cette méthode est plus sûre que simplement ajouter ou retrancher le nombre de secondes dans une journée ou un mois à un timestamp, à cause des heures d'hiver et d'été.

Voici maintenant quelques exemples de formatage avec date . Notez que vous devriez échapper tous les autres caractères, car s'ils ont une signification spéciale, ils risquent de produire des effets secondaires indésirables. Notez aussi que les versions futures de PHP peuvent attribuer une signification à des lettres qui sont actuellement inertes. Lorsque vous échappez les caractères, pensez à utiiser des guillemets simples, pour que les séquences \n ne deviennent pas des nouvelles lignes.

Formatage avec date

```
<?php
/* Aujourd'hui, le 12 Mars 2001, 10:16:18 pm */
  $aujourdhui = date("F j, Y, g:i a");                 // March 12, 2001, 10:16 pm
  $aujourdhui = date("m.d.y");                         // 03.12.01
  $aujourdhui = date("j, m, Y");                       // 12, 3, 2001
  $aujourdhui = date("Ymd");                           // 20010312
  $aujourdhui = date('h−i−s, j−m−y, it is w Day z ');  // 05−16−17, 12−03−01, 1631 1618 6 Monpm01
  $aujourdhui = date('\C\'\e\s\t\ \l\e\ jS \j\o\u\r\.');   // C'est le 12th jour.
  $aujourdhui = date("D M j G:i:s T Y");               // Mon Mar 12 15:16:08 MST 2001
  $aujourdhui = date('H:m:s \m \e\s\t\ \l\e\ \m\o\i\s');     // 17:03:18 m est le mois
  $aujourdhui = date("H:i:s");                         // 10:16:18
// notation française
$aujourdhui = date("d/m/y");                         // 12/03/01
$aujourdhui = date("d/m/Y");                         // 12/03/2001
?>
```

Pour formater des dates dans d'autres langues, utilisez les fonctions setlocale et strftime .

Voir aussi getlastmod , time , strftime , gmdate et mktime .

## 6.6.9 getdate() : Retourne la date/heure

array **getdate** ( *int timestamp* )

getdate retourne un tableau associatif contenant les informations de date et d'heure du timestamp timestamp (lorsqu'il est fourni), avec les champs suivants :

- "seconds" − secondes
- "minutes" − minutes
- "hours" − heures
- "mday" − jour du mois
- "wday" − jour de la semaine, numérique. 0: dimanche jusqu'à 6: samedi
- "mon" − mois, numérique
- "year" − année, numérique
- "yday" − jour de l'année, numérique; i.e. "299"
- "weekday" − jour de la semaine, texte complet (en anglais); i.e. "Friday"
- "month" − mois, texte complet (en anglais); i.e. "January"

Exemple avec getdate

```
<?php
 $aujourdhui = getdate();
 $mois = $aujourdhui['month'];
 $mjour = $aujourdhui['mday'];
 $annee = $aujourdhui['year'];
 echo "$mjour/$mois/$annee";
?>
```

Voir aussi date , time et setlocale .

## 6.6.10 gettimeofday() : Retourne l'heure actuelle

array **gettimeofday** ( void )

gettimeofday est une interface vers gettimeofday(2). Elle retourne un tableau associatif qui contient les informations retournées par le système :

- "sec" − secondes
- "usec" − microsecondes
- "minuteswest" − minutes de décalage par rapport à Greenwich, vers l'Ouest.
- "dsttime" − type de correction dst

## 6.6.11 gmdate() : Formate une date/heure GMT/CUT

string **gmdate** ( string format , *int timestamp* )

gmdate est identique à la fonction date , hormis le fait que le temps retourné est GMT (Greenwich Mean Time). Par exemple, en Finlande (GMT +0200), la première ligne ci−dessous affiche "Jan 01 1998 00:00:00", tandis que la seconde "Dec 31 1997 22:00:00".

Exemple avec gmdate

```
<?php
 echo date("M d Y H:i:s", mktime(0,0,0,1,1,1998));
 echo gmdate("M d Y H:i:s", mktime(0,0,0,1,1,1998));
?>
```

Note

Sous Microsoft Windows, les librairies qui implémentent cette fonction sont buggées, et gmdate ne

supporte pas les valeurs négatives pour le paramètre timestamp . Pour plus de détails, voyez ces rapports de bugs : #22620 , #22457 , et #14391 .

Ce problème n'est pas présent sur les systèmes d'exploitation Unix et Linux, où les librairies se comportent comme prévu.

PHP ne peut pas réparer ces librairies systèmes. Contactez l'auteur de votre système d'exploitation pour corriger ce problème.

Voir aussi date , strftime , mktime et gmmktime .

## 6.6.12 gmmktime() : Retourne le timestamp UNIX d'une date GMT

int **gmmktime** ( *int hour* , *int minute* , *int second* , *int month* , *int day* , *int year* , *int is_dst* )

**gmmktime** est identique à mktime hormis le fait que les paramètres passés sont GMT.

Comme mktime , les arguments restants peuvent être ignorés. Ils prendront alors leur valeur GMT actuelle.

## 6.6.13 gmstrftime() : Formate une date/heure GMT/CUT en fonction des paramétrages locaux.

string **gmstrftime** ( string format , *int timestamp* )

gmstrftime se comporte exactement comme strftime hormis le fait que l'heure utilisée est celle de Greenwich (Greenwich Mean Time (GMT)). Par exemple, dans la zone Eastern Standard Time (est des USA) (GMT −0500), la première ligne de l'exemple ci−dessous affiche "Dec 31 1998 20:00:00", tandis que la seconde affiche "Jan 01 1999 01:00:00".

Exemple avec gmstrftime

```
<?php
 setlocale(LC_TIME, 'en_US');
 echo strftime("%b %d %Y %H:%M:%S", mktime (20,0,0,12,31,98))."\n";
 echo gmstrftime("%b %d %Y %H:%M:%S", mktime (20,0,0,12,31,98))."\n";
?>
```

Voir aussi strftime .

## 6.6.14 localtime() : Lit l'heure locale

array **localtime** ( *int timestamp* , *bool is_associative* )

localtime retourne un tableau identique à la structure retournée par la fonction C localtime. Le premier argument timestamp est un timestamp UNIX. S'il n'est pas fourni, l'heure courante est utilisée. Le second argument is_associative , s'il est mis à 0 ou ignoré, force localtime à retourner un tableau à index numérique. S'il est mis à 1, localtime retourne un tableau associatif, avec tous les éléments de la structure C, accessible avec les clés suivantes :

- "tm_sec" − secondes
- "tm_min" − minutes
- "tm_hour" − heure

- "tm_mday" − jour du mois
- "tm_mon" − mois de l'année
- "tm_year" − Année, incompatible an 2000
- "tm_wday" − Jour de la semaine
- "tm_yday" − Jour de l'année
- "tm_isdst" − Est−ce que l'heure d'hiver a pris effet?

## 6.6.15 microtime() : Retourne le timestamp UNIX actuel avec microsecondes

string **microtime** ( void )

microtime retourne la chaîne "msec sec" avec sec qui est mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT), et msec qui est le nombre de microsecondes de cette heure. Cette fonction est seulement disponible sur les systèmes d'exploitation qui supportent la fonction système gettimeofday().

Les deux parties de la chaîne sont exprimées en secondes.

Exemple avec microtime

```
 <?php
function getmicrotime(){
     list($usec, $sec) = explode(" ",microtime());
     return ((float)$usec + (float)$sec);
     }

 $time_start = getmicrotime();

for ($i=0; $i < 1000; $i++){
    //do nothing, 1000 times
    }

$time_end = getmicrotime();
$time = $time_end − $time_start;

echo "Ne fait rien pendant $time secondes";
?>
```

Voir aussi time .

## 6.6.16 mktime() : Retourne le timestamp UNIX d'une date

int **mktime** ( *int hour* , *int minute* , *int second* , *int month* , *int day* , *int year* , *int is_dst* )

**ATTENTION** : l'ordre des arguments est différent de celui de la commande UNIX habituelle mktime(), et fournit des résultats aléatoires si on oublie cet ordre. C'est une erreur très commune que de se tromper de sens.

mktime retourne un timestamp UNIX correspondant aux arguments fournis. Ce timestamp est un entier long, contenant le nombre de secondes entre le début de l'époque UNIX (1er Janvier 1970) et le temps spécifié.

Les arguments peuvent être omis, de droite à gauche, et tous les arguments manquants sont utilisés avec la valeur courante de l'heure et du jour.

is_dst peut être mis à 1 si l'heure d'hiver est appliquée, 0 si elle ne l'est pas, et −1 (par défaut) si on ne sait pas.

| Note |
| --- |
| is_dst a été ajouté à partir de la version 3.0.10. |

mktime est pratique pour faire des calculs de dates et des validations, car elle va automatiquement corriger les valeurs invalides. Par exemple, toutes les lignes suivantes vont retourner la même date : "Jan−01−1998".

Exemple mktime

```
<?php
 echo date("M-d-Y", mktime (0,0,0,12,32,1997));
 echo date("M-d-Y", mktime (0,0,0,13,1,1997));
 echo date("M-d-Y", mktime (0,0,0,1,1,1998));
 echo date("M-d-Y", mktime (0,0,0,1,1,98));
?>
```

year peut prendre deux ou quatre chiffres, avec les valeurs entre 0−69 qui correspondent à 2000−2069 et 70−99 à 1970−1999 (sur les systèmes où time_t sont sur des entiers 32bit signés, comme cela se fait le plus souvent de nos jours, year est valide dans l'intervalle 1902 et 2037.

Le dernier jour d'un mois peut être décrit comme le jour "0" du mois suivant, et non pas le jour −1. Les deux exemples suivants vont donner : "Le dernier jour de Fevrier 2000 est: 29".

Dernier jour du mois

```
<?php
 $lastday = mktime (0,0,0,3,0,2000);
 echo strftime ("Le dernier jour de Fevrier 2000 est: %d", $lastday);
 $lastday = mktime (0,0,0,4,-31,2000);
 echo strftime ("Le dernier jour de Fevrier 2000 est: %d", $lastday);
?>
```

Voir aussi date et time .

## 6.6.17 strftime() : Formate une date/heure locale avec les options locales.

string **strftime** ( string format , *int timestamp* )

strftime retourne la date sous la forme d'une chaîne formatée conformément au format format , en utilisant le timestamp timestamp donné. Si le timestamp est omis, la date actuelle est utilisée. Les mois et jours de la semaine, et toutes les chaînes dépendantes de la langue sont fixées avec la commande setlocale .

Les caractères suivants sont utilisés pour spécifier le format de la date :

- %a − nom abrégé du jour de la semaine (local).
- %A − nom complet du jour de la semaine (local).
- %b − nom abrégé du mois (local).
- %B − nom complet du mois (local).
- %c − représentation préférée pour les dates et heures, en local.
- %C − numéro de siècle (l'année, divisée par 100 et arrondie entre 00 et 99)
- %d − jour du mois en numérique (intervalle 01 à 31)
- %D − identique à %m/%d/%y
- %e − numéro du jour du mois. Les chiffres sont précédés d'un espace (de ' 1' à '31')
- %h − identique à %b
- %H − heure de la journée en numérique, et sur 24−heures (intervalle de 00 à 23)
- %I − heure de la journée en numérique, et sur 12− heures (intervalle 01 à 12)
- %j − jour de l'année, en numérique (intervalle 001 à 366)
- %m − mois en numérique (intervalle 1 à 12)

- %M − minute en numérique
- %n − newline character
- %p − soit `am' ou `pm' en fonction de l'heure absolue, ou en fonction des valeurs enregistrées en local.
- %r − l'heure au format a.m. et p.m.
- %R − l'heure au format 24h
- %S − secondes en numérique
- %t − tabulation
- %T − l'heure actuelle (égal à %H:%M:%S)
- %u − le numéro de jour dans la semaine, de 1 à 7. (1 représente Lundi)
- %U − numéro de semaine dans l'année, en considérant le premier dimanche de l'année comme le premier jour de la première semaine.
- %V − le numéro de semaine comme défini dans l'ISO 8601:1988, sous forme décimale, de 01 à 53. La semaine 1 est la première semaine qui a plus de 4 jours dans l'année courante, et dont Lundi est le premier jour.
- %W − numéro de semaine dans l'année, en considérant le premier lundi de l'année comme le premier jour de la première semaine
- %w − jour de la semaine, numérique, avec Dimanche = 0
- %x − format préféré de représentation de la date sans l'heure
- %X − format préféré de représentation de l'heure sans la date
- %y − l'année, numérique, sur deux chiffres (de 00 à 99)
- %Y − l'année, numérique, sur quatre chiffres
- %Z − fuseau horaire, ou nom ou abréviation
- %% − un caractère `%' littéral

Note

Tous les caractères suivants ne sont pas toujours supportés par toutes les librairies C. Dans ce cas, ils ne seront pas supportés par PHP non plus. De plus, toutes les plates−formes ne supportent pas les timestamps négatifs, et vos dates pourraient être limitées par le début de l'époque Unix. Cela signifie que %e, %T, %R et %D (et peut être d'autres) et les dates antérieures au 1er Janvier 1970 ne fonctionneront pas sous Windows, sur certaines distributions de Linux, et sur certains systèmes d'exploitation. Pour Windows, une liste complète des options de conversion est disponible sur le site de MSDN .

Exemple avec strftime

```
<?php
 setlocale(LC_TIME, "C");
 echo strftime("%A en Finlandais est ");
 setlocale(LC_TIME, "fi");
 echo strftime("%A, en Français ");
 setlocale(LC_TIME, "fr");
 echo strftime("%A et en Allemand ");
 setlocale(LC_TIME, "de");
 echo strftime("%A.\n");
?>
```

Cet exemple ne fonctionnera que si vous avez les locales respectives installées sur votre système.

Note

%G et %V, qui sont basées sur la semaine ISO 8601:1988, peut conduire à des résultat inattendus (bien que corrects) si le système de numérotation n'est pas connu. Voyez l'exemple %V ci−dessus et l'exemple ci−dessous.

Exemple au format de date ISO 8601:1988

```
<?php
/*     December 2002 / January 2003
ISOWk  M   Tu  W   Thu F   Sa  Su
----- ----------------------------
51     16  17  18  19  20  21  22
 52     23  24  25  26  27  28  29
1      30  31   1   2   3   4   5
2       6   7   8   9  10  11  12
3      13  14  15  16  17  18  19   */

// Affiche : 12/28/2002 − %V,%G,%Y = 52,2002,2002
print "12/28/2002 − %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/28/2002")) . "\n";
```

```
// Affiche : 12/30/2002 - %V,%G,%Y = 1,2003,2002
print "12/30/2002 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/30/2002")) . "\n";

// Affiche : 1/3/2003 - %V,%G,%Y = 1,2003,2003
print "1/3/2003 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/3/2003")) . "\n";

// Affiche : 1/10/2003 - %V,%G,%Y = 2,2003,2003
print "1/10/2003 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/10/2003")) . "\n";



/*     December 2004 / January 2005
ISOWk  M   Tu  W   Thu F   Sa  Su
----- ---------------------------
51     13  14  15  16  17  18  19
52     20  21  22  23  24  25  26
53     27  28  29  30  31   1   2
1       3   4   5   6   7   8   9
2      10  11  12  13  14  15  16   */

// Affiche : 12/23/2004 - %V,%G,%Y = 52,2004,2004
print "12/23/2004 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/23/2004")) . "\n";

// Affiche : 12/31/2004 - %V,%G,%Y = 53,2004,2004
print "12/31/2004 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("12/31/2004")) . "\n";

// Affiche : 1/2/2005 - %V,%G,%Y = 53,2004,2005
print "1/2/2005 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/2/2005")) . "\n";

// Affiche : 1/3/2005 - %V,%G,%Y = 1,2005,2005
print "1/3/2005 - %V,%G,%Y = " . strftime("%V,%G,%Y",strtotime("1/3/2005")) . "\n";

?>
```

Voir aussi setlocale , mktime et le groupe de spécifications de strftime() .

## 6.6.18 strtotime() : Transforme un texte anglais en timestamp

int **strtotime** ( string time , *int* *now* )

strtotime essaye de lire une date au format anglais dans la chaîne time , et de la transformer en timestamp UNIX, relativement au timestamp now , ou à la date courante si ce dernier est omis. En cas d'erreur, −1 est retourné.

Comme strtotime suit la syntaxe de date GNU, vous pouvez lire le manuel GNU intitulé Date Input Formats . Ci−dessous, vous trouverez une syntaxe valide pour le paramètre time .

Exemple avec strtotime

```
<?php
// l'exemple n'est pas traduit, car cela ne fonctionne qu'en anglais
echo strtotime ("now"), "\n";
echo strtotime ("10 September 2000"), "\n";
echo strtotime ("+1 day"), "\n";
echo strtotime ("+1 week"), "\n";
echo strtotime ("+1 week 2 days 4 hours 2 seconds"), "\n";
echo strtotime ("next Thursday"), "\n";
echo strtotime ("last Monday"), "\n";
?>
```

Vérification des erreurs avec strtotime

```
$str = 'Pas une bonne date';
if (($timestamp = strtotime($str)) === -1) {
    echo "La chaîne ($str) est bogus";
} else {
    echo "$str == ". date('l dS of F Y h:i:s A',$timestamp);
}
```

Note

L'intervalle de validité d'un timestamp va du Vendredi 13 Décembre 1901 20:45:54 GMT au Mardi 19 Jan 2038 03:14:07 GMT : cela correspond aux dates maximales et minimales pour un entier de 32 bits signé.

## 6.6.19 time() : Retourne le timestamp UNIX actuel.

int **time** ( void )

time retourne l'heure courante, mesurée en secondes depuis le début de l'époque UNIX, (1er janvier 1970 00:00:00 GMT).

Voir aussi date .

# 6.7 dBase

## 6.7.1 Introduction

Ces fonctions vous permettront d'accéder aux enregistrements d'une base au format dBase ( .dbf ).

dBase ne permet pas l'utilisation d'index, de "memo fields" , ni le blocage de la base. Deux processus de serveurs web différents modifiant le même fichier dBase risquent de rendre votre base de données incohérente.

Les fichiers dBase sont de simples fichiers séquentiels d'enregistrements de longueur fixe. Les enregistrements sont ajoutés à la fin du fichier et les enregistrements supprimés sont conservés jusqu'à l'appel de dbase_pack .

Nous vous recommandons de ne pas utiliser les fichiers dBase comme base de données de production. Choisissez n'importe quel serveur SQL à la place. MySQL et PostgreSQL sont des choix classiques avec PHP . Le support de dBase ne se justifie ici que pour vous permettre d'importer et d'exporter des données depuis et vers votre base de données issues du web, car ce format de fichier est communément accepté par les feuilles et assistants personnes type Windows. L'import et l'export de données sont les deux uniques fonctionnalités qui parlent en faveur de l'utilisation de dBase.

## 6.7.2 Installation

Pour activer la librairie dBase fournie avec PHP , vous devez compiler PHP avec l'option −−enable−dbase .

## 6.7.3 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.7.4 Types de ressources

Cette extension ne définit aucune ressource.

## 6.7.5 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- dbase_add_record
- dbase_close
- dbase_create
- dbase_delete_record

- dbase_get_header_info
- dbase_get_record_with_names
- dbase_get_record
- dbase_numfields
- dbase_numrecords
- dbase_open
- dbase_pack
- dbase_replace_record

## 6.7.7 dbase_close() : Ferme une base dBase

bool **dbase_close** ( int dbase_identifier )

dbase_close ferme la base associée à dbase_identifier .

## 6.7.8 dbase_create() : Crée une base de données dBase

int **dbase_create** ( string filename , array fields )

dbase_create crée une base de données dBase dans le fichier filename , et avec les champs fields .

fields est un tableau de tableaux. Chaque tableau décrit le format d'un fichier de la base. Chaque champ est constitué d'un nom, d'un caractère de type de champ, d'une longueur et d'une précision.

Les types de champ disponibles sont :

*L*
: Boolean (booléen). Pas de longueur ou de précision pour ces valeurs.

*M*
: Memo . (Note importante : les mémos ne sont pas supportés par PHP .) Elles n'ont pas de longueur ou de précision.

*D*
: Date (enregistrée au format 'YYYYMMDD' ). Elles n'ont pas de longueur ou de précision.

*N*
: Number (nombre). Possède une longueur et une précision (le nombre de chiffres après la virgule).

*C*
: String (chaîne de caractère).

Si la base de données a été créée, un identifiant de base dbase_identifier est retourné, sinon, FALSE est retourné.

Création d'une base dBase

```
<?php
// "database" name
  $dbname = "/tmp/test.dbf";
// database "definition"
  $def =
    array(
        array("date",     "D"),
        array("name",     "C",  50),
        array("age",      "N",   3, 0),
        array("email",    "C", 128),
        array("ismember", "L")
    );
```

```
// création
  if (!dbase_create($dbname, $def))
    print "<strong>Erreur!</strong>";
?>
```

## 6.7.9 dbase_delete_record() : Efface un enregistrement dans une base dBase

bool **dbase_delete_record** ( int dbase_identifier , int record )

dbase_delete_record marque l'enregistrement record pour l'effacement, dans la base dbase_identifier . Pour effacer réellement l'enregistrement, il faut utiliser aussi dbase_pack .

## 6.7.10 dbase_get_header_info() : Affichage des informations d'en−tête d'une base de données dBase.

array **dbase_get_header_info** ( int dbase_identifier )

dbase_get_header_info retourne des informations sur la structure des colonnes de la base de données référencée par dbase_identifier . Pour chaque colonne de la base de données, il y'a une entrée dans un tableau indexé numériquement. L'index du tableau commence à 0. Chaque entrée du tableau contient un tableau associatif contenant les informations des colonnes. Si les en−têtes d'informations de la base de données ne peuvent pas être lus, cette fonction retourne FALSE .

Les éléments du tableau sont :

*name*
: Le nom de la colonne

*type*
: Le type compréhensible de la colonne dBase (i.e. date, boolean, etc..)

*length*
: Le nombre d'octets que cette colonne peut contenir

*precision*
: Le nombre de décimales de précision pour la colonne

*format*
: Une suggestion de format pour printf

*offset*
: L'octet d'offset de la colonne à partir du début de la ligne

Affichage des informations d'en−tête d'un fichier de base de données dBase

```
 <?php
// Chemin vers un fichier dBase
$db_path = "/tmp/test.dbf";

// Ouvre le fichier
$dbh = dbase_open($db_path)
    or die("Erreur! Impossible d'ouvrir le fichier '$db_path'.");

// Obtient les informations des colonnes
$column_info = dbase_get_header_info($dbh);

// Affiche les informations
print_r($column_info);
?>
```

## 6.7.11 dbase_get_record_with_names() : Lit un enregistrement dans une base dBase, sous la forme d'un tableau associatif

array **dbase_get_record_with_names** ( int dbase_identifier , int record )

dbase_identifier retourne les données de l'enregistrement record dans un tableau associatif. Le tableau inclut un membre nommé "deleted" (effacé), qui sera mis à 1 si l'enregistrement a été marqué pour l'effacement (voir dbase_delete_record ).

Chaque champ est converti au format approprié PHP, hormis :

- Les dates, qui sont laissées au format chaîne
- Les entiers qui causeraient des dépassement de capacité (> 32 bits) sont retournés comme des chaînes.

## 6.7.12 dbase_get_record() : Lit un enregistrement dans une base dBase

array **dbase_get_record** ( int dbase_identifier , int record )

dbase_get_record retourne les données de l'enregistrement record dans un tableau. Les données sont lues dans la base dbase_identifier . Le tableau est indexé à partir de 0, et inclus un membre nommé "deleted" (effacé), qui sera mis à 1 si l'enregistrement a été marqué pour l'effacement (voir dbase_delete_record ).

Chaque champ est converti au format approprié PHP, hormis :

- Les dates, qui sont laissées au format chaîne
- Les entiers qui causeraient des dépassement de capacité (> 32 bits) sont retournés comme des chaînes.

## 6.7.13 dbase_numfields() : Compte le nombre de champs d'une base dBase

int **dbase_numfields** ( int dbase_identifier )

dbase_numfields retourne le nombre de champs (colonnes) de la base de données dbase_identifier . Les numéros de champs sont numérotés de 0 à dbase_numfields($db)−1 , tandis que les numéros d'enregistrements sont numérotés de 1 à dbase_numrecords($db) .

Utiliser dbase_numfields

```
<?php
 $rec = dbase_get_record($db, $recno);
 $nf  = dbase_numfields($db);
 for ($i=0; $i < $nf; $i++) {
   print $rec[$i]."<BR>\n";
 }
?>
```

## 6.7.14 dbase_numrecords() : Compte le nombre d'enregistrements dans une base dBase

int **dbase_numrecords** ( int dbase_identifier )

dbase_numrecords retourne le nombre d'enregistrements (lignes) dans la base dbase_identifier . Les numéros de champ sont numérotés de 0 à dbase_numfields($db)−1 , tandis que les numéros d'enregistrements sont numérotés de 1 à dbase_numrecords($db) .

## 6.7.15 dbase_open() : Ouvre une base dBase

int **dbase_open** ( string filename , int flags )

dbase_open retourne un identifiant de base de données, représentant la base dBase contenue dans le fichier filename ou FALSE si la base n'a pas pu être sélectionnée.

flags est une option, comme pour la fonction open() . (Typiquement, 0 signifie lecture seule, 1 signifie écriture seule, et 2 écriture/lecture).

dbase_open retourne un identifiant de base de données ou FALSE si la base n'a pas pu être sélectionnée.

| Note |
| --- |
| Lorsque le safe−mode est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |

## 6.7.16 dbase_pack() : Compacte une base dBase

bool **dbase_pack** ( int dbase_identifier )

dbase_pack compacte la base de données dbase_identifier (effacement définitif de tous les enregistrements marqués pour l'effacement, avec la fonction dbase_delete_record ).

## 6.7.17 dbase_replace_record() : Remplace un enregistrement dans une base dBase

bool **dbase_replace_record** ( int dbase_identifier , array record , int dbase_record_number )

dbase_replace_record remplace les données associées à l'enregistrement dbase_record_number par les données enregistrées dans record , dans la base dbase_identifier . Si le nombre de colonnes fourni n'est pas celui du nombre de champ dans la base, l'opération échouera, et FALSE sera retourné.

dbase_record_number est un entier qui peut aller de 1 jusqu'au nombre maximal d'enregistrements de la base (retourné par dbase_numrecords ).

# 6.8 Accès aux dossiers

## 6.8.1 Introduction

## 6.8.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.8.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.8.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.8.5 Types de ressources

## 6.8.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*DIRECTORY_SEPARATOR ( string )*
*PATH_SEPARATOR ( string )*

## 6.8.7 Voir aussi

Pour des fonctions connexes telles que dirname , is_dir , mkdir et rmdir , voyez donc les fonctions Système de fichiers .


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Voir aussi
- chdir
- chroot
- closedir
- dir
- getcwd
- opendir
- readdir

- rewinddir
- scandir

## 6.8.9 chroot() : Change le dossier racine

bool **chroot** ( string directory )

chroot change la racine du script en cours, et la remplace par directory . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

| Note |
| --- |
| Il n'est pas conseillé d'utiliser cette fonction sur un site web, car il n'est pas possible de restaurer la racine à sa valeur initiale à la fin de la requête. Cette fonction n'est viable que lorsque PHP est utilisé en CGI. |

| Note |
| --- |
| Cette fonction n'est pas implémentée sous Windows. |

## 6.8.10 closedir() : Ferme le pointeur sur le dossier

void **closedir** ( resource dir_handle )

closedir ferme le pointeur de dossier dir_handle . Le dossier devait avoir été ouvert avec opendir .

## 6.8.11 dir() : Classe dossier

new **dir** ( string directory )

Un mécanisme pseudo−objet permet la lecture d'un dossier. L'argument directory doit être ouvert. Deux propriétés sont disponibles une fois le dossier ouvert : le pointeur peut être utilisé avec d'autres fonctions telles que readdir , rewinddir et closedir . Le chemin du dossier est le chemin fourni lors de la construction de l'objet. Trois méthodes permettent de lire, remettre à zéro et fermer le dossier.

Exemple avec dir

```
 <?php
$d = dir("/etc");
echo "Pointeur: ".$d->handle."<BR>\n";
echo "Chemin: ".$d->path."<BR>\n";
while($entry=$d->read()) {
    echo $entry."<BR>\n";
}
$d->close();
?>
```

## 6.8.12 getcwd() : Retourne le dossier de travail

string **getcwd** ( void )

getcwd retourne le nom du dossier courant.

Voir aussi chdir .

## 6.8.13 opendir() : Ouvre un dossier, et récupère un pointeur dessus.

resource **opendir** ( string path )

opendir retourne un pointeur sur un dossier pour être utilisé avec les fonctions closedir , readdir et rewinddir .

Si le paramètre path n'est pas un dossier valide, ou si le dossier ne peut être accédé pour des raisons de permissions ou des erreurs liées au système de fichiers, opendir retourne FALSE et génère une erreur PHP. Vous pouvez supprimer cette erreur en ajoutant @ avant le nom de la fonction.

Exemple avec opendir

```
 <?php
$dir = "/tmp/";

// Ouvre un dossier bien connu, et liste tous les fichiers
if (is_dir($dir)) {
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            print "fichier : $file : type: " . filetype($dir . $file) . "\n";
        }
    closedir($dh);
    }
}
?>
```

Voir aussi is_dir , readdir et dir

## 6.8.14 readdir() : Lit une entrée du dossier

string **readdir** ( resource dir_handle )

readdir retourne le nom du fichier suivant dans le dossier identifié par dir_handle . Les noms sont retournés dans n'importe quel ordre.

Liste tous les fichiers du dossier courant

```
 <?php
  $handle=opendir('.');
  echo "Pointeur de dossier : $handle\n";
  echo "Fichiers :\n";
  while ($file = readdir($handle)) {
    echo "$file\n";
  }
  closedir($handle);
?>
```

Notez que readdir retournera aussi les dossiers "." et "..". Si vous ne les voulez pas, supprimez les simplement :

Liste tous les fichiers du dossier courant, sauf "." et ".."

```
 <?php
$handle=opendir('.');
while ($file = readdir($handle)) {
    if ($file != "." && $file != "..") {
        echo "$file\n";
    }
}
closedir($handle);
?>
```

Voir aussi is_dir et glob .

## 6.8.15 rewinddir() : Retourne à la première entrée du dossier

void **rewinddir** ( resource dir_handle )

rewinddir retourne à la première entrée du dossier identifié par dir_handle : le prochain fichier lu sera le premier.

## 6.8.16 scandir() : Liste les fichiers et dossier dans un dossier

array **scandir** ( string directory , *int sorting_order* )

**scandir** retourne un array de fichier et dossier, issus de directory . Si directory n'est pas un dossier, le booléen FALSE est retourné, et une erreur de niveau E_WARNING est générée.

Par défaut, les fichiers sont donnés par ordre alphabétique. Si le paramètre optionnel sorting_order est utilisé avec la valeur 1, le tri sera fait en ordre alphabétique inverse.

Listage simple de dossier avec scandir

```
 <?php
$dir    = '/tmp';
$files1 = scandir($dir);
$files2 = scandir($dir, 1);

print_r($files1);
print_r($files2);

/* Affichera ceci :
 Array
(
    [0] => .
    [1] => ..
    [2] => bar.php
    [3] => foo.txt
    [4] => somedir
)
Array
(
    [0] => somedir
    [1] => foo.txt
    [2] => bar.php
    [3] => ..
    [4] => .
)
*/
?>
```

Alternatives à scandir

```
 <?php
$dir = "/tmp";
$dh  = opendir($dir);
while (false !== ($filename = readdir($dh))) {
    $files[] = $filename;
}

sort($files);

print_r($files);

rsort($files);

print_r($files);

/* Affichera ceci :
 Array
(
    [0] => .
    [1] => ..
    [2] => bar.php
    [3] => foo.txt
    [4] => somedir
)
Array
(
    [0] => somedir
    [1] => foo.txt
    [2] => bar.php
```

```
    [3] => ..
    [4] => .
)
*/
?>
```

Voir aussi opendir , readdir , glob , is_dir et sort .

# 6.9 Gestion des erreurs

## 6.9.1 Introduction

Ces fonctions permettent de gérer les erreurs, et de les enregistrer. Vous pouvez définir les règles de traitement des erreurs et choisir la manière de les enregistrer : vous pouvez adapter le rapport d'erreurs à vos besoins.

Avec les fonctions d'enregistrements, vous pouvez envoyer directement les rapports à d'autres machines (ou même les envoyer par email à un pager), à l'historique système, ou encore sélectionner les erreurs les plus importantes et ne pas enregistrer les autres.

La fonction de niveau d'erreur vous permet de personnaliser le niveau et le type d'erreur noté : depuis les inoffensives alertes jusqu'au erreurs personnalisées retournées par les fonctions.

## 6.9.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.9.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.9.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| error_reporting | E_ALL & ~E_NOTICE & ~E_STRICT | PHP_INI_ALL |
| display_errors | "1" | PHP_INI_ALL |
| display_startup_errors | "0" | PHP_INI_ALL |
| log_errors | "0" | PHP_INI_ALL |
| log_errors_max_len | "1024" | PHP_INI_ALL |
| ignore_repeated_errors | "0" | PHP_INI_ALL |
| ignore_repeated_source | "0" | PHP_INI_ALL |
| report_memleaks | "1" | PHP_INI_SYSTEM |
| track_errors | "0" | PHP_INI_ALL |
| html_errors | "1" | PHP_INI_ALL |
| docref_root | "http://www.php.net/" | PHP_INI_ALL |
| docref_ext | ".html" | PHP_INI_ALL |
| error_prepend_string | NULL | PHP_INI_ALL |
| error_append_string | NULL | PHP_INI_ALL |
| error_log | NULL | PHP_INI_ALL |
| warn_plus_overloading | NULL | PHP_INI?? |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*error_reporting integer*

Fixe le niveau d'erreur. Ce paramètre est un entier, représentant un champs de bits. Ajoutez les valeurs suivantes pour choisir le niveau que vous désirez, telles que décrite dans la section Constantes prédéfinies , et dans le fichier php.ini . Pour

modifier cette configuration durant l'exécution du script, utilisez la fonction error_reporting . Voyez aussi la directive display_errors .

En PHP 4, la configuration par défaut est E_ALL & ~ E_NOTICE . Elle montre toutes les erreurs, sauf les E_NOTICE . Il est recommandée de les afficher durant le développement.

En PHP 5, la configuration par défaut est E_ALL & ~ E_NOTICE & ~ E_STRICT . Elle montre toutes les erreurs, sauf les E_NOTICE et les E_STRICT . Il est recommandée de les afficher durant le développement.

| Note |
| --- |
| Activer le rapport d'erreur de niveau E_NOTICE durant le développement a des avantages. En terme de débogage, les message d'alertes vous signalent des bugs potentiels dans votre code. Par exemple, l'utilisation de valeurs non initialisées est signalé. Il est aussi plus pratique pour trouver des coquilles, et ainsi, gagner du temps. Les messages NOTICE vous signaleront aussi les mauvaises pratiques de codages. Par exemple $arr[item] doit toujours être écrit $arr['item'] car PHP va considéder "item" comme une constante, au premier abord. Si cette constante n'est pas définie, alors il va l'utilise comme une chaîne. |

En PHP 3, la configuration par défaut est (E_ERROR | E_WARNING | E_PARSE) , ce qui correspond à la même configuration qu'en PHP 4. Notez toutefois que les constantes ne sont pas supportées dans le php3.ini de PHP 3, et que le niveau d'erreur doit être spécifié numériquement, c'est à dire 7 .

*display_errors boolean*
: Cette directive détermine si les erreurs doivent être affichées à l'écran au format HTML ou non.

*display_startup_errors boolean*
: Même si display_errors est activé, des erreurs peuvent survenir lors de la séquence de démarrage de PHP, et ces erreurs sont cachées. Avec cette option, vous pouvez les afficher, et c'est recommandé pour le débogage. En dehors de ce cas, il est fortement recommandé de laisser display_startup_errors à off.

*log_errors boolean*
: Indique où les messages d'erreur générés doivent être écrits, dans l'historique du serveur ou dans error_log . Cette fonction est spécifique aux serveurs.

| Note |
| --- |
| Il est recommandé d'utiliser l'historique d'erreur, plutôt que d'afficher les erreurs sur les sites de production. |

*log_errors_max_len integer*
: Configure la taille maximale des erreurs qui seront enregistrées dans l'historique, en kilo octets. Dans les informations de error_log , l'origine est ajoutée. La valeur par défaut est de 1024. 0 signifie qu'il n'y a pas de limite de taille.

*ignore_repeated_errors boolean*
: Ne pas enregistrer des messages répétitifs. Les erreurs répétées doivent survenir au même moment, à la même ligne et depuis le même fichier de script, jusqu'à ce que ignore_repeated_source soit mis à TRUE .

*ignore_repeated_source boolean*
: Ignore la source du message lors des messages répétés. Lorsuqe vous avez configuré cette option à On, vous n'enregistrerez par les erreurs les erreurs répétées provenant de fichier et lignes de code différents.

*report_memleaks boolean*
: Si ce paramètre est mis à Off, alors les fuites de mémoires ne seront pas affichées (sur la sortie standard, stdout ou dans les logs). Cette option n'a d'effet que si PHP a été compilé avec l'option de débogage, et si error_reporting inclut E_WARNING dans

sa liste.

*track_errors boolean*

Si cette option est activée, le dernier message d'erreur sera placé dans la variable globale $php_errormsg .

*html_errors boolean*

Désactive les balises HTML dans les messages d'erreurs. Le nouveau format d'erreurs HTML fournit des messages cliquable, qui redirige l'utilisateur vers la documentation de l'erreur ou de la fonction. Ces références sont affectées par docref_root et docref_ext .

*docref_root string*

Le nouveau format d'erreur contient une référence à une page décrivant l'erreur, ou la fonction ayant causé l'erreur. Pour le manuel, vous pouvez télécharger le manuel dans votre langue, et configurer cette option pour qu'elle pointe sur votre manuel. Si votre copie du manuel est accessible à '/manual/', vous pouvez simplement utiliser docref_root=/manual/ . De plus, vous devez configurer docref_ext pour qu'elle corresponde aux extensions de votre manuel. docref_ext=.html . Il est possible d'utiliser des références externes. Par exemple, vous pouvez utiliser docref_root=http://manual/en/ ou docref_root="http://landonize.it/?how=url&theme=classic&filter=Landon&url=http%3A%2F%2Fw

La plupart du temps, vous utilisez l'option docref_root avec un slash a la fin ('/'). Mais ce n'est pas obligatoire, comme le montre le second exemple ci−dessus.

*docref_ext string*

Voir aussi docref_root .

| Note |
| --- |
| La valeur de docref_ext doit commencer avec un point '.'. |

*error_prepend_string string*

La chaîne à placer avant les messages d'erreur.

*error_append_string string*

La chaîne à placer après les messages d'erreur.

*error_log string*

Nom du fichier où seront enregistrées les erreurs. Si la valeur spéciale syslog est utilisée, les erreurs seront envoyées au système d'historique du serveur. Sous Unix, cela correspond à syslog(3) et sous Windows NT, à l'historique d'événement. L'historique n'est pas supporté sous Windows 95. Voir aussi : syslog .

*warn_plus_overloading boolean*

Si cette option est activée, PHP va afficher une alerte lorsque l'opérateur d'addition ( + ) est utilisé avec des chaînes de caractères. Cela peut aider à trouver les erreurs où le plus est utilisé comme opérateur de concaténation au lieu de point ( . ).

## 6.9.5 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

| Note |
| --- |
| Vous pouvez utiliser ces constantes dans le fichier php.ini mais pas hors de PHP , comme dans le fichier httpd.conf , où vous devez utiliser les valeurs de champs de bits. |

| Valeur | Constante | Description | Note |
| --- | --- | --- | --- |
| 1 | E_ERROR ( integer ) | Les erreurs sont aussi affichées par défaut, et l'exécution du script est interrompue. Elles indiquent des erreurs qui ne peuvent pas être ignorées, comme des problèmes d'allocation de mémoire, par exemple. | |
| 2 | E_WARNING ( integer ) | Les alertes sont affichées par défaut, mais n'interrompent pas l'exécution du script. Elles indiquent un problème qui doit être intercepté par le script durant l'exécution du script. Par exemple, appeler ereg avec une regex invalide. | |
| 4 | E_PARSE ( integer ) | Les erreurs d'analyse ne doivent être générées que par l'analyseur. Elles ne sont | |

| | | | |
|---|---|---|---|
| | | citées ici que dans le but d'être exhaustif. | |
| 8 | E_NOTICE ( integer ) | Les notes ne sont pas affichées par défaut, et indique que le script a rencontré quelque chose qui peut être une erreur, mais peut aussi être un événement normal dans la vie du script. Par exemple, essayer d'accéder à une valeur qui n'a pas été déclarée, ou appeler stat sur un fichier qui n'existe pas. | |
| 16 | E_CORE_ERROR ( integer ) | Elles sont similaires aux erreurs E_ERROR , mais elles sont générées par le code de PHP . Les fonctions ne doivent pas générer ce genre d'erreur. | PHP 4 seulement |
| 32 | E_CORE_WARNING ( integer ) | Elles sont similaires à E_WARNING, mais elles sont générées par le code de PHP . Les fonctions ne doivent pas générer ce genre d'erreur. | PHP 4 seulement |
| 64 | E_COMPILE_ERROR ( integer ) | Elles sont similaires à E_ERROR, mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur. | PHP 4 seulement |
| 128 | E_COMPILE_WARNING ( integer ) | Elles sont similaires à E_WARNING, mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur. | PHP 4 seulement |
| 256 | E_USER_ERROR ( integer ) | E_USER_ERROR est comparable à E_ERROR. Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur. | PHP 4 seulement |
| 512 | E_USER_WARNING ( integer ) | E_USER_WARNING est comparable à E_WARNING. Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur. | PHP 4 seulement |
| 1024 | E_USER_NOTICE ( integer ) | E_USER_WARNING est comparable à E_NOTICE. Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur. | PHP 4 seulement |
| 2048 | E_STRICT ( integer ) | Notices au moment de l'exécution. Permet d'obtenir des suggestions de PHP pour modifier votre code, assurant ainsi une meilleure interopérabilité et compatibilité de celui-ci. | PHP 5 seulement. |
| 4095 | E_ALL ( integer ) | Toutes les erreurs et alertes supportées. | A partir de PHP 5.0.0 la valeur a changé de 2047 à 4095. |

Les valeurs ci-dessus (numérique ou symbolique) sont utilisées pour constituer des champs de bits, qui spécifie le niveau de rapport d'erreur. Vous pouvez utiliser les opérateurs de bits pour combiner ces valeurs pour en faire des masques qui filtrent certaines erreurs. Notez bien que seuls '|', '~', '!', '^' et '&' seront compris dans le fichier php.ini , et que aucun opérateur logique ne sera compris en php3.ini .

## 6.9.6 Exemples

Ci-dessous, vous trouverez un exemple de gestion des erreurs par PHP. Il y est défini un gestionnaire d'erreur, qui enregistre les informations dans un fichier (au format XML), et envoie un courriel au développeur si l'erreur est critique.

Gestion d'erreurs avancées en PHP

```
 <?php
// Nous allons faire notre propre gestion
error_reporting(0);

// Fonction spéciale de gestion des erreurs
function userErrorHandler ($errno, $errmsg, $filename, $linenum, $vars) {
    // Date et heure de l'erreur
    $dt = date("Y-m-d H:i:s (T)");

    // Définit un tableau associatif avec les chaînes d'erreur
    // En fait, les seuls niveaux qui nous interessent
    // sont 2,8,256,512 et 1024
    $errortype = array (
                1   =>  "Erreur",
                2   =>  "Alerte",
                4   =>  "Error d'analyse",
                8   =>  "Note",
                16  =>  "Core Error",
                32  =>  "Core Warning",
                64  =>  "Compile Error",
                128 =>  "Compile Warning",
                256 =>  "Erreur spécifique",
                512 =>  "Alerte spécifique",
                1024=>  "Note spécifique"
                );
    // Les niveaux qui seront enregistrés
    $user_errors = array(E_USER_ERROR, E_USER_WARNING, E_USER_NOTICE);

    $err = "<errorentry>\n";
    $err .= "\t<datetime>".$dt."</datetime>\n";
    $err .= "\t<errornum>".$errno."</errornum>\n";
    $err .= "\t<errortype>".$errortype[$errno]."</errortype>\n";
    $err .= "\t<errormsg>".$errmsg."</errormsg>\n";
    $err .= "\t<scriptname>".$filename."</scriptname>\n";
    $err .= "\t<scriptlinenum>".$linenum."</scriptlinenum>\n";

    if (in_array($errno, $user_errors))
        $err .= "\t<vartrace>".wddx_serialize_value($vars,"Variables")."</vartrace>\n";
```

```
    $err .= "</errorentry>\n\n";

    // test
    // echo $err;

    // sauvegarde de l'erreur, et mail si c'est critique
    error_log($err, 3, "/usr/local/php4/error.log");
    if ($errno == E_USER_ERROR)
        mail("phpdev@example.com","Critical User Error",$err);
}


function distance ($vect1, $vect2) {
    if (!is_array($vect1) || !is_array($vect2)) {
        trigger_error("Incorrect parameters, arrays expected", E_USER_ERROR);
        return NULL;
    }

    if (count($vect1) != count($vect2)) {
        trigger_error("Vectors need to be of the same size", E_USER_ERROR);
        return NULL;
    }

    for ($i=0; $i<count($vect1); $i++) {
        $c1 = $vect1[$i]; $c2 = $vect2[$i];
        $d = 0.0;
        if (!is_numeric($c1)) {
            trigger_error("Coordinate $i in vector 1 is not a number, using zero",
                          E_USER_WARNING);
            $c1 = 0.0;
        }
        if (!is_numeric($c2)) {
            trigger_error("Coordinate $i in vector 2 is not a number, using zero",
                          E_USER_WARNING);
            $c2 = 0.0;
        }
        $d += $c2*$c2 - $c1*$c1;
    }
    return sqrt($d);
}

$old_error_handler = set_error_handler("userErrorHandler");

// constante non définie, qui génère une alerte
$t = I_AM_NOT_DEFINED;

// définition de quelques vecteurs
$a = array(2,3,"foo");
$b = array(5.5, 4.3, -1.6);
$c = array (1,-3);

// génère une erreur utilisateur
$t1 = distance($c,$b)."\n";

// génère une erreur utilisateur
$t2 = distance($b,"i am not an array")."\n";

// Génère une alerte
$t3 = distance($a,$b)."\n";

?>
```

## 6.9.7 Voir aussi

Voir aussi syslog .

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Constantes prédefinies
- Exemples
- Voir aussi
- debug_backtrace
- debug_print_backtrace
- error_log
- error_reporting
- restore_error_handler

- set_error_handler
- trigger_error
- user_error

## 6.9.9 debug_print_backtrace() : Affiche une backtrace

void **debug_print_backtrace** ( void )

debug_print_backtrace affiche la backtrace de PHP .

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Voir aussi debug_backtrace .

## 6.9.10 error_log() : Stocke un message d'erreur

int **error_log** ( string message , *int message_type* , *string destination* , *string extra_headers* )

error_log envoie un message d'erreur à l'historique du serveur web, à un port TCP ou un fichier. message est le message d'erreur qui doit être enregistré. message_type indique où le message doit être envoyé :

| | |
| --- | --- |
| 0 | message est envoyé à l'historique PHP, qui est basé sur l'historique système ou un fichier, en fonction de la configuration de error_log. |
| 1 | message est envoyé par email à l'adresse destination . C'est le seul type qui utilise le quatrième paramètre extra_headers . Ce message utilise la même fonction interne que mail . |
| 2 | message est envoyé par la connexion de debuggage PHP. Cette option n'est disponible que si l'option remote_debugging a été désactivée. Dans ce cas, le paramètre destination spécifie l'hôte ou l'adresse IP, et optionnellement le numéro de port, de la socket qui recevra les informations de débuggage. |
| 3 | message est ajouté au fichier destination . |

| Attention |
| --- |
| Le débuggage à distance via TCP/IP est une fonctionnalité PHP 3 qui **n'est pas** disponible en PHP 4. |

Exemples avec error_log

```
<?php
// Envoi une notification par l'historique du serveur, si la connexion à la base
// de données est impossible.
if (!Ora_Logon ($username, $password)) {
    error_log ("Base Oracle indisponible!", 0);
}
// Indiquer à l'administrateur, par email, qu'il n'y a plus de FOO
if (!($foo = allocate_new_foo())) {
    error_log ("Aya!, Il ne reste plus de FOO disponibles!", 1,
               "operateur@mondomaine.com");
}
// D'autres manières d'appeler error_log():
error_log ("Grosse bourde!", 2, "127.0.0.1:7000");
error_log ("Grosse bourde!", 2, "loghost");
error_log ("Grosse bourde!", 3, "/var/tmp/my-errors.log");
?>
```

## 6.9.11 error_reporting() : Fixe le niveau de rapport d'erreurs PHP

int **error_reporting** ( *int level* )

error_reporting fixe le niveau de rapport d'erreur PHP et retourne l'ancienne valeur. Le niveau d'erreur peut être un champ de bits, ou une constante. L'utilisation des constantes est vivement

recommandée, pour assurer une compatibilité maximale avec les futures versions. Au fur et à mesure que de nouveaux niveaux d'erreurs sont créés, l'intervalle de validité des niveaux évolue, et les anciennes valeurs n'ont plus les mêmes significations.

Exemple de modification de niveau d'erreur

```
<?php
 error_reporting (55);
// En PHP 3, équivalent à E_ALL ^ E_NOTICE
// ...en PHP 4, '55' signifie (E_ERROR | E_WARNING | E_PARSE | E_CORE_ERROR | E_CORE_WARNING)

error_reporting (2039);
 // PHP 4 équivalent à E_ALL ^ E_NOTICE

error_reporting (E_ALL ^ E_NOTICE);
```

Suivez les liens de chaque valeur interne pour connaître leur signification :

| constante | valeur |
|---|---|
| 1 | E_ERROR |
| 2 | E_WARNING |
| 4 | E_PARSE |
| 8 | E_NOTICE |
| 16 | E_CORE_ERROR |
| 32 | E_CORE_WARNING |
| 64 | E_COMPILE_ERROR |
| 128 | E_COMPILE_WARNING |
| 256 | E_USER_ERROR |
| 512 | E_USER_WARNING |
| 1024 | E_USER_NOTICE |
| 2048 | E_STRICT |
| 4096 | E_ALL |

Attention

A partir de PHP 5.0.0, la constante E_STRICT avec comme valeur 2048 est introduite. La valeur de E_ALL est donc passé à 4096.

Voir aussi la directive display_errors et ini_set .

## 6.9.12 restore_error_handler() : Réactive l'ancienne fonction de gestion des erreurs

void **restore_error_handler** ( void )

Utilisée après avoir modifié la fonction de gestion des erreurs, grâce à set_error_handler , restore_error_handler permet de réutiliser l'ancienne version de gestion des erreurs (qui peut être la fonction PHP par défaut, ou une autre fonction utilisateur).

Voir aussi error_reporting , set_error_handler et trigger_error .

## 6.9.13 set_error_handler() : Choisi une fonction utilisateur comme gestionnaire d'erreurs

string **set_error_handler** ( callback error_handler )

set_error_handler choisit la fonction utilisateur error_handler pour gérer les erreurs dans un script. Retourne un pointeur sur l'ancienne fonction de gestion des erreurs (si il y en avait une), ou FALSE , en cas d'erreur. set_error_handler sert à définir votre propre gestionnaire d'erreurs, qui prendra en charge leur traitement durant l'exécution d'un script. Cela peut être utile lorsque vous devez repérer des erreurs critiques lors d'un nettoyage de bases, ou bien si vous souhaitez générer une erreur dans certaines conditions (avec trigger_error ).

La fonction utilisateur doit accepter deux arguments : le code de l'erreur, et une chaîne décrivant l'erreur. L'exemple ci dessous montre le traitement d'exceptions en déclenchant des erreurs, et en les gérant avec une fonction utilisateur.

| Note |
| --- |
| Au lieu d'un nom de fonction, vous pouvez utiliser un tableau contenant deux éléments, une référence sur un objet et un nom de méthode. Cette fonctionnalité a été ajoutée en PHP 4.3.0. |

| Note |
| --- |
| Les types d'erreur suivants ne peuvent pas être gérés avec cette fonction : E_ERROR , E_PARSE , E_CORE_ERROR , E_CORE_WARNING , E_COMPILE_ERROR et E_COMPILE_WARNING . |

L'exemple ci−dessous illustre l'interception d'erreurs internes avec génération d'erreur, et exploitation dans une fonction utilisateur :

Traitement des erreurs avec set_error_handler et trigger_error

```
 <?php

// redéfini les constantes utilisateur (PHP 4 Uniquement)
define ("FATAL",E_USER_ERROR);
define ("ERROR",E_USER_WARNING);
define ("WARNING",E_USER_NOTICE);

// configure le niveau de rapport d'erreur pour ce script
error_reporting (FATAL | ERROR | WARNING);

// gestionnaire d'erreurs
function myErrorHandler ($errno, $errstr, $errfile, $errline) {
  switch ($errno) {
  case FATAL:
    echo "<b>FATAL</b> [$errno] $errstr<br>\n";
    echo "  Fatal error in line ".$errline." of file ".$errfile;
    echo ", PHP ".PHP_VERSION." (".PHP_OS.")<br>\n";
    echo "Aborting...<br>\n";
    exit(1);
    break;
  case ERROR:
    echo "<b>ERROR</b> [$errno] $errstr<br>\n";
    break;
  case WARNING:
    echo "<b>WARNING</b> [$errno] $errstr<br>\n";
    break;
    default:
    echo "Unkown error type: [$errno] $errstr<br>\n";
    break;
  }
}

// fonction pour tester la gestion d'erreur
function scale_by_log ($vect, $scale) {
  if ( !is_numeric($scale) || $scale <= 0 )
    trigger_error("log(x) for x <= 0 is undefined, you used: scale = $scale",
      FATAL);
  if (!is_array($vect)) {
    trigger_error("Incorrect input vector, array of values expected", ERROR);
    return null;
  }
  for ($i=0; $i<count($vect); $i++) {
    if (!is_numeric($vect[$i]))
      trigger_error("Value at position $i is not a number, using 0 (zero)",
         WARNING);
    $temp[$i] = log($scale) * $vect[$i];
  }
  return $temp;
}

// Configuration du gestionnaire d'erreurs
$old_error_handler = set_error_handler("myErrorHandler");

// génération de quelques erreurs. Commençons par créer un tableau
echo "vecteur a \n";
$a = array(2,3,"foo",5.5,43.3,21.11);
print_r($a);

// générons maintenant un autre tableau, avec des alertes
echo "----\nvecteur b − une alerte (b = log(PI) * a)\n";
$b = scale_by_log($a, M_PI);
print_r($b);

// ceci est un problème, nous avons utilisé une chaîne au lieu d'un tableau
echo "----\nvecteur c − une erreur\n";
$c = scale_by_log("not array",2.3);
```

```
var_dump($c);

// Ceci est une erreur critique : le logarithme de zéro ou d'un nombre négatif est indéfini
echo "----\nvecteur d - erreur fatale\n";
$d = scale_by_log($a, -2.5);

?>
```

L'exécution du script devrait donner ceci :

```
Résultat de set_error_handler et trigger_error
 vecteur a
 Array
(
    [0] => 2
    [1] => 3
    [2] => foo
    [3] => 5.5
    [4] => 43.3
    [5] => 21.11
)
----
vecteur b - une alerte (b = log(PI) * a)
<b>WARNING</b> [1024] Value at position 2 is not a number, using 0 (zero)<br>
Array
(
    [0] => 2.2894597716988
    [1] => 3.4341896575482
    [2] => 0
    [3] => 6.2960143721717
    [4] => 49.566804057279
    [5] => 24.165247890281
)
----
vecteur c - une erreur
<b>ERROR</b> [512] Incorrect input vector, array of values expected<br>
NULL
----
vecteur d - erreur fatale
<b>FATAL</b> [256] log(x) for x <= 0 is undefined, you used: scale = -2.5<br>
  Fatal error in line 37 of file /Users/imac/-, PHP 5.0.0-dev (Darwin)<br>
Aborting...<br>
```

Il faut se rappeler que la fonction standard de traitement des erreurs de PHP est alors complètement ignorée. error_reporting n'aura plus d'effet, et votre fonction de gestion des erreurs sera toujours appelée. Vous pourrez toujours lire la valeur de l'erreur courante de error_reporting et faire réagir la fonction de gestion des erreurs en fonction. Cette remarque est notamment valable si la commande a été préfixée par @ (0 sera retourné).

Notez aussi qu'il est alors confié à cette fonction de terminer le script ( die ) si nécessaire. Si la fonction de gestion des erreurs se termine normalement, l'exécution du script se poursuivra avec l'exécution de la prochaine commande.

| Note |
| --- |
| Si une erreur survient avant que le script ne soit exécuté (par exemple un téléchargement de fichier), le gestionnaire d'erreurs personnalisés ne pourra pas être appelé, car il n'est pas encore enregistré. |

Voir aussi error_reporting , restore_error_handler , trigger_error et les constantes de niveau d'erreurs .

## 6.9.14 trigger_error() : Déclenche une erreur utilisateur

void **trigger_error** ( string error_msg , *int* *error_type* )

trigger_error est utilisé pour déclencher une erreur utilisateur. Elle peut aussi être utilisée en conjonction avec un gestionnaire d'erreur interne, ou un gestionnaire d'erreurs utilisateur qui a été choisi comme gestionnaire d'erreur avec set_error_handler .

trigger_error est pratique lorsque vous devez générer une réponse particulière lors de l'exécution. Par exemple :

Exemple avec trigger_error

```
<?php
if (assert ($divisor == 0))
   trigger_error ("Impossible de diviser par zéro", E_USER_ERROR);
?>
```

Note

Voyez set_error_handler pour une illustration.

Note

error_msg est limité à 1024 caractères en longueur. Tous les caractères supplémentaires seront ignorés.

Voir aussi error_reporting , set_error_handler , restore_error_handler et les constantes de niveau d'erreurs .

## 6.9.15 user_error() : Alias de trigger_error

user_error est un alias de la fonction trigger_error .

# 6.10 Système de fichiers

## 6.10.1 Introduction

## 6.10.2 Pré−requis

Aucune librairie externe n'est requise pour avoir cette extension, mais si vous voulez le support LFS (grands fichiers) sur Linux, vous devez avoir une version récente de gclib et vous devrez compiler PHP avec les options suivantes du compilateur : −D_LARGEFILE_SOURCE −D_FILE_OFFSET_BITS=64 .

## 6.10.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.10.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| allow_url_fopen | "1" | PHP_INI_ALL |
| user_agent | NULL | PHP_INI_ALL |
| default_socket_timeout | "60" | PHP_INI_ALL |
| from | NULL | ?? |
| auto_detect_line_endings | "Off" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*allow_url_fopen booléen*

Cette option active les versions étendues des fonctions d'accès aux fichiers, qui savent exploiter les URL. Les extensions par défaut permettent d'accéder aux fichiers distants avec les protocoles FTP ou HTTP. Certaines extensions apparaissent comme zlib , qui ajoutent d'autres fonctionnalités.

| Note |
|---|
| Cette option a été introduite immédiatement après la publication de la version 4.0.3. Pour les versions jusqu'à la, vous ne pouvez désactiver cette fonctionnalité qu'au moment de la compliation, avec l'option −−disable−url−fopen−wrapper . |
| Attention |
| Sous Windows, dans les versions antérieures à la version 4.3, les fonctions suivantes ne supportent pas l'accès aux fichiers distants : include , include_once , require , require_once et les fonctions imagecreatefromXXX de l'extension Images . |

*user_agent string*

Définit le type d'"user agent" (Définition du navigateur web) utilisé par PHP .

*default_socket_timeout entier*

Durée d'expiration (en secondes) pour les flots basés sur les sockets.

| Note |
|---|
| Cette option de configuration a été introduite en PHP 4.3. |

*from="joe@example.com" string*

Définit le mot de passe FTP en mode anonyme (en général, votre adresse email).

*auto_detect_line_endings booléen*

Lorsque cette option est activée, PHP va examiner les données lues par fgets et file pour voir si le fichier utilise les conventions de ligne de Unix, MS−Dos ou Macintosh.

Cela permet à PHP d'interopérer avec des systèmes MacIntosh, mais par défaut, cette option est désactivée, car cette détection impose un légère pénalité en temps de traitement, mais aussi parce que ceux qui utilisent les retours chariots comme séparateurs auront des soucis de compatibilité.

| Note |
| --- |
| Cette option de configuration a été introduite en PHP 4.3. |

## 6.10.5 Types de ressources

## 6.10.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*GLOB_BRACE ( integer )*
*GLOB_ONLYDIR ( integer )*
*GLOB_MARK ( integer )*
*GLOB_NOSORT ( integer )*
*GLOB_NOCHECK ( integer )*
*GLOB_NOESCAPE ( integer )*
*PATHINFO_DIRNAME ( integer )*
*PATHINFO_BASENAME ( integer )*
*PATHINFO_EXTENSION ( integer )*

## 6.10.7 Voir aussi

Pour les fonctions connexes, voyez aussi la section sur les accès aux Dossiers et sur les exécutions de programme .

Pour une liste et une explication sur les différents outils d'accès aux fichiers distants, voyez aussi Liste des protocoles supportés .


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Voir aussi
- basename
- chgrp
- chmod
- chown
- clearstatcache
- copy

- delete
- dirname
- disk_free_space
- disk_total_space
- diskfreespace
- fclose
- feof
- fflush
- fgetc
- fgetcsv
- fgets
- fgetss
- file_exists
- file_get_contents
- file_put_contents
- file
- fileatime
- filectime
- filegroup
- fileinode
- filemtime
- fileowner
- fileperms
- filesize
- filetype
- flock
- fnmatch
- fopen
- fpassthru
- fputs
- fread
- fscanf
- fseek
- fstat
- ftell
- ftruncate
- fwrite
- glob
- is_dir
- is_executable
- is_file
- is_link
- is_readable
- is_uploaded_file
- is_writable
- is_writeable
- link
- linkinfo
- lstat
- mkdir
- move_uploaded_file
- parse_ini_file
- pathinfo
- pclose

- popen
- readfile
- readlink
- realpath
- rename
- rewind
- rmdir
- set_file_buffer
- stat
- symlink
- tempnam
- tmpfile
- touch
- umask
- unlink

## 6.10.9 chgrp() : Change le groupe propriétaire d'un fichier

bool **chgrp** ( string filename , mixed group )

chgrp essaie de remplacer le groupe propriétaire courant du fichier filename par group . Seul le super−utilisateur (root) peut changer le groupe propriétaire d'un fichier arbitrairement. Les utilisateurs classiques ne peuvent changer le groupe propriétaire d'un fichier que si l'utilisateur propriétaire du fichier est membre du groupe.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi chown et chmod .

## 6.10.10 chmod() : Change le mode du fichier

bool **chmod** ( string filename , int mode )

chmod remplace le mode du fichier filename par le mode mode .

Il est à noter que le mode mode est considéré comme un nombre en notation octale. Afin de vous en assurer, vous pouvez préfixer cette valeur par un zéro ( mode ):

```
Comment utiliser chmod
 chmod ("/somedir/somefile", 755);
// notation décimale : probablement faux
chmod ("/somedir/somefile", "u+rwx,go+rx");
 // chaîne : incorrect
chmod ("/somedir/somefile", 0755);
// notation octale : valeur du mode correcte
```

Le paramètre mode est constitué de trois valeurs octacles qui spécifient les droits pour le propriétaire, le groupe du propriétaire et les autres, respectivement. Chaque composant peut être calculé en ajoutant les droits désirés. Le chiffre 1 donne les droits d'exécution, le chiffre 2 les droits d'écriture et le chiffre 4 les droits de lecture. Ajoutez simplement ces nombres pour spécifier les droits voulus. Vous pouvez aussi lire le manuel des systèmes Unix avec 'man 1 chmod' et 'man 2 chmod'.

```
Comment utiliser le paramètre mode de chmod
```

```
 // Lecture et écriture pour le propriétaire, rien pour les autres
chmod ("/un_dossier/un_fichier", 0600);

// Lecture et écriture pour le propriétaire, écriture pour les autres
chmod ("/un_dossier/un_fichier", 0644);

// Tous pour le propriétaire, lecture et écriture pour les autres
chmod ("/un_dossier/un_fichier", 0755);

// Tout pour le propriétaire, lecture exécution pour le groupe, rien pour les autres
chmod ("/un_dossier/un_fichier", 0750);
```

| Note |
| --- |
| L'utilisateur courant est l'utilisateur avec lequel PHP fonctionne. Il est probablement différent de l'utilisateur que vous utilisez en mode shell ou FTP. |

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local. |

Voir aussi chown et chgrp .

## 6.10.11 chown() : Change le groupe propriétaire du fichier

bool **chown** ( string filename , mixed user )

chown change le groupe propriétaire du fichier. Seul le super−utilisateur (root) peut changer arbitrairement le propriétaire d'un fichier.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local. |

Voir aussi chmod .

## 6.10.12 clearstatcache() : Efface le cache de stat

void **clearstatcache** ( void )

L'appel à la fonction stat ou lstat est relativement coûteux en terme de temps d'exécution. Pour cela, le résultat du dernier appel à l'une des fonctions de statut, (voir la liste ci−dessous), est sauvegardé pour ré−utilisation ultérieure. Si vous voulez forcer la vérification du statut d'un fichier, dans le cas où le fichier aurait pu être modifié ou aurait disparu, vous devez utiliser la fonction clearstatcache afin d'effacer de la mémoire les résultats du dernier appel à la fonction.

La valeur du cache n'est valable que pour la durée d'une requête.

Les fonctions affectées sont : stat , lstat , file_exists , is_writable , is_readable , is_executable , is_file , is_dir , is_link , filectime , fileatime , filemtime , fileinode , filegroup , fileowner , filesize , filetype , et fileperms .

## 6.10.13 copy() : Copie un fichier

bool **copy** ( string source , string dest )

copy fait une copie du fichier source vers le fichier dest . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec copy

```
<?php
 if (!copy($file, $file.'.bak')) {
   print("La copie du fichier $file n'a pas réussi...<BR>\n");
 }
?>
```

Note

Depuis PHP 4.3.0, les deux paramètres source et dest peuvent être des URL si les gestionnaires d'URL ont été activés ("fopen wrappers"). Voyez la fonction fopen pour plus de détails. Si dest est une URL, la copie peut échouer si ce protocole ne supporte pas l'écrasement de fichiers existants.

Attention

Si le fichier de destination dest existe déjà, il sera écrasé.

Voir aussi move_uploaded_file et rename .

## 6.10.14 delete() : Effacer

void **delete** ( string file )

Ceci est une fausse entrée du manuel pour ceux qui recherchent en fait la fonction unlink ou unset .

Voir aussi unlink et unset .

## 6.10.15 dirname() : Renvoie le nom du dossier

string **dirname** ( string path )

dirname retourne le nom du dossier qui contient le fichier ou dossier path .

Sous Windows, les slash ( / ) et anti−slash ( \ ) sont utilisés comme séparateurs de dossier. Dans les autres environnements, seul le slash ( / ) est utilisé.

Exemple avec dirname

```
<?php
 $path = "/etc/passwd";
 $file = dirname ($path);
// $file vaut "/etc"
?>
```

Note

En PHP 4.0.3, dirname a été modifié pour être compatible POSIX. Cela signifie notamment que si il n'y a pas de slash dans le paramètre path , un point (' . ') est retourné, indiquant le dossier courant. Sinon, la chaîne retournée est path avec le composant final / supprimé. Notez que cela signifie que vous allez obtenir un slash ou un point là où l'ancien comportement de dirname retournait une chaîne vide.

Voir aussi basename , pathinfo et realpath .

## 6.10.16 disk_free_space() : Renvoie l'espace disque disponible dans le répertoire.

float **disk_free_space** ( string directory )

disk_free_space retournera le nombre d'octets disponibles sur le disque contenant le dossier directory .

Exemple avec disk_free_space

```
<?php
 $df = disk_free_space("/");
 // $df contient le nombre d'octets libres sur "/"
?>
```

Note

Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

## 6.10.17 disk_total_space() : Retourne la taille d'un dossier

float **disk_total_space** ( string directory )

disk_total_space lit récursivement toutes les tailles du dossier directory et retourne la somme. directory peut être aussi une partition de disque.

Exemple avec disk_total_space

```
<?php
 $df = disk_total_space("/"); // $df contient le nombre d'octets libres
                              // dans le dossier "/"
?>
```

Note

Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local.

## 6.10.18 diskfreespace() : Alias de disk_free_space

float **diskfreespace** ( string directory )

diskfreespace est l'ancien nom de la fonction disk_free_space . Elle est obsolète, et il est vivement recommandé d'utiliser disk_free_space .

## 6.10.19 fclose() : Ferme un fichier

bool **fclose** ( resource handle )

fclose ferme le fichier handle .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le pointeur de fichier doit être valide et doit avoir été correctement ouvert par fopen ou fsockopen .

Exemple avec fclose

```
<?php
```

```
  $handle = fopen('un_fichier.txt', 'r');

  fclose($handle);

?>
```

## 6.10.20 feof() : Teste la fin du fichier

bool **feof** ( resource handle )

feof retourne TRUE si le pointeur handle est à la fin du fichier, ou si une erreur survient, sinon, retourne FALSE .

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par fopen , popen , ou fsockopen .

## 6.10.21 fflush() : Envoi tout le contenu généré dans un fichier

bool **fflush** ( resource handle )

fflush force l'écriture de toutes les données bufferisées dans le fichier désigné par handle . fflush retourne TRUE en cas de succès, et FALSE sinon.

handle est un pointeur de fichier ouvert avec fopen , popen , ou fsockopen .

## 6.10.22 fgetc() : Lit un caractère dans un fichier

string **fgetc** ( resource handle )

fgetc retourne une chaîne contenant un seul caractère, lu depuis le fichier pointé par handle . fgetc retourne FALSE à la fin du fichier (contrairement à feof ).

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par fopen , popen , ou fsockopen .

| Attention |
| --- |
| Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur exacte de cette fonction. |

Exemple avec fgetc

```
 <?php
$fp = fopen('unfichier.txt', 'r');
if (!$fp) {
    echo "Impossible d'ouvrir le fichier unfichier.txt";
}
while (false !== ($char = fgetc($fp))) {
    echo "$char\n";
}
?>
```

| Note |
| --- |
| Cette fonction gère les chaînes binaires |

Voir aussi fread , fopen , popen , fsockopen et fgets .

## 6.10.23 fgetcsv() : Renvoie la ligne courante et cherche les champs CSV

array **fgetcsv** ( resource handle , int length , *string delimiter* , *string enclosure* )

fgetcsv est identique à fgets mais fgetcsv analyse la ligne qu'il lit et recherche les champs CSV, qu'il va retourner dans un tableau les contenant. Le délimiteur de champs delimiter est la virgule, à moins que vous ne fournissiez un troisième argument.

| Note |
| --- |
| Le paramètre enclosure a été ajouté en PHP 4.3.0. |

handle doit être un pointeur valide, et avoir été correctement ouvert par fopen , popen , ou fsockopen .

length doit être plus grand que la plus grande ligne trouvée dans un fichier CSV (en comptant les caractères de fin de ligne).

fgetcsv retourne FALSE en cas d'erreur, ou en cas de fin du fichier.

| Note |
| --- |
| Une ligne vide dans un fichier CSV sera retournée sous la forme d'un tableau contenant la valeur NULL et ne sera pas traitée comme une erreur. |

Lit et affiche le contenu d'un fichier CSV avec fgetcsv

```
 <?php
$row = 1;
$fp = fopen ("test.csv","r");
while ($data = fgetcsv ($fp, 1000, ",")) {
    $num = count ($data);
    print "<p> $num champs dans la ligne $row: <br>\n";
    $row++;
    for ($c=0; $c < $num; $c++) {
        print $data[$c] . "<br>\n";
    }
}
fclose ($fp);
?>
```

Voir aussi explode , file et pack .

## 6.10.24 fgets() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier.

string **fgets** ( resource handle , *int length* )

fgets retourne la chaîne lue jusqu'à la longueur length − 1 octet, ou bien la fin du fichier, ou encore un retour chariot (le premier des trois qui sera rencontré).

Si une erreur survient, fgets retourne FALSE .

Erreur courante :

Les programmeurs habitués à la programmation 'C' noteront que fgets ne se comporte pas comme son équivalent C lors de la rencontre de la fin du fichier.

handle doit être valide, et avoir été correctement ouvert par fopen , popen , ou fsockopen .

Un exemple simple :

Lecture d'un fichier ligne par ligne

```
<?php
 $fd = fopen ("/tmp/inputfile.txt", "r");
 while (!feof ($fd)) {
   $buffer = fgets($fd, 4096);
   echo $buffer;
 }
 fclose ($fd);
?>
```

Note

Le paramètre length est devenu optionnel depuis PHP 4.2.0. Si il est omis, il prend la valeur par défaut de 1024. Depuis PHP 4.3.0, omettre le paramètre length fait que la fonction va continuer à lire le flux jusqu'à ce que la ligne s'achève. Si la majorité des lignes du fichier dépassent les 8 ko, il est plus efficace pour votre script de spécifier ce paramètre.

Note

fgets est compatible avec les données binaires depuis PHP 4.3. Les versions plus anciennes ne le sont pas.

Note

Si vous avez des problèmes avec PHP qui ne reconnait pas certaines lignes lors de la lecture de fichier qui ont été créé ou lus sur un MacIntosh, vous pouvez activer l'option de configuration auto_detect_line_endings .

Voir aussi fread , fgetc , stream_get_line , fopen , popen , fsockopen et socket_set_timeout .

## 6.10.25 fgetss() : Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier et élimine les balises HTML

string **fgetss** ( resource handle , int length , *string allowable_tags* )

fgetss est identique à fgets , mais fgetss supprime toutes les balises HTML et PHP qu'il trouve dans le texte lu.

Vous pouvez aussi préciser les balises qui seront ignorées dans le troisième paramètre optionnel.

Note

allowable_tags a été ajouté en PHP 3.0.13 et PHP 4.0.0.

Note

Si vous avez des problèmes avec PHP qui ne reconnait pas certaines lignes lors de la lecture de fichier qui ont été créé ou lus sur un MacIntosh, vous pouvez activer l'option de configuration auto_detect_line_endings .

Voir aussi fgets , fopen , fsockopen , popen et strip_tags .

## 6.10.26 file_exists() : Vérifie si un fichier existe

bool **file_exists** ( string filename )

file_exists retourne TRUE si le fichier filename existe, et FALSE sinon.

Note

Utilisation des dossiers partagés Windows

Sous Windows, utilisez le format de chemin //computername/share/filename ou \\\\computername\share\filename pour vérifier qu'un fichier est disponible sur le partage réseau.

Test d'existence d'un fichier

```
 <?php
$filename = '/chemin/jusqu/a/fichier.txt';

if (file_exists($filename)) {
    print "Le fichier $filename existe";
} else {
    print "Le fichier $filename n'existe pas";
}
?>
```

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

Array

Voir aussi is_readable , is_writable , is_file et file .

## 6.10.27 file_get_contents() : Lit tout un fichier dans une chaîne

string **file_get_contents** ( string filename , *bool use_include_path* , *resource context* )

file_get_contents est identique à la fonction readfile , hormis le fait que file_get_contents retourne le fichier filename dans une chaîne.

Note

Cette fonction a été introduite en PHP 4.3.0.

Note

Cette fonction gère les chaînes binaires

Array

Note

Le support de l'argument context sera ajouté en PHP 5.0.

Voir aussi fgets , file , fread , include et readfile .

## 6.10.28 file_put_contents() : Ecrit une chaîne dans un fichier

int **file_put_contents** ( string filename , string data , *int flags* , *resource context* )

file_put_contents revient à appeler successivement fopen , fwrite , et fclose .

Le paramètre flags peut valoir FILE_USE_INCLUDE_PATH et/ou FILE_APPEND , mais FILE_USE_INCLUDE_PATH doit être utilisé avec prudence.

Note

Cette fonction gère les chaînes binaires

Array

Voir aussi fopen , fwrite , fclose et file_get_contents .

## 6.10.29 file() : Lit le fichier et renvoie le résultat dans un tableau

array **file** ( string filename , *int use_include_path* , *resource context* )

file est identique à readfile , hormis le fait que file retourne le fichier dans un tableau. Chaque élément du tableau correspond à une ligne du fichier, et les retour−chariots sont placés en fin de ligne.

| Note |
| --- |
| Chaque élément du tableau résultat contiendra la nouvelle ligne de fin de chaîne. Il faudra donc utiliser rtrim sur cette valeur pour la supprimer. |

| Note |
| --- |
| Si vous avez des problèmes avec PHP qui ne reconnait pas certaines lignes lors de la lecture de fichier qui ont été créé ou lus sur un MacIntosh, vous pouvez activer l'option de configuration auto_detect_line_endings . |

Vous pouvez utiliser l'option use_include_path : en la mettant à "1", vous rechercherez aussi dans le dossier include_path .

Exemple avec file

```
 <?php
// Lit une page web dans un tableau.
$lines = file ('http://www.exemple.com/');

// Affiche toutes les lignes du tableau comme code HTML, avec les numéros de ligne
foreach ($lines as $line_num => $line) {
    echo "Ligne No <b>$line_num</b> : " . htmlspecialchars($line) . "<br>\n";
}

// Un autre exemple, pour obtenir une page web dans une chaîne. Voir aussi la fonction file_get_contents().
$html = implode ('', file ('http://www.exemple.com/'));
?>
```

| Note |
| --- |
| Depuis PHP 4.3.0, vous pouvez utiliser file_get_contents pour lire le contenu d'un fichier dans une chaîne.<br><br>En PHP 4.3.0, file est désormais compatible avec les données binaires. |

Array

Voir aussi readfile , fopen , fsockopen , popen , file_get_contents et include .

## 6.10.30 fileatime() : Renvoie la date à laquelle le fichier a été accédé pour la dernière fois.

int **fileatime** ( string filename )

fileatime renvoie la date à laquelle le fichier a été accédé pour la dernière fois, ou FALSE en cas d'erreur.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

## 6.10.31 filectime() : Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois.

int **filectime** ( string filename )

filectime renvoie l'heure à laquelle l'inode filename a été accédé pour la dernière fois, ou FALSE en cas d'erreur.

Note: Sur la plupart des serveurs UNIX, un fichier est considéré comme modifié si les données de son inode sont modifiées. C'est−à−dire lorsque les permissions (utilisateur, groupe ou autre) ont été modifiées. Voyez aussi filemtime (que vous pourrez utiliser lorsque vous créerez des indications telles que "Dernière modification : " sur les pages web) et fileatime .

Notez aussi que sur certains systèmes UNIX, le ctime d'un fichier texte est considéré comme sa date de création. Cela est faux! Il n'y a pas de date de création de fichier sous la plupart des systèmes UNIX.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

## 6.10.32 filegroup() : Lire le nom du groupe

int **filegroup** ( string filename )

filegroup renvoie le groupe qui possède le fichier filename , ou FALSE en cas d'erreur. L'identifiant de groupe est retourné au format numérique, utilisez posix_getgrgid pour retrouver le nom du groupe.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

## 6.10.33 fileinode() : Renvoie le numéro d'inode du fichier

int **fileinode** ( string filename )

fileinode renvoie le numéro d'inode du fichier filename , ou FALSE en cas d'erreur.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

## 6.10.34 filemtime() : Renvoie la date de dernière modification du fichier.

int **filemtime** ( string filename )

filemtime renvoie la date de dernière modification du fichier filename , ou FALSE en cas d'erreur.

filemtime retourne l'heure d'écriture des blocs données d'un fichier. Utilisez date sur ce résultat pour obtenir une date de modification humainement lisible.

Exemple avec filemtime

```
 <?php
// l'affichage sera : lefichier.txt a été modifié le : December 29 2002 22:16:23.

$filename = 'lefichier.txt';
if (file_exists($filename)) {
    echo "$filename  a été modifié le : " . date ("F d Y H:i:s.", filemtime($filename));
}
?>
```

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

Array

Voir aussi filectime , stat , touch et getlastmod .

## 6.10.35 fileowner() : Renvoie le nom du propriétaire du fichier

int **fileowner** ( string filename )

fileowner renvoie le nom du possesseur du fichier filename , ou FALSE en cas d'erreur. L'identification du possesseur de fichier est numérique : il faut utiliser posix_getpwuid pour retrouver le nom d'utilisateur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

Array

## 6.10.36 fileperms() : Renvoie les permissions affectées au fichier.

int **fileperms** ( string filename )

fileperms renvoie les permissions affectées au fichier filename , ou FALSE en cas d'erreur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

Array

## 6.10.37 filesize() : Renvoie la taille du fichier

int **filesize** ( string filename )

filesize renvoie la taille du fichier filename , ou FALSE en cas d'erreur.

Note

Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails.

Array

## 6.10.38 filetype() : Retourne le type de fichier

string **filetype** ( string filename )

filetype renvoie le type du fichier filename . Les réponses possibles sont : fifo, char, dir, block, link, file, et unknown.

filetype retourne FALSE en cas d'erreur.

filetype ne fonctionne pas sur les fichiers distants . Les fichiers doivent être accessibles par le système de fichier du serveur.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

## 6.10.39 flock() : Verrouille le fichier

bool **flock** ( resource handle , int operation , *int &wouldblock* )

PHP dispose d'un système complet de verrouillage de fichiers. Tous les programmes qui accèdent au fichier doivent utiliser la même méthode de verrouillage pour qu'il soit efficace.

flock agit sur le fichier handle qui doit avoir été ouvert au préalable. operation est une des valeurs suivantes :

- Acquisition d'un verrou en lecture : operation = LOCK_SH (valeur de 1 pour les versions plus anciennes que PHP 4.0.1).
- Acquisition d'un verrou exclusif en écriture : operation = LOCK_EX (valeur de 2 pour les versions plus anciennes que PHP 4.0.1).
- Libération d'un verrou partagé ou exclusif, operation = LOCK_UN (valeur de 3 pour les versions plus anciennes que PHP 4.0.1).
- Si vous voulez que flock ne se bloque pas durant le verrouillage, ajoutez LOCK_NB à operation (valeur de 4 pour les versions plus anciennes que PHP 4.0.1).

flock permet de réaliser un système simple de verrous écriture / lecture, qui peut être utilisé sur n'importe quelle plate−forme (Unix et Windows compris). Le troisième paramètre optionnel vaut par défaut TRUE , si le verrou doit bloquer le script (condition d'erreur EWOULDBLOCK).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Comme flock requiert un pointeur de fichier, vous aurez peut être à utiliser un verrou spécial pour protéger l'accès au fichier que vous voulez tronquer en l'ouvrant en mode d'écriture (avec "w" ou "w+" comme argument de fopen ). |

| Attention |
| --- |
| flock ne fonctionne pas sur NFS ou sur les autres systèmes de fichiers réseaux. Vérifiez la documentation de votre système d'exploitation pour plus de détails. |

Sur certains systèmes d'exploitation, flock est implémenté au niveau processus. Lorsque vous utilisez une API multi−thread comme ISAPI, vous risquez de ne pas pouvoir avoir confiance en flock pour protéger vos fichiers contre d'autres scripts PHP qui fonctionnent en paralelle sur d'autres thread du même serveur.

flock n'est pas supporté sur les vieux systèmes de fichiers comme FAT et ses dérivés, et elle retournera forcément FALSE sous ces environnement (ceci est particulièrement vrai pour les utilisateurs de Windows 98).

## 6.10.40 fnmatch() : Repère un fichier à partir d'un masque

array **fnmatch** ( string pattern , string string , *int* *flags* )

fnmatch vérifie si la chaîne string va passer le masque shell pattern .

C'est tout particulièrement pratique pour les noms de fichiers, et il peut être utilisé sur des chaînes de caractères. L'utilisateur moyen de shell peut être familier avec les masques shell, ou tout au moins, leur expression les plus simples, comme '?' et '*' . De cette façon, utiliser fnmatch au lieu de ereg ou preg_match pour des recherches peut être plus pratique pour les non−initiés.

Vérifier le nom d'une couleur avec un masque shell

```
<?php
 if(fnmatch("*gr[ae]y", $color)) {
   echo "some form of gray ...";
 }
?>
```

Voir aussi glob , ereg et preg_match .

## 6.10.41 fopen() : Ouverture d'un fichier ou d'une URL

resource **fopen** ( string filename , string mode , *int* *use_include_path* , *resource* *zcontext* )

fopen crée une ressource nommée, spécifiée par le paramètre filename , sous la forme d'un flux. Si filename est de la forme "protocole://...", filename est supposé être une URL, et PHP va rechercher un gestionnaire de protocole adapté pour lire ce fichier. Si aucun gestionnaire pour ce protocole n'est disponible, PHP va émettre une alerte qui vous permettra de savoir que vous avez des problèmes dans votre script, et il tentera d'exploiter filename comme un fichier classique.

Si PHP décide que le fichier filename est un fichier local, il va essayer d'ouvrir un flux avec ce fichier. Le fichier doit être accessible à PHP . Il vous faut donc vous assurer que vous avez les droits d'accès à ce fichier. Si vous activé le safe mode , ou la directive open_basedir , d'autres conditions peuvent aussi s'appliquer.

Si PHP a décidé que filename spécifie un protocole enregistré, et que ce protocole est enregistré comme un protocole réseau, PHP s'assurera que la directive allow_url_fopen est activée. Si elle est inactive, PHP va émettre une alerte, et l'ouverture va échouer.

Note

La liste des protocoles supportés est disponible Liste des protocoles supportés . Certains protocoles (appelé aussi wrappers ou gestionnaires) supportent des contexte et/ou des options dans le fichier php.ini . Référez vous aux pages du manuel traitant le protocole, pour connaître la liste des options qui sont disponibles. ( i.e. l'option de php.ini user_agent est utilisée par le gestionnaire http ) Pour une description des contexts et du paramètre zcontext , voyez ref.stream .

mode peut prendre les valeurs suivantes :

- 'r' − Ouvre en lecture seule, et place le pointeur de fichier au début du fichier.
- 'r+' − Ouvre en lecture et écriture, et place le pointeur de fichier au début du fichier.
- 'w' − Ouvre en écriture seule; place le pointeur de fichier au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, on tente de le créer.
- 'w+' − Ouvre en lecture et écriture; place le pointeur de fichier au début du fichier et réduit la taille du fichier à 0. Si le fichier n'existe pas, on tente de le créer.
- 'a' − Ouvre en écriture seule; place le pointeur de fichier à la fin du fichier file. Si le fichier n'existe pas, on tente de le créer.
- 'a+' − Ouvre en lecture et écriture; place le pointeur de fichier à la fin du fichier. Si le fichier n'existe pas, on tente de le créer.

De plus, mode peut contenir la lettre 'b'. Cette option n'est utile que sur les systèmes qui font la différence entre les fichiers binaires et les fichiers textes (en bref, c'est une fonctionnalité Windows, totalement inutile sous Unix). S'il n'est pas nécessaire, il sera ignoré.

Vous pouvez utiliser le troisième paramètre optionnel pour explorer le dossier include_path , en le mettant à 1.

Le paramètre optionnel zcontext est utilisé pour spécifier des paramètres d'optimisation et de callback.

Si l'ouverture échoue, fopen retournera FALSE .

Exemple avec fopen

```
 <?php
$handle = fopen ("/home/rasmus/file.txt", "r");
$handle = fopen ("/home/rasmus/file.gif", "wb");
$handle = fopen ("http://www.example.com/", "r");
$handle = fopen ("ftp://user:password@example.com/somefile.txt", "w");
?>
```

Si vous rencontrez des problèmes en lecture ou écriture de fichier et que vous utilisez PHP en version module de serveur, n'oubliez pas que les fichiers auxquels vous accédez ne sont pas nécessairement accessibles au processus serveur.

Sous Windows, assurez−vous de bien échapper les anti−slash utilisés dans le chemin du fichier, ou bien utilisez des slash.

Exemple avec fopen sous Windows

```
 <?php
  $handle = fopen ("c:\\data\\info.txt", "r");
?>
```

Voir aussi Liste des protocoles supportés , fclose , fgets , fsockopen , file , file_exists , is_readable , socket_set_timeout et popen .

## 6.10.42 fpassthru() : Affiche le reste du fichier

int **fpassthru** ( resource  handle )

fpassthru lit tout le reste d'un fichier jusqu'à la fin, et dirige le résultat vers la sortie standard.

Si une erreur survient, fpassthru retourne FALSE . Sinon, fpassthru retourne le nombre de caractères lu dans handle et passé à la sortie standard.

Le pointeur de fichier doit être valide, et doit avoir été correctement ouvert par fopen , popen , ou fsockopen . Après lecture, fpassthru va fermer le fichier (le pointeur handle sera alors invalide).

Note

Lorque vous utilisez la fonction fpassthru sur un fichier binaire sous Windows, assurez vous d'avoir ouvert le fichier en mode binaire en ajoutant la lettre b au mode d'accès utilisé dans fopen .

Vous êtes encouragés à utiliser l'option b lorsque vous traitez des fichiers binaires, même si votre système ne le requiert pas, de façon à rendre vos scripts portables.

Voir aussi readfile , fopen , popen et fsockopen .

## 6.10.43 fputs() : Ecrit dans un fichier

int **fputs** ( resource handle , string str , *int length* )

fputs est un alias de fwrite , et lui est identique en tous points. Notez que length est un paramètre optionnel, et s'il n'est pas spécifié, toute la chaîne est écrite.

## 6.10.44 fread() : Lecture du fichier en mode binaire

string **fread** ( resource handle , int length )

fread lit jusqu'à length octets dans le fichier referencé par handle . La lecture s'arrête lorsque length octets ont été lus, ou que l'on a atteint la fin du fichier, ou qu'une erreur survient (le premier des trois).

Exemple avec fread

```
<?php
// Lit un fichier, et le place dans une chaîne
$filename = "/usr/local/something.txt";
$handle = fopen ($filename, "r");
$contents = fread ($handle, filesize ($filename));
fclose ($handle);
?>
```

Note

Sur les systèmes qui différencient les fichiers textes et binaires (i.e. Windows) le fichier doit être ouvert avec la lettre 'b' ajoutée au paramètre de mode de la fonction fopen .

Note

Utilisation des fichiers binaires avec fread

```
<?php
 $filename = "c:\\fichiers\\uneimage.gif";
 $handle = fopen ($filename, "rb");
 $contents = fread ($handle, filesize ($filename));
 fclose ($handle);
?>
```

Voir aussi fwrite , fopen , fsockopen , popen , fgets , fgetss , file et fpassthru .

## 6.10.45 fscanf() : Analyse un fichier en fonction d'un format

mixed **fscanf** ( resource handle , string format , *string var1* )

fscanf est similaire à sscanf , mais elle prend comme entrée un fichier, associé à handle et l'interprète en fonction du format format . Si seulement deux paramètres sont passés à la fonction, les valeurs analysées seront retournées sous forme de tableau. Si des arguments optionnels sont passés, la fonction retournera le nombre de valeurs assignées. Les options doivent être passées par référence.

Tous les caractères blancs de la chaîne de formattage correspondent à autant d'espaces dans le flux d'entrée. Cela signifie qu'une tabulation \t dans la chaîne de format peut remplacer un espace simple dans le flux d'entrée.

Exemple avec fscanf

```
<?php
 $handle = fopen ("users.txt","r");
 while ($userinfo = fscanf ($handle, "%s\t%s\t%s\n")) {
   list ($name, $profession, $countrycode) = $userinfo;
   //... traitement des données
 }
 fclose($handle);
?>
```

utilisateurs.txt

```
 javier  argonaute        pe
hiroshi sculpteur        jp
robert  hockeyuer us
luigi   fleuriste it
```

Note

Avant PHP 4.3.0, le nombre maximum de caractères lus dans le fichier était de 512 (ou bien jusqu'à la première nouvelle ligne "\n" : en fait, le premier des deux). Depuis PHP 4.3.0, des lignes de n'importe quelle taille peuvent être lues.

Voir aussi fread , fgets , fgetss , sscanf , printf et sprintf .

## 6.10.46 fseek() : Modifie le pointeur de fichier

int **fseek** ( resource handle , int offset , *int whence* )

fseek modifie le curseur de position dans le fichier handle . La nouvelle position mesurée en octets à partir du début du fichier, est obtenue en additionnant la distance offset à la position whence . Ce paramètre peut prendre les valeurs suivantes :

- SEEK_SET − La position finale vaut offset octets.
- SEEK_CUR − La position finale vaut la position courante ajoutée à offset octets.
- SEEK_END − La position finale vaut la position courante par rapport à la fin du fichier, ajoutée de offset .

Si whence n'est pas spécifiée, il vaut par défaut SEEK_SET .

fseek retourne 0 en cas de succès, et sinon −1. Notez que positionner le pointeur au delà de la fin du fichier n'est pas une erreur.

fseek ne peut pas être utilisé sur les pointeurs retournés par fopen s'ils sont au format HTTP ou FTP.

Note

Le paramètre whence a été ajouté en PHP 4.0.0.

Voir aussi ftell et rewind .

## 6.10.47 fstat() : Lit les informations sur un fichier à partir d'un pointeur de fichier

array **fstat** ( resource handle )

fstat rassemble les informations sur le fichier dont on connaît le pointeur handle . fstat est similaire à la fonction stat , hormis le fait qu'elle utilise un pointeur de fichier, au lieu d'un nom de fichier.

fstat retourne un tableau avec les éléments suivants :

- volume
- inode
- nombre de liens
- id de l'utilisateur propriétaire
- id du groupe propriétaire
- type du volume de l'inode *
- taille en octets
- date du dernier accès
- date de la dernière modification
- date du dernier changement
- taille de bloc du système pour les entrées/sorties(*)
- nombre de blocs alloués

* − uniquement sur les systèmes qui supportent le type st_blksize. Les autres systèmes (i.e. Windows) retournent −1.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |
| Note |
| Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local. |

## 6.10.48 ftell() : Renvoie la position du pointeur du fichier

int **ftell** ( resource handle )

ftell retourne la position courante du pointeur dans le fichier repéré par le pointeur handle , i.e., son offset.

Si une erreur survient, retourne FALSE .

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par fopen ou popen .

Voir aussi fopen , popen , fseek et rewind .

## 6.10.49 ftruncate() : Tronque un fichier

bool **ftruncate** ( resource handle , int size )

ftruncate prend le pointeur de fichier handle et le tronque à la taille de size . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi fopen et fseek

Voir aussi fopen et fseek .

## 6.10.50 fwrite() : Ecriture du fichier en mode binaire

int **fwrite** ( resource handle , string string , *int length* )

fwrite écrit le contenu de la chaîne string dans le fichier pointé par handle . Si la longueur length est fournie, l'écriture s'arrêtera après length octets, ou à la fin de la chaîne (le premier des deux).

fwrite retourne le nombre de bytes écrites ou FALSE en cas d'erreur.

Notez que si length est fourni, alors l'option de configuration magic_quotes_runtime sera ignorée, et les slash seront conservés.

| Note |
| --- |
| Sur les systèmes qui font la différence entre les fichiers binaires et les fichiers textes (par exemple, Windows), le fichier doit être ouvert avec l'option 'b' inclus dans le paramètre de mode de fopen . |

Exemple d'écriture dans un fichier avec fwrite

```
<?php
$filename = 'test.txt';
$somecontent = "Ajout de chaîne dans le fichier \n";

// Assurons nous que le fichier est accessible en écriture
if (is_writable($filename)) {

    // Dans notre exemple, nous ouvrons le fichier $filename en mode d'ajout
    // Le pointeur de fichier est placé à la fin du fichier
    // c'est là que $somecontent sera placé
    if (!$handle = fopen($filename, 'a')) {
         print "Impossible d'ouvrir le fichier ($filename)";
         exit;
    }

    // Write $somecontent to our opened file.
    if (!fwrite($handle, $somecontent)) {
       print "Impossible d'écrire dans le fichier ($filename)";
       exit;
    }

    print "L'écriture de ($somecontent) dans le fichier ($filename) a réussi";

    fclose($handle);

} else {
    print "Le fichier $filename n'est pas accessible en écriture.";
}
?>
```

Voir aussi fread , fopen , fsockopen , popen et fputs .

## 6.10.51 glob() : Recheche des chemins qui vérifie un masque

array **glob** ( string pattern , *int flags* )

glob recherche tous les chemins qui vérifient le masque pattern , en suivant les règles du shell. Aucun remplacement de tilde ~ ou de paramètre n'est fait.

glob retourne un tableau contenant les chemins et fichiers trouvés, ou FALSE en cas d'erreur.

Un moyen pratique pour remplacer opendir par glob

```
<?php
 foreach (glob("*.txt") as $filename) {
```

```
    echo "$filename occupe " . filesize($filename) . " octets\n";
  }
?>
```

Le résultat du script ci−dessus pourrait être :

```
funclist.txt occupe 44686 octets
funcsummary.txt occupe 267625 octets
quickref.txt occupe 137820 octets
```

| Note |
| --- |
| Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local. |

Voir aussi opendir , readdir , closedir et fnmatch .

## 6.10.52 is_dir() : Indique si le nom de fichier est un dossier

bool **is_dir** ( string filename )

is_dir retourne TRUE si filename existe et est un dossier.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

Voir aussi chdir , opendir , is_file et is_link .

## 6.10.53 is_executable() : Indique si le fichier est exécutable

bool **is_executable** ( string filename )

is_executable retourne TRUE si filename existe et est exécutable.

is_executable est disponible avec Windows depuis PHP 5.0.0.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

Voir aussi is_file et is_link .

## 6.10.54 is_file() : Indique si le fichier est un véritable fichier.

bool **is_file** ( string filename )

is_file retourne TRUE si filename existe et est un fichier (et pas un dossier).

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

Voir aussi is_dir et is_link .

## 6.10.55 is_link() : Indique si le fichier est un lien symbolique.

bool **is_link** ( string filename )

is_link retourne TRUE si filename existe et est un lien symbolique.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

Voir aussi is_dir , is_file et readlink .

## 6.10.56 is_readable() : Indique un fichier est autorisé en lecture

bool **is_readable** ( string filename )

is_readable retourne TRUE si filename existe et est accessible en lecture.

N'oubliez pas que PHP accède aux fichiers avec les mêmes autorisations que l'utilisateur qui fait tourner le serveur web (souvent, c'est 'nobody', personne).

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

Voir aussi is_writable , file_exists et fgets .

## 6.10.57 is_uploaded_file() : Indique si le fichier a été téléchargé par HTTP POST

bool **is_uploaded_file** ( string filename )

is_uploaded_file est disponible à partir des versions PHP 3.0.16 et 4.0.2.

is_uploaded_file retourne TRUE si le fichier filename a été téléchargé par HTTP POST. Cela est très utile pour vous assurer qu'un utilisateur n'essaie pas d'accéder intentionnellement à un fichier auquel il n'a pas droit (comme /etc/passwd ).

Ce type de vérification est spécialement important s'il est possible que les fichiers téléchargés révélent leur contenu à l'utilisateur, ou même aux utilisateurs du même système.

is_uploaded_file est disponible dans les versions de PHP 3 après PHP 3.0.16, et dans les versions de PHP 4 après la 4.0.2. Si vous êtes coincés avec une version plus ancienne, vous pouvez utiliser l'une des fonctions suivantes pour vous protéger :

| Note |
| --- |
| L'exemple suivant **ne fonctionne pas** avec les versions de PHP ultérieures à la 4.0.2. Elle dépend de fonctionnalités internes de PHP qui ont disparu après cette version. |

```
<?php
/* Test utilisateur du fichier téléchargé. */
function is_uploaded_file($filename) {
    if (!$tmp_file = get_cfg_var('upload_tmp_dir')) {
        $tmp_file = dirname(tempnam('', ''));
    }
    $tmp_file .= '/' . basename($filename);
    /* L'utilisateur peut avoir laissé des / finaux dans php.ini... */
    return (ereg_replace('/+', '/', $tmp_file) == $filename);
}

/* Voici comment l'utiliser, car move_uploaded_file() n'est pas disponible
   dans les anciennes versions : */
if (is_uploaded_file($HTTP_POST_FILES['userfile'])) {
    copy($HTTP_POST_FILES['userfile'], "/dossier/ou/placer/un/fichier/uploadé");
} else {
    echo "Tentative d'attaque possible : '$HTTP_POST_FILES[userfile]'.";
}
?>
```

Voir aussi move_uploaded_file .

## 6.10.58 is_writable() : Indique si un fichier est autorisé en écriture

bool **is_writable** ( string filename )

is_writable retourne TRUE si filename existe et est accessible en écriture.

N'oubliez pas que PHP accède aux fichiers avec les mêmes autorisations que l'utilisateur qui fait tourner le serveur web (souvent, c'est 'nobody', personne).

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

Voir aussi is_readable , file_exists et fwrite .

## 6.10.59 is_writeable() : Indique si un fichier est autorisé en écriture

bool **is_writeable** ( string filename )

is_writeable est un alias de is_writable .

## 6.10.60 link() : Crée un lien

bool **link** ( string target , string link )

link crée un lien en le fichier source link et le fichier destination target .

| Note |
| --- |
| Cette fonction ne fonctionne pas avec les fichiers distants , car le fichier utilisé doit être accessible sur le système de fichiers local. |

Voir aussi symlink , readlink et linkinfo .

## 6.10.61 linkinfo() : Renvoie les informations à propos d'un lien

int **linkinfo** ( string path )

linkinfo renvoie les informations à propos d'un lien, c'est−à−dire le champ st_dev de la structure d'information UNIX (comme en langage C). linkinfo sert à vérifier si un lien (repéré par path ) existe (en utilisant la même méthode que la macro S_ISLNK de stat.h). linkinfo retourne FALSE en cas d'erreur.

Voir aussi symlink , link et readlink .

## 6.10.62 lstat() : Retourne les informations à propos d'un fichier ou d'un lien symbolique.

array **lstat** ( string filename )

lstat renvoie les informations à propos du fichier filename . Cette fonction est identique à la fonction stat à part que si filename est un lien symbolique, les informations seront alors basées sur le lien symbolique.

Voyez la page de manuel de stat pour plus d'informations sur la structure du tableau retourné par lstat .

Array

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Voir aussi stat .

## 6.10.63 mkdir() : Crée un dossier

bool **mkdir** ( string pathname , *int mode* )

mkdir tente de créer un dossier dans le chemin pathname .

Notez que vous aurez à préciser le mode en base octale, ce qui signifie que vous aurez probablement un 0 comme premier chiffre. Le mode sera aussi modifié par le umask courant, que vous pouvez modifier avec la fonction umask .

| Note |
| --- |
| Le paramètre mode est ignoré sous Windows, et il est devenu optionnel depuis PHP 4.2.0. |

Le mode par défaut est le mode 0777, ce qui correspond au maximum de droits possible. Pour plus d'informations sur les modes, lisez en détail la documentation de la fonction chmod .

Exemple avec mkdir

```
<?php
 mkdir ("/chemin/de/mon/dossier", 0700);
?>
```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi rmdir .

## 6.10.64 move_uploaded_file() : Déplace un fichier téléchargé

bool **move_uploaded_file** ( string filename , string destination )

move_uploaded_file s'assure que le fichier filename est un fichier téléchargé par HTTP POST. Si le fichier est valide, il est déplacé jusqu'à destination .

Si filename n'est pas valide, rien ne se passe, et move_uploaded_file retournera FALSE .

Si filename est un fichier téléchargé, mais que pour une raison quelconque, il ne peut être déplacé, rien ne se passe, et move_uploaded_file retourne FALSE . De plus, une alerte sera affichée.

Ce type de vérification est spécialement important s'il est possible que les fichiers téléchargés révèlent leur contenu à l'utilisateur, ou même aux utilisateurs du même système.

| Note |
| --- |
| Lorsque le safe−mode est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |

| Note |
| --- |
| move_uploaded_file n'est pas affectée par les restrictictions du safe mode . Ce n'est pas un problème, car move_uploaded_file ne fonctionne que sur des fichiers qui ont été téléchargés avec PHP . |

| Attention |
| --- |
| Si le fichier de destination existe déjà, il sera écrasé. |

Voir aussi is_uploaded_file .

## 6.10.65 parse_ini_file() : Traite un fichier de configuration

array **parse_ini_file** ( string filename , *bool process_sections* )

parse_ini_file charge le fichier filename et retourne les configuration qui s'y trouve sous forme d'un tableau associatif. En passant le deuxième paramètre optionnel à process_sections , vous obtiendrez un tableau multi−dimensionnel, avec les noms des sections. La valeur par défaut de ce paramètre est FALSE

| Note |
| --- |
| Cette fonction n'a rien a voir avec le fichier php.ini . Ce dernier a déjà était traité lorsque vous commencez a exécuter votre script. Cette fonction peut vous permettre de lire vos propres fichiers de configuration. |

La structure des fichiers de configuration lus est similaire à celle de php.ini .

| Note |
| --- |
| Si une valeur du fichier .ini contient des données non alphanumérique, il faut la protéger en la plaçant entre guillemets doubles ("). |

| Note |
| --- |
| Depuis PHP 4.2.1 cette fonction est aussi affectée par le safe mode et open_basedir . |

La structure du fichier .ini est similaire à celle du fichier php.ini .

Les constantes peuvent aussi être utilisées dans le fichier .ini, ce qui fait que si vous définissez une constante avant d'exécuter parse_ini_file , elle sera intégrée dans les résultats. Seules les valeurs de configuration sont remplacées par leur équivalent en constantes. Par exemple :

Contenu de exemple.ini

```
 ; Ceci est un fichier de configuration
; Les commentaires commencent par ';', comme dans php.ini

[premiere_section]
un = 1
cinq = 5
animal = oiseau

[seconde_section]
chemin = /usr/local/bin
URL = "http://www.exemple.com/~utilisateur"
```

Exemple avec parse_ini_file

```
 <?php

define ('oiseau', 'dodu dodo');

// Analyse sans les sections
$ini_array = parse_ini_file("exemple.ini");
print_r($ini_array);

// Analyse avec les sections
$ini_array = parse_ini_file("exemple.ini", TRUE);
print_r($ini_array);

?>
```

Cet exemple va produire :

```
 Array
(
    [un] => 1
    [cinq] => 5
    [animal] => dodu dodo
    [chemin] => /usr/local/bin
    [URL] => http://www.example.com/~username
)
Array
(
    [premiere_section] => Array
        (
            [un] => 1
            [cinq] => 5
            [animal] => dodu dodo
        )

    [seconde_section] => Array
        (
            [chemin] => /usr/local/bin
            [URL] => http://www.exemple.com/~utilisateur
        )

)
```

## 6.10.66 pathinfo() : Retourne des informations sur un chemin système

array **pathinfo** ( string path )

pathinfo retourne un tableau associatif, contenant les informations sur le chemin path . Les éléments suivants sont retournés : dirname , basename et extension .

Exemple avec pathinfo

```
 <?php

$path_parts = pathinfo("/www/htdocs/index.html");

echo $path_parts["dirname"] . "\n";
echo $path_parts["basename"] . "\n";
echo $path_parts["extension"] . "\n";

?>
```

Va afficher :

```
 /www/htdocs
index.html
html
```

Note

Pour plus d'informations sur la lecture du chemin courant, lisez la section sur les variables prédéfinies .

Voir aussi dirname , basename , parse_url et realpath .

## 6.10.67 pclose() : Ferme un processus de pointeur de fichier

int **pclose** ( resource handle )

pclose ferme un processus de pointeur de fichier ouvert par popen .

Le pointeur de fichier doit être valide, et avoir été ouvert correctement par popen .

pclose retourne le statut final du processus exécuté.

Voir aussi popen .

## 6.10.68 popen() : Crée un processus de pointeur de fichier

resource **popen** ( string command , string mode )

popen ouvre un processus fils en faisant un fork de la commande.

popen retourne un pointeur de fichier identique à celui retourné par fopen , hormis le fait qu'il sera unidirectionnel (lecture seule, ou écriture seule), et doit être terminé par pclose . Ce pointeur peut être utilisé avec fgets , fgetss et fputs .

Si une erreur survient, retourne FALSE .

Exemple avec popen

```
 <?php
  $handle = popen ("/bin/ls", "r");
?>
```

Note

Si la commande à exécuter n'a pu être trouvée, une ressource valide sera retournée. Cela semble étrange, mais c'est pratique : cela vous permet d'accéder aux messages d'erreur qui ont été retournés par le shell :

Lire des messages d'erreurs avec popen

```
 <?php
  error_reporting(E_ALL);

/* Ajoute une redirection pour que vous puissions lire stderr */
  $handle = popen('/chemin/vers/spooge 2>&1', 'r');
  echo "'$handle'; " . gettype($handle) . "\n";
  $read = fread($handle, 2096);
  echo $read;
  pclose($handle);
?>
```

Voir aussi pclose , fopen et proc_open .

## 6.10.69 readfile() : Affiche un fichier

int **readfile** ( string filename , *bool use_include_path* , *resource context* )

readfile lit le fichier filename et l'envoie à la sortie standard.

readfile retourne le nombre d'octets lus depuis le fichier. Si une erreur survient, retourne FALSE , et à moins que la fonction n'ait été appelée avec @readfile, un message d'erreur est affiché.

Array

Vous pouvez utiliser le deuxième paramètre optionnel pour explorer le dossier include_path , en passant la valeur de 1.

Voir aussi fpassthru , file , fopen , include , require et virtual .

## 6.10.70 readlink() : Renvoie le nom du fichier vers lequel pointe un lien symbolique

string **readlink** ( string path )

readlink fait la même chose que la fonction readlink en C : elle retourne le contenu du lien symbolique repéré par path , ou FALSE en cas d'erreur.

Voir aussi is_link , symlink et linkinfo .

## 6.10.71 realpath() : Retourne le chemin canonique absolu

string **realpath** ( string path )

realpath résoud tous les liens symboliques, et remplace toutes les références '/./', '/../' et '/' de path puis retourne le chemin canonique absolu ainsi trouvé. Le résultat ne contient aucun lien symbolique, '/./' ou '/../'.

Exemple avec realpath

```
<?php
 $real_path = realpath ("../../index.php");
?>
```

Voir aussi basename , dirname et pathinfo .

## 6.10.72 rename() : Renomme un fichier

bool **rename** ( string oldname , string newname )

rename tente de renommer le fichier oldname en newname .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec rename

```
<?php
 rename("/tmp/tmp_file.txt", "/home/user/login/docs/mon_file.txt");
?>
```

Voir aussi copy , unlink et move_uploaded_file

Voir aussi copy , unlink et move_uploaded_file .

## 6.10.73 rewind() : Replace le pointeur de fichier au début

bool **rewind** ( resource handle )

rewind replace le pointeur du fichier handle au début. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le pointeur de fichier doit être valide, et avoir été correctement ouvert par fopen .

Note

Si vous avez ouvert le fichier en mode d'ajout ("a"), toutes les données que vous écrirez dans ce fichier seront toujours ajoutées, indépendemment de la position du pointeur de fichier.

Voir aussi fseek et ftell .

## 6.10.74 rmdir() : Efface un dossier

bool **rmdir** ( string dirname )

rmdir tente d'effacer le dossier dont le chemin est dirname . Le dossier doit être vide, et le script doit avoir les autorisations adéquates.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi mkdir et unlink .

## 6.10.75 set_file_buffer() : Alias de stream_set_write_buffer

int **set_file_buffer** ( int handle , int buffer )

set_file_buffer est un alias de stream_set_write_buffer .

## 6.10.76 stat() : Renvoie les informations à propos d'un fichier

array **stat** ( string filename )

stat renvoie les informations à propos du fichier filename . Si filename est un lien symbolique, les informations proviennent du fichier lui−même, et non du lien symbolique. lstat est identique à stat sauf que les informations seront alors basées sur le lien symbolique.

En cas d'erreurs, stat retourne FALSE . Elle générera aussi un avertissement.

Retourne un tableau avec des informations sur le fichier avec les éléments suivants. Ce tableau commence à zéro. En plus de retourner ces attributs dans un tableau indexé, ils peuvent être accédés par leurs clefs, comme mentionné à côté de chaque paramètre; ceci est disponible depuis PHP 4.0.6 :

1. 0 : volume (ou encore dev )
2. 1 : inode (ou encore ino )
3. 2 : droits d'accès au fichier (mode de protection du inode). A convertir en octal. Voir aussi fileperms . (ou encore mode )
4. 3 : nombre de liens (ou encore nlink )
5. 4 : id de l'utilisateur propriétaire (ou encore uid )
6. 5 : id du groupe propriétaire (ou encore gid )
7. 6 : type du volume de l'inode * (ou encore rdev )
8. 7 : taille en octets (ou encore size )
9. 8 : date du dernier accès (ou encore atime )
10. 9 : date de la dernière modification (ou encore mtime )
11. 10 : date du dernier changement (ou encore ctime )
12. 11 : taille de bloc du système pour les entrées/sorties * (ou encore blksize )
13. 12 : nombre de blocs alloués (ou encore blocks )

* − uniquement sur les systèmes qui supportent le type st_blksize. Les autres systèmes (i.e. Windows) retournent −1.

| Note |
| --- |
| Les résultats de cette fonction sont mis en cache. Voyez la fonction clearstatcache plus de détails. |

Array

Voir aussi lstat , fstat , filemtime et filegroup .

## 6.10.77 symlink() : Crée un lien symbolique

bool **symlink** ( string target , string link )

symlink crée un lien symbolique pour l'objet target avec le nom de link . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi link , readlink et linkinfo .

## 6.10.78 tempnam() : Crée un fichier avec un nom unique

string **tempnam** ( string dir , string prefix )

tempnam crée un fichier temporaire unique dans le dossier dir . Si le dossier n'existe pas, tempnam va générer un nom de fichier dans le dossier temporaire du système.

Avant PHP 4.0.6, le comportement de tempnam dépendait de l'OS sous−jacent. Sous Windows, la variable d'environnement TMP remplace le paramètre dir ; sous Linux, la variable d'environnement TMPDIR a la priorité, tandis que pour les OS en système V R4, le paramètre dir sera toujours utilisé, si le dossier qu'il représente existe. Consultez votre documentation pour plus de détails.

tempnam retourne le nom du fichier temporaire, ou la chaîne NULL en cas d'échec.

Exemple avec tempnam

```
 $tmpfname = tempnam ("/tmp", "FOO");

$handle = fopen($tmpfname, "w");
fwrite($handle, "Ecrire dans le fichier temporaire");
fclose($handle);

// faire quelquechose ici

unlink($tmpfname);
```

Note

Le comportement de tempnam a été modifié en PHP 4.0.3. Le fichier temporaire est aussi créé pour éviter des conflits si le fichier apparait dans le système de fichiers entre le moment où la chaîne est générée et avant que le script n'ai terminé la création du fichier. Notez que vous devez effacer le fichier si vous n'en avez plus besoin, car cela n'est pas fait automatiquement.

Voir aussi tmpfile et unlink .

## 6.10.79 tmpfile() : Crée un fichier temporaire

resource **tmpfile** ( void )

tmpfile crée un fichier temporaire avec un nom unique, ouvert en écriture, et retourne un pointeur de fichier, identique à ceux retournés par fopen . Ce fichier sera automatiquement effacé lorsqu'il sera fermé (avec fclose ), ou lorsque le script sera terminé.

Pour plus de détails, consultez votre documentation système sur la fonction tmpfile(3) , et sur stdio.h .

Exemple avec tmpfile

```
 <?php
  $temp = tmpfile();
  fwrite($temp, "Ecriture dans le fichier temporaire");
  fclose($temp); // Ceci supprime le fichier temporaire
?>
```

Voir aussi tempnam .

## 6.10.80 touch() : Modifie la date de modification et de dernièr accès d'un fichier

bool **touch** ( string filename , *int time* , *int atime* )

touch tente de forcer la date de modification du fichier nommé filename à la date de time . Si time est omis, c'est l'heure courante qui est utilisée.

Si le fichier n'existe pas, PHP tente de le créer.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec touch

```
 <?php
if ( touch($NomDeFichier) ) {
  print "La date de modification de $NomDeFichier a
          été fixée à maintenant";
} else {
  print "Désolé, il est impossible de changer
          la date de modification de $NomDeFichier";
}
?>
```

## 6.10.81 umask() : Change le "umask" courant

int **umask** ( *int* *mask* )

umask change le umask de PHP : mask & 0777 et retourne le vieux umask. Lorsque PHP est utilisé comme module de serveur, le umask reprend sa valeur à la fin de chaque script.

umask appelé sans arguments, retourne simplement le umask courant.

## 6.10.82 unlink() : Efface un fichier

bool **unlink** ( string filename , *resource* *context* )

unlink efface filename . Identique à la fonction Unix C unlink(). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Depuis PHP 5.0.0 , unlink peut aussi être utilisée avec **quelques** prôtocoles d'url. Voyez Liste des protocoles supportés pour une liste des prôtocoles supportant unlink . |

| Note |
| --- |
| La paramètre context a été ajouté en PHP 5.0.0 . |

Voir aussi rmdir pour la suppression de dossiers.

# 6.11 FTP

## 6.11.1 Introduction

FTP : File Transfer Protocol (Protocole de transfert de fichiers). Ces fonctions implémentent un client pour accéder aux serveurs FTP, comme défini dans http://www.faqs.org/rfcs/rfc959.html .

## 6.11.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.11.3 Installation

Pour activer le module FTP de votre configuration PHP, il faut utiliser l'option −−enable−ftp en PHP 4, et l'option −−with−ftp en PHP 3 avec le script de configuration.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

## 6.11.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.11.5 Types de ressources

This extension uses one resource type, which is the link identifier of the FTP connection.

## 6.11.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*FTP_ASCII (entier)*
*FTP_TEXT (entier)*
*FTP_BINARY (entier)*
*FTP_IMAGE (entier)*
*FTP_TIMEOUT_SEC (entier)*
Voir ftp_set_option pour les détails.

Les constantes suivantes ont été introduites en PHP 4.3.0.

*FTP_AUTOSEEK (entier)*
Voir ftp_set_option pour les détails.
*FTP_AUTORESUME (entier)*
Déterimine automatiquement la position de reprise (RESUME) et la position de début pour les requêtes GET et PUT (ne fonctionne qu'avec FTP_AUTOSEEK)

*FTP_FAILED (entier)*
Le mode asynchrone a échoué
*FTP_FINISHED (entier)*
Le mode asynchrone a terminé
*FTP_MOREDATA (entier)*
Le mode asynchrone est encore actif

## 6.11.7 Exemples

Exemple FTP

```
 <?php
// création de la connexion
$conn_id = ftp_connect($ftp_server);

 // authentification avec nom de compte et mot de passe
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

 // vérification de la connexion
if ((!$conn_id) || (!$login_result)) {
         echo "La connexion FTP a échoué!";
        echo "Tentative de connexion à $ftp_server avec $ftp_user_name";
        die;
     } else {
        echo "Connected to $ftp_server, for user $ftp_user_name";
    }

// téléchargement d'un fichier
$upload = ftp_put($conn_id, $destination_file, $source_file, FTP_BINARY);

 // Vérification de téléchargement
if (!$upload) {
         echo "Le téléchargement FTP a échoué!";
    } else {
        echo "Téléchargement de $source_file sur $ftp_server en $destination_file";
    }

// fermeture de la connexion FTP.
ftp_close($conn_id);
 ?>
```


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- ftp_alloc
- ftp_cdup
- ftp_chdir
- ftp_chmod
- ftp_close
- ftp_connect
- ftp_delete
- ftp_exec
- ftp_fget
- ftp_fput
- ftp_get_option
- ftp_get
- ftp_login
- ftp_mdtm
- ftp_mkdir
- ftp_nb_continue
- ftp_nb_fget
- ftp_nb_fput

- ftp_nb_get
- ftp_nb_put
- ftp_nlist
- ftp_pasv
- ftp_put
- ftp_pwd
- ftp_quit
- ftp_raw
- ftp_rawlist
- ftp_rename
- ftp_rmdir
- ftp_set_option
- ftp_site
- ftp_size
- ftp_ssl_connect
- ftp_systype

## 6.11.9 ftp_cdup() : Change de dossier et passe au dossier parent

bool **ftp_cdup** ( resource ftp_stream )

ftp_cdup change de dossier et passe au dossier parent.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ftp_chdir .

## 6.11.10 ftp_chdir() : Change le dossier FTP courant

bool **ftp_chdir** ( resource ftp_stream , string directory )

ftp_chdir change le dossier courant en directory .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec ftp_chdir

```php
<?php

// Connexion
$conn_id = ftp_connect($ftp_server);

 // Identification par login et mot de passe
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

 // Vérification de la connexion
if ((!$conn_id) || (!$login_result)) {
    die("La connexion FTP a échoué!");
}

echo "Dossier courant : ", ftp_pwd($conn_id), "\n";

if (@ftp_chdir($conn_id, "somedir")) {
    echo "Le dossier courant est maintenant : ", ftp_pwd($conn_id), "\n";
} else {
     echo "Impossible de changer le dossier courant.\n";
}
?>
```

Voir aussi ftp_cdup .

## 6.11.11 ftp_chmod() : Modifie les droits d'un fichier via FTP

string **ftp_chmod** ( resource ftp_stream , int mode , string filename )

ftp_chmod modifie les droits d'accès au fichier filename sur le serveur FTP ftp_stream , en leur attribuant les droits de mode , spécifié sous la forme d'un entier en base octale.

ftp_chmod retourne les permissions du fichier en cas de succès et FALSE sinon.

Voir aussi chmod .

## 6.11.12 ftp_close() : Ferme une connexion FTP

void **ftp_close** ( resource ftp_stream )

| Note |
| --- |
| ftp_close n'est disponible que dans les sources du CVS. |

ftp_close ferme la connexion ftp_stream et libère les ressources. Après avoir appelé cette fonction, vous ne pouvez plus utiliser l'ancienne connexion, et vous devez en ouvrir une nouvelle avec ftp_connect .

## 6.11.13 ftp_connect() : Ouvre une connexion FTP

resource **ftp_connect** ( string host , *int port* , *int timeout* )

ftp_connect retourne un flux FTP en cas de succès, et FALSE sinon.

ftp_connect ouvre une connexion FTP avec l'hôte host . Le paramètre port spécifie le port de connexion. S'il est omis, le port 21 sera utilisé.

Le paramètre timeout spécifie la durée d'expiration des opérations réseaux ultérieures. Si il est omis, il vaut par défaut 90 secondes. Ce délai d'expiration peut être modifiée et lu à tout moment avec les fonctions ftp_set_option et ftp_get_option .

| Note |
| --- |
| Le paramètre timeout a été ajouté en PHP 4.2.0. |

Exemple avec ftp_connect

```
<?php

$ftp_server = "ftp.exemple.com";

// ouvre une connexion ou échoue
$conn_id = ftp_connect($ftp_server) or die("Impossible de se connecter à $ftp_server");

?>
```

Voir aussi ftp_close et ftp_ssl_connect .

## 6.11.14 ftp_delete() : Efface un fichier sur un serveur FTP

bool **ftp_delete** ( resource ftp_stream , string path )

ftp_delete efface le fichier path sur un serveur FTP.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.11.15 ftp_exec() : Exécute une commande sur le serveur FTP

bool **ftp_exec** ( resource ftp_stream , string command )

ftp_exec envoie une commande SITE EXEC au serveur FTP, pour qu'il exécute le programme command . ftp_exec retourne le résultat de la commande si elle a réussi, et FALSE sinon.

Voir aussi ftp_raw .

## 6.11.16 ftp_fget() : Télécharge un fichier via FTP dans un flux local

bool **ftp_fget** ( resource ftp_stream , resource handle , string remote_file , int mode , *int resumepos* )

ftp_fget télécharge le fichier remote_file depuis le serveur FTP, et l'écrit dans le fichier identifié par handle . Le mode de transfert mode spécifié doit être FTP_ASCII ou FTP_BINARY .

| Note |
| --- |
| La paramètre resumepos a été ajouté en PHP 4.3.0. |

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ftp_get , ftp_nb_get et ftp_nb_fget .

## 6.11.17 ftp_fput() : Charge un fichier sur un serveur FTP

bool **ftp_fput** ( resource ftp_stream , string remote_file , resource handle , int mode , *int startpos* )

ftp_fput charge les données issues du fichier identifié par handle jusqu'à la fin du fichier. Le résultat est stocké dans le fichier remote_file sur le serveur FTP. Le mode de transfert mode spécifié doit être FTP_ASCII ou FTP_BINARY .

| Note |
| --- |
| Le paramètre startpos a été ajouté en PHP 4.3.0. |

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ftp_put , ftp_nb_fput et ftp_nb_put .

## 6.11.18 ftp_get_option() : Lit différenctes options pour la connexion FTP courante

mixed **ftp_get_option** ( resource ftp_stream , int option )

| Note |
| --- |
| ftp_get_option n'est disponible que dans le CVS. |

ftp_get_option retourne la valeur de l'option option en cas de succès, et FALSE si l'option option n'est pas supportée. Dans ce dernier cas, un message d'erreur est alors affiché.

ftp_get_option retourne la valeur de l'option option pour la connexion ftp_stream . Actuellement, les options suivantes sont supportées :

| | |
| --- | --- |
| FTP_TIMEOUT_SEC | Retourne le délai d'expiration pour les opérations liées au réseau. |

Exemple avec ftp_get_option

```
<?php
 // Lit le délai d'expiration pour la connexion en cours
 $timeout = ftp_get_option($conn_id, FTP_TIMEOUT_SEC);
?>
```

Voir aussi ftp_set_option .

## 6.11.19 ftp_get() : Télécharge un fichier depuis un serveur FTP

bool **ftp_get** ( resource ftp_stream , string local_file , string remote_file , int mode , *int resumepos* )

ftp_get télécharge le fichier remote_file depuis le serveur FTP, et le sauve dans le fichier local local_file . Le mode de transfert mode spécifié doit être soit FTP_ASCII ou FTP_BINARY .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ftp_fget , ftp_nb_get et ftp_nb_fget .

## 6.11.20 ftp_login() : Identification avec une connexion FTP

bool **ftp_login** ( resource ftp_stream , string username , string password )

ftp_login authentifie le flux FTP.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec ftp_login

```
<?php

$ftp_server = "ftp.exemple.com";
$ftp_user = "foo";
$ftp_pass = "bar";

// Connexion
$conn_id = ftp_connect($ftp_server) or die("Couldn't connect to $ftp_server");

 // Tentative de login
if (@ftp_login($conn_id, $ftp_user, $ftp_pass)) {
    echo "Connecté avec $ftp_user@$ftp_server\n";
} else {
    echo "Impossible de se connecter avec $ftp_user\n";
}

?>
```

## 6.11.21 ftp_mdtm() : Retourne la date de dernière modification d'un fichier sur un serveur FTP.

int **ftp_mdtm** ( resource ftp_stream , string remote_file )

ftp_mdtm retourne un UNIX timestamp en cas de succès, et FALSE sinon.

ftp_mdtm lit la date de dernière modification d'un fichier et retourne le UNIX timestamp. Si une erreur survient, ou si le fichier n'existe pas,la valeur −1 est retournée. Certains serveurs FTP ne supportent pas cette fonction.

## 6.11.22 ftp_mkdir() : Crée un dossier sur un serveur FTP

string **ftp_mkdir** ( resource ftp_stream , string directory )

ftp_mkdir retourne le nom du dossier ainsi créé en cas de succès, et FALSE sinon.

ftp_mkdir crée le dossier nommé directory .

Voir aussi ftp_rmdir .

## 6.11.23 ftp_nb_continue() : Reprend le téléchargement d'un fichier (non bloquant)

bool **ftp_nb_continue** ( resource ftp_stream )

ftp_nb_continue reprend le téléchargement d'un fichier, de manière asynchrone.

ftp_nb_continue retourne FTP_FAILED ou FTP_FINISHED , ou encore FTP_MOREDATA .

## 6.11.24 ftp_nb_fget() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)

bool **ftp_nb_fget** ( resource ftp_stream , resource handle , string remote_file , int mode , *int resumepos* )

ftp_nb_fget lit le fichier remote_file présent sur le serveur FTP ftp_stream , puis écrit ce fichier dans le pointeur de fichier handle . Le mode de transfert mode spécifié peut être FTP_ASCII ou FTP_BINARY . La différence entre cette fonction et ftp_fget est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ftp_nb_get , ftp_nb_continue , ftp_fget et ftp_get .

## 6.11.25 ftp_nb_fput() : Ecrit un fichier sur un serveur FTP, et le lit depuis un fichier (non bloquant)

bool **ftp_nb_fput** ( resource ftp_stream , string remote_file , resource handle , int mode , *int startpos* )

ftp_nb_fput écrit le fichier remote_file présent sur la machine locale, sur le serveur FTP ftp_stream , puis lui donne le nom de handle . Le mode de transfert mode spécifié peut être FTP_ASCII ou FTP_BINARY . La différence entre cette fonction et ftp_fput est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ftp_nb_put , ftp_nb_continue , ftp_put et ftp_fput .

## 6.11.26 ftp_nb_get() : Lit un fichier sur un serveur FTP, et l'écrit dans un fichier (non bloquant)

bool **ftp_nb_get** ( resource ftp_stream , string local_file , string remote_file , int mode , *int resumepos* )

ftp_nb_get lit le fichier remote_file présent sur le serveur FTP ftp_stream , puis écrit ce fichier dans le pointeur de fichier handle . Le mode de transfert mode spécifié peut être FTP_ASCII ou FTP_BINARY . La différence entre cette fonction et ftp_fget est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec ftp_nb_get

```
 // Initialisation
$ret = ftp_nb_get($my_connection, "test", "README", FTP_BINARY);
while ($ret == FTP_MOREDATA) {

   // Faites ce que vous voulez
   echo ".";

   // Continue le chargement
   $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
   echo "Il y a eu une erreur lors du chargement du fichier...";
   exit(1);
}
```

Reprise d'un chargement avec ftp_nb_get

```
 // Initialisation
$ret = ftp_nb_get ($my_connection, "test", "README", FTP_BINARY,
                   filesize("test"));
// Ou : $ret = ftp_nb_get ($my_connection, "test", "README",
 //                         FTP_BINARY, FTP_AUTORESUME);
while ($ret == FTP_MOREDATA) {

   // Faites ce que vous voulez
   echo ".";

   // Continue le chargement
   $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
   echo "Il y a eu une erreur lors du chargement du fichier...";
   exit(1);
}
```

```
Reprise du téléchargement à la position 100 avec ftp_nb_get
 // Désactive l'option de Autoseek
ftp_set_option ($my_connection, FTP_AUTOSEEK, FALSE);

// Initialisation
$ret = ftp_nb_get ($my_connection, "newfile", "README", FTP_BINARY, 100);
while ($ret == FTP_MOREDATA) {

   ...

   // Continue le chargement
   $ret = ftp_nb_continue ($my_connection);
}
```

Dans l'exemple ci−dessus, "newfile" est 100 octets plus petit que "README" sur le site FTP, car nous avons commencé à lire à l'offset 100. Si nous n'avions pas désactivé l'option FTP_AUTOSEEK , les premiers 100 octets du fichier seraient complétés avec '\0' .

Voir aussi ftp_nb_fget , ftp_nb_continue , ftp_get et ftp_fget .

## 6.11.27 ftp_nb_put() : Envoie un fichier sur le serveur FTP (non−bloquant)

bool **ftp_nb_put** ( resource ftp_stream , string remote_file , string local_file , int mode , *int startpos* )

ftp_nb_put écrit le fichier remote_file présent sur la machine locale, sur le serveur FTP ftp_stream , puis lui donne le nom de handle . Le mode de transfert mode spécifié peut être FTP_ASCII ou FTP_BINARY . La différence entre cette fonction et ftp_fput est que cette fonction peut lire le fichier de manière asynchrone, de manière à ce que votre programme fasse autre chose, pendant que le fichier est téléchargé.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

```
Exemple avec ftp_nb_put
 // Initialisation
$ret = ftp_nb_put($my_connection, "test.remote", "test.local", FTP_BINARY);
while ($ret == FTP_MOREDATA) {

   // Faites ce que vous voulez
   echo ".";

   // Continue le téléchargement...
   $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
   echo "Il y a eu une erreur lors du chargement...";
   exit(1);
}
```

```
Resuming an upload with ftp_nb_put
 // Initialisation
$ret = ftp_nb_put ($my_connection, "test.remote", "test.local",
                      FTP_BINARY, ftp_size("test.remote"));
// ou bien : $ret = ftp_nb_put ($my_connection, "test.remote", "test.local",
 //                           FTP_BINARY, FTP_AUTORESUME);

while ($ret == FTP_MOREDATA) {

   // Faites ce que vous voulez
   echo ".";

   // Continue le téléchargement...
   $ret = ftp_nb_continue ($my_connection);
}
if ($ret != FTP_FINISHED) {
   echo "Il y a eu une erreur lors du chargement...";
   exit(1);
}
```

Voir aussi ftp_nb_fput , ftp_nb_continue , ftp_put et ftp_fput .

## 6.11.28 ftp_nlist() : Retourne la liste des fichiers dans un dossier

array **ftp_nlist** ( resource ftp_stream , string directory )

ftp_nlist retourne un tableau contenant les noms des fichiers en cas de succès, et FALSE sinon.

Exemple avec ftp_nlist

```
<?php

// Connexion
$conn_id = ftp_connect($ftp_server);
$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);

// Vérification de la connexion
if ((!$conn_id) || (!$login_result)) {
    die("La connexion a échoué !");
}

// Lire le contenu du dossier root
$contents = ftp_nlist($conn_id, "/");

// Afficher les entrées
foreach ($contents as $entry) {
    echo $entry, "<br />\n";
}

?>
```

Voir aussi ftp_rawlist .

## 6.11.29 ftp_pasv() : Active ou désactive le mode passif

bool **ftp_pasv** ( resource ftp_stream , bool pasv )

ftp_pasv active le mode passif si pasv est à TRUE (et le désactive si pasv est à FALSE ). En mode passif, les données de connexion sont initiées par le client, plutôt que par le serveur.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.11.30 ftp_put() : Charge un fichier sur un serveur FTP

bool **ftp_put** ( resource ftp_stream , string remote_file , string local_file , int mode , *int startpos* )

ftp_put enregistre le fichier local_file sur le serveur FTP, sous le nom de remote_file . Le mode de transfert mode spécifié doit être FTP_ASCII ou FTP_BINARY .

Note

Le paramètre startpos a été ajouté en PHP 4.3.0.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec ftp_put

```
<?php
 $upload = ftp_put($conn_id, $destination_file, $source_file, FTP_ASCII);
?>
```

Voir aussi ftp_fput , ftp_nb_fput et ftp_nb_put .

## 6.11.31 ftp_pwd() : Retourne le nom du dossier courant

string **ftp_pwd** ( resource ftp_stream )

ftp_pwd retourne le nom du dossier courant, ou FALSE en cas d'erreur.

## 6.11.32 ftp_quit() : Ferme une connexion FTP

void **ftp_quit** ( resource ftp_stream )

ftp_quit ferme la connexion ftp_stream .

ftp_quit est un alias de ftp_close .

## 6.11.33 ftp_raw() : Envoie une commande FTP brute

array **ftp_raw** ( resource ftp_stream , string command )

ftp_raw envoie la commande FTP brute command au serveur FTP identifié par ftp_stream . ftp_raw retourne alors la réponse du serveur sous forme d'un tableau de chaînes. Aucun traitement de la réponse n'est fait et ftp_raw ne tente pas de savoir si la commande a réussi ou échouée.

Utilisation de ftp_raw pour se logger manuellement

```
<?php
 $fp = ftp_connect("ftp.exemple.com");

/* Ceci est la même chose que
   ftp_login($fp, "joeblow", "secret"); */

 ftp_raw($fp, "USER joeblow");
 ftp_raw($fp, "PASS secret");
?>
```

Voir aussi ftp_exec .

## 6.11.34 ftp_rawlist() : Fait une liste détaillée de fichiers dans un dossier

array **ftp_rawlist** ( resource ftp_stream , string directory )

ftp_rawlist exécute la commande FTP LIST, et retourne le résultat dans un tableau. Chaque élément du tableau correspond à une ligne du résultat de la commande. Le résultat n'est pas analysé, et est retourné brut. L'identifiant de système retourné par ftp_systype sera utile pour déterminer la façon d'interpréter le résultat.

Voir aussi ftp_nlist .

## 6.11.35 ftp_rename() : Renomme un fichier sur un serveur FTP

bool **ftp_rename** ( resource ftp_stream , string from , string to )

ftp_rename renomme le fichier ou dossier from en to , sur le serveur ftp_stream .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.11.36 ftp_rmdir() : Efface un dossier FTP

bool **ftp_rmdir** ( resource ftp_stream , string directory )

ftp_rmdir efface le dossier directory .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ftp_mkdir .

## 6.11.37 ftp_set_option() : Modifie les options de la connexion FTP

bool **ftp_set_option** ( resource ftp_stream , int option , mixed value )

ftp_set_option retourne TRUE si l'option option a pu être modifiée avec la nouvelle valeur value , et FALSE sinon. Un message d'erreur sera affiché si option n'est pas supporté, ou si la valeur value n'est pas une valeur valide pour l'option option .

Cette fonction contrôle différentes options de connexion. Le paramètre value dépend de l'option option choisie. Actuellement, les options suivantes sont supportées :

| | |
|---|---|
| FTP_TIMEOUT_SEC | Modifie le délai d'expiration pour les opérations réseau. value doit être un entier positif ou null. La valeur par défaut est de 90 secondes. |
| FTP_AUTOSEEK | Lorsque cette option est activée, les requêtes de GET et PUT avec le paramètre resumepos ou startpos va générer un positionnement automatique du pointeur dans le fichier. C'est le comportement par défaut. |

Exemple avec ftp_set_option

```
<?php
 // modifie le délai d'expiration pour les opérations
  //réseau à 10 secondes
 ftp_set_option($conn_id, FTP_TIMEOUT_SEC, 10);
?>
```

Voir aussi ftp_get_option .

## 6.11.38 ftp_site() : Envoie la commande SITE au serveur

bool **ftp_site** ( resource ftp_stream , string cmd )

ftp_site envoie la commande cmd au serveur FTP. Les commandes SITE ne sont pas normalisées, et peuvent varier d'un serveur à l'autre. Elles permettent de gérer notamment les permissions de fichier, et les groupes.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.11.39 ftp_size() : Retourne la taille d'un fichier.

int **ftp_size** ( resource ftp_stream , string remote_file )

ftp_size retourne la taille du fichier en cas de succès, et FALSE sinon.

ftp_size retourne la taille d'un fichier sur un serveur FTP. Si une erreur survient, ou que le fichier n'existe pas, la valeur −1 est retournée. Certains serveurs FTP ne supportent pas cette fonction.

Voir aussi ftp_rawlist .

## 6.11.40 ftp_ssl_connect() : Ouvre une connexion FTP sécurisée avec SSL

resource **ftp_ssl_connect** ( string host , *int port* , *int timeout* )

ftp_ssl_connect retourne un flux FTP sécurisé par SSL en cas de succès et FALSE sinon.

ftp_ssl_connect ouvre une connexion SSL−FTP avec le serveur FTP host . Le paramètre port spécifie un port alternatif. Si il est omis, ou bien configuré à 0, alors la valeur par défaut est 21.

Le paramètre timeout spécifie la durée d'expiration des opérations de réseaux ultérieures. Si elle est omise, cette valeur vaut 90 secondes. La durée d'expiration des opérations réseau peut être lue et modifiée avec les fonctions ftp_set_option et ftp_get_option .

| Note |
| --- |
| Pourquoi cette fonction peut ne pas exister<br><br>ftp_ssl_connect est uniquement disponible si OpenSSL a été ajouté dans la liste des extensions de PHP . Si vous avez compilé PHP avec le support FTP mais que cette fonction est indisponible, alors c'est la raison. |

Voir aussi ftp_connect .

## 6.11.41 ftp_systype() : Retourne un identifiant de type de serveur FTP.

string **ftp_systype** ( resource ftp_stream )

ftp_systype retourne le type de serveur, ou FALSE en cas d'erreur.

# 6.12 Fonctions

## 6.12.1 Introduction

Ces fonctions effectuent les manipulations liées à la gestion des fonctions.

## 6.12.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.12.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.12.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.12.5 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Constantes prédefinies
- call_user_func_array
- call_user_func
- create_function
- func_get_arg
- func_get_args
- func_num_args
- function_exists
- get_defined_functions
- register_shutdown_function
- register_tick_function
- unregister_tick_function


## 6.12.7 call_user_func() : Appelle une fonction utilisateur

mixed **call_user_func** ( callback function , *mixed parameter* , *mixed ...* )

call_user_func appelle la fonction utilisateur function_name , et lui passe les paramètres parameter .

Exemple avec call_user_func

```
<?php
 function barbier ($type) {
   print "Vous vouliez une coupe $type, pas de problème";
 }

 call_user_func('barbier', "à la one again");
 call_user_func('barbier', "militaire");
 call_user_func('barbier', "au bol");
?>
```

Les méthodes d'objets peuvent être appelées statiquement en passant array($objectname, $methodname) au paramètre function .

Exemple avec call_user_func et une méthode de classe

```
<?php
class maclasse {
  function dit_bonjour() {
    print "Bonjour!\n";
  }
}

$classname = "maclasse";

call_user_func(array($classname,'dit_bonjour'));
?>
```

Voir aussi call_user_func_array , call_user_method et call_user_method_array .

## 6.12.8 create_function() : Crée une fonction anonyme (style lambda)

string **create_function** ( string args , string code )

create_function crée une fonction anonyme, à partir des paramètres passés, et retourne un nom de fonction unique. Généralement, les arguments args sont présentés sous la forme d'une chaîne à guillemets simples, et la même recommandation vaut pour code . La raison de l'utilisation des guillemets simples est de proteger les noms de variables du remplacement par leur valeur. Si vous utilisez les guillemets doubles, n'oubliez pas d'échapper les noms de variables (i.e. \$avar ).

Vous pouvez utiliser cette fonction pour (par exemple) créer une fonction à partir d'informations récoltés durant l'exécution.

Création d'une fonction anonyme avec create_function

```
<?php
 $newfunc = create_function('$a,$b','return "ln($a) + ln($b) = ".log($a * $b);');
 echo "Nouvelle fonction anonyme : $newfunc\n";
 echo $newfunc(2,M_E)."\n";
// affichera :
// Nouvelle fonction anonyme : lambda_1
// ln(2) + ln(2.718281828459) = 1.6931471805599
?>
```

Ou, pour pouvoir appliquer une fonction générique à une liste d'arguments.

Traitement générique par fonction avec create_function

```
<?php
function process($var1, $var2, $farr) {
    for ($f=0; $f < count($farr); $f++)
    echo $farr[$f]($var1,$var2)."\n";
}

// Création d'une série de fonction mathématiques
$f1 = 'if ($a>=0) {return "b*a^2 = ".$b*sqrt($a);} else {return FALSE;}';
$f2 = "return \"min(b^2+a, a^2,b) = \".min(\$a*\$a+\$b,\$b*\$b+\$a);";
$f3 = 'if ($a> 0 && $b != 0) {return "ln(a)/b = ".log($a)/$b;} else {return FALSE;}';
$farr = array(
    create_function('$x,$y', 'return "un peu de trigo : ".(sin($x) + $x*cos($y));'),
    create_function('$x,$y', 'return "une hypoténuse: ".sqrt($x*$x + $y*$y);'),
    create_function('$a,$b', $f1),
    create_function('$a,$b', $f2),
    create_function('$a,$b', $f3)
    );
```

```
echo "\nUtilisation de la première liste de fonctions anonymes\n";
echo "paramétres: 2.3445, M_PI\n";
process(2.3445, M_PI, $farr);

// Maintenant une liste de fonction sur chaîne de caractères
$garr = array(
    create_function('$b,$a','if (strncmp($a,$b,3) == 0) return "** \"$a\" '.
    'et \"$b\"\n** Ces chaînes de ressemblent!! (regarde les trois premiers caractères)";'),
    create_function('$a,$b','; return "CRCs: ".crc32($a)." , ".crc32(b);'),
    create_function('$a,$b','; return "similarité(a,b) = ".similar_text($a,$b,&$p)."($p%)";')
    );
echo "\nUtilisation de la secondes liste de fonctions anonymes\n";
process("Twas brilling and the slithy toves", "Twas the night", $garr);
?>
```

Et lorsque vous utilisez le code ci−dessus, l'affichage sera

```
 Utilisation de la première liste de fonctions anonymes
paramétres: 2.3445, M_PI
un peu de trigo : −1.6291725057799
une hypoténuse: 3.9199852871011
b*a^2 = 4.8103313314525
min(b^2+a, a^2,b) = 8.6382729035898
ln(a)/b = 0.27122299212594

Utilisation de la secondes liste de fonctions anonymes
** "Twas the night" et "Twas brilling and the slithy toves"
** Ces chaînes de ressemblent!! (regarde les trois premiers caractères)
CRCs: −725381282 , 1908338681
similarité(a,b) = 11(45.833333333333%)
```

Mais l'utilisation la plus courante des fonctions lambda est la fonction de callback, par exemple lors de l'utilisation de array_walk ou usort

```
Utilisation de fonctions anonymes comme fonction de callback
```

```
 <?php
$av = array("la ","une ","cette ","une certaine ");
array_walk($av, create_function('&$v,$k','$v = $v."maison";'));
print_r($av);  // En PHP 3 utilisez var_dump()
// affiche:
// Array
// (
//   [0] => la maison
//   [1] => une maison
//   [2] => cette maison
//   [3] => une certaine maison
// )
// un tableau de chaîne classé par taille
$sv = array("petite","moyenne","tres longue","vraiment tres longue");
print_r($sv);
// affiche:
// Array
// (
//   [0] => petite
//   [1] => moyenne
//   [2] => tres longue
//   [3] => vraiment tres longue
// )
// Tri par ordre de taille décroissant
usort($sv, create_function('$a,$b','return strlen($b) − strlen($a);'));
print_r($sv);
// outputs:
// Array
// (
//   [0] => vraiment tres longue
//   [1] => tres longue
//   [2] => moyenne
//   [3] => petite
// )
?>
```

## 6.12.9 func_get_arg() : Retourne un élément de la liste des arguments

mixed **func_get_arg** ( int arg_num )

func_get_arg retourne l'argument à la position arg_num dans la liste d'argument d'une fonction utilisateur. Les arguments sont comptés en commençant à zéro. func_get_arg générera une alerte si elle est appelée hors d'une fonction.

Si arg_num est plus grand que le nombre d'arguments passés, une alerte est générée et la fonction retourne FALSE .

Exemple avec func_get_arg

```
<?php
 function foo() {
    $numargs = func_num_args();
    echo "Nombre d'arguments: $numargs<BR>\n";
    if ($numargs >= 2) {
    echo "Le second argument est: " . func_get_arg (1) . "<BR>\n";
    }
 }
 foo(1, 2, 3);
?>
```

func_get_arg peut être utilisé conjointement à func_num_args et func_get_args pour permettre aux fonctions utilisateurs d'accepter un nombre variable d'arguments.

Note

func_get_arg a été ajoutée en PHP 4.

## 6.12.10 func_get_args() : Retourne les arguments d'une fonction sous forme de tableau

array **func_get_args** ( void )

func_get_args retourne un tableau dont les éléments correspondent aux éléments de la liste d'arguments de la fonction. func_get_args générera une alerte si elle est appelée hors d'une fonction.

Exemple avec func_get_args

```
<?php
 function foo() {
   $numargs = func_num_args();
   echo "Nombre d'arguments: $numargs<BR>\n";
   if ($numargs >= 2) {
      echo "Le second argument est : " . func_get_arg (1) . "<BR>\n";
   }
   $arg_list = func_get_args();
   for ($i = 0; $i < $numargs; $i++) {
      echo "L'argument $i est : " . $arg_list[$i] . "<BR>\n";
   }
 }
 foo(1, 2, 3);
?>
```

func_get_arg peut être utilisé conjointement à func_num_args et func_get_args pour permettre aux fonctions utilisateurs d'accepter un nombre variable d'arguments.

Note

func_get_arg a été ajoutée en PHP 4.

## 6.12.11 func_num_args() : Retourne le nombre d'arguments passé à la fonction

int **func_num_args** ( void )

func_num_args retourne le nombre d'arguments passé à la fonction utilisateur courante. func_num_args générera une alerte si elle est appelée hors d'une fonction.

Exemple avec func_num_args

```
<?php
 function foo() {
   $numargs = func_num_args();
   echo "Nombre d'arguments: $numargs\n";
 }
 foo (1, 2, 3);    // affiche 'Nombre d'arguments: 3'
```

```
?>
```

func_get_arg peut être utilisé conjointement à func_num_args et func_get_args pour permettre aux fonctions utilisateurs d'accepter un nombre variable d'arguments.

| Note |
| --- |
| func_get_arg a été ajoutée en PHP 4. |

## 6.12.12 function_exists() : Indique si une fonction est définie

bool **function_exists** ( string function_name )

function_exists vérifie la liste des fonctions définies par l'utilisateur. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec function_exists

```
<?php
 if (function_exists('imap_open')) {
   echo "Les fonctions IMAP sont disponibles.<BR>\n";
 } else {
   echo "Les fonctions IMAP ne sont pas disponibles.<BR>\n";
 }
?>
```

Notez qu'une fonction peut exister, même si elle est indisponible, à cause de la configuration ou des options de compilation.

Voir aussi method_exists et get_defined_functions .

## 6.12.13 get_defined_functions() : Liste toutes les fonctions définies

array **get_defined_functions** ( void )

get_defined_functions retourne un tableau multi− dimensionnel, contenant la liste de toutes les fonctions définies, aussi bien les fonctions internes à PHP que celle déjà définie par l'utilisateur. Les noms des fonctions internes sont accessibles via $arr["internal"] , et les fonctions utilisateur sont accessibles via $arr["user"] .

Exemple avec get_defined_functions

```
<?php
 function maligne($id, $data) {
   return "<tr><th>$id</th><td>$data</td></tr>\n";
 }
 $arr = get_defined_functions();
 print_r($arr);
?>
```

Ce script va afficher :

```
 Array
(
    [internal] => Array
        (
            [0] => zend_version
            [1] => func_num_args
            [2] => func_get_arg
            [3] => func_get_args
            [4] => strlen
            [5] => strcmp
            [6] => strncmp
            ...
            [750] => bcscale
            [751] => bccomp
        )
    [user] => Array
```

```
        (
            [0] => maligne
        )
)
```

Voir aussi get_defined_vars et get_defined_constants .

## 6.12.14 register_shutdown_function() : Enregistre une fonction pour exécution à l'extinction

void **register_shutdown_function** ( callback function )

register_shutdown_function enregistre la fonction func pour exécution à l'extinction du script.

Plusieurs appels à register_shutdown_function sont possibles dans le même script, et les fonctions seront appelées dans le même ordre que celui dans elles sont enregistrées. SI vous appelez exit durant l'une des fonctions d'extinction, le processus sera définitivement arrété, sans que les autres fonctions soient appelées.

Les fonctions d'extinction sont appelées après la fin de la requête (notamment les buffers ont été vidés), ce qui fait qu'il est impossible d'afficher du texte depuis cette fonction, avec echo ou print , ou encore de lire le contenu des buffers avec ob_get_contents .

| Note |
| --- |
| Typiquement, les fonctions non définies causent des erreurs fatales en PHP , mais lorsque la fonction function est appelée avec register_shutdown_function en étant non définie, une erreur de type E_WARNING est générée. De plus, pour des raisons internes à PHP cette erreur va être indiquée comme Unknown at line #0 . |

Voir aussi auto_append_file , exit , et la section sur la gestion des connexions .

## 6.12.15 register_tick_function() : Enregistre une fonction exécutée à chaque tick

void **register_tick_function** ( callback function , *mixed arg* )

register_tick_function enregistre la fonction func pour être exécutée à chaque fois qu'un tick est appelé.

Exemple avec register_tick_function

```php
<?php
// Utilisation d'une fonction de callback
register_tick_function('ma_function', TRUE);

// Utilisation d'une méthode d'objet
$objet = new ma_classe();
register_tick_function(array(&$objet, 'ma_methode'), TRUE);
?>
```

Voir aussi declare et unregister_tick_function .

## 6.12.16 unregister_tick_function() : Annule la fonction exécutée à chaque tick

void **unregister_tick_function** ( string function_name )

unregister_tick_function annule l'exécution automatique de func à chaque tick .

# 6.13 HTTP

## 6.13.1 Introduction

Ces fonctions permettent de travailler sur les informations transmises au navigateur, via le protocole HTTP.

## 6.13.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.13.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.13.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.13.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.13.6 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- header
- headers_list
- headers_sent
- setcookie


## 6.13.8 headers_list() : Retourne la liste des entêtes de réponse du script courant

array **headers_list** ( void )

headers_list retourne un tableau avec la liste des entêtes qui seront retournés au navigateur. Pour déterminer si ces entêtes ont été déjà envoyés ou pas, utilisez la fonction headers_sent .

Exemple avec headers_list

```
<?php

/* setcookie() va ajouter un entête */
setcookie('foo','bar');

/* Définit un entête de réponse
   Il sera ignoré par la majorité des navigateurs */
header("X-Sample-Test: foo");

/* Spécification de la réponse en texte simple */
header('Content-type: text/plain');

/* Quels sont les entêtes qui vont être envoyés? */
var_dump(headers_list());


/* Export
   ======

array(4) {
  [0]=>
  string(29) "X-Powered-By: PHP/5.0.0"
  [1]=>
  string(19) "Set-Cookie: foo=bar"
  [2]=>
  string(18) "X-Sample-Test: foo"
  [3]=>
  string(24) "Content-type: text/plain"
}

*/

?>
```

Voir aussi headers_sent , header et setcookie .

## 6.13.9 headers_sent() : Indique si les en-têtes HTTP ont déjà été envoyés

bool **headers_sent** ( *string &file* , *int &line* )

headers_sent retourne TRUE si les en-têtes HTTP ont déjà été envoyés, et FALSE sinon. Si les paramètres optionnels file et line sont fournis, headers_sent va placer le nom du fichier source et le numéro de ligne qui ont débuté l'affichage, dans les variables file et line .

Vous ne pouvez plus envoyer d'en-têtes avec la fonction header une fois que le bloc d'en-tête a été fermé. En utilisant cette fonction, vous pouvez au moins éviter de voir s'afficher les erreurs HTTP reliées. Une autre option est d'utilise le contrôle de sortie .

Note

Nouveaux paramètres

Les paramètres optionnels file et line ont été ajouté en PHP 4.3.0.

Exemple avec headers_sent

```
<?php

// Si aucun en-tête n'a été envoyé, envoyons en un
if (!headers_sent()) {
    header ('Location: http://www.nexen.net/');
    exit;
}

// Voici un exemple d'utilisation des paramètres optionnels de fichier et de ligne
// disponibles depuis PHP 4.3.0. Notez que $filename et $linenum sont
// transmis pour utilisation ultérieure. Ne les assignez pas avant de les utiliser.
if (!headers_sent($filename, $linenum)) {
    header ('Location: http://www.nexen.net/');
    exit;

// Vous allez probablement déclencher une erreur ici
} else {

    print "Les en-têtes ont déjà été envoyés, depuis le fichier $filename à la ligne $linenum\n" .
```

```
        "Il est donc impssible de vous rediriger automatiquement, aussi veuillez
        cliquez <a href=\"http://www.nexen.net\">ici</a>.\n";
    exit;
}

?>
```

Voir aussi ob_start , trigger_error et header pour plus de détails sur les tenants et aboutissants.

## 6.13.10 setcookie() : Envoie un cookie

bool **setcookie** ( string name , *string value* , *int expire* , *string path* , *string domain* , *int secure* )

setcookie définit un cookie qui sera envoyé avec le reste des en−têtes. Les cookies doivent passer avant toute autre en−tête (c'est une restriction des cookies, pas de PHP). Cela vous impose d'appeler cette fonction avant toute balise <HTML> ou <HEAD> .

| Note |
| --- |
| En PHP 4, vous pouvez utiliser la bufferisation de sortie pour pouvoir envoyer du contenu avant d'appeler cette fonction, avec la contrepartie que toute votre page sera envoyé en une fois. Vous pouvez faire cela en appelant ob_start et ob_end_flush dans votre script, ou en activant la directive output_buffering dans votre fichier de configuration php.ini . |

Tous les arguments sauf name (nom) sont optionnels. Si seul le nom est présent, le cookie portant ce nom sera supprimé du navigateur de l'internaute. Vous pouvez aussi utiliser une chaîne vide comme valeur, pour ignorer un argument. Le paramètre expire est un timestamp UNIX, du même genre que celui retourné par time ou mktime . Le paramètre secure indique que le cookie doit être uniquement transmis à travers une connexion HTTPS sécurisée.

Une fois que le cookie a été placé, il est accessible dans les variables globales $_COOKIE ou bien $HTTP_COOKIE_VARS arrays. Notez que les autoglobales telles que $_COOKIE sont disponibles en PHP depuis la version 4.1.0 . $HTTP_COOKIE_VARS existe depuis PHP 3. Les valeurs de cookies existent aussi dans la variable $_REQUEST .

Erreurs communes :

- Les cookies ne seront accessibles qu'au chargement de la prochaine page, ou au rechargement de la page courante.
- Les cookies doivent être effacés avec les mêmes paramètres que ceux utilisés lors de leur création.

En PHP 3, les appels multiples à setcookie dans le même script seront effectués dans l'ordre inverse. Si vous essayez d'effacer un cookie avant d'insérer une nouvelle valeur, vous devez placer l'insertion avant l'effacement. En PHP 4, les appels multiples à setcookie sont effectués dan un ordre naturel.

Les appels multiples à setcookie dans la même page seront réalisés dans l'ordre inverse. Si vous essayez d'effacer un cookie avant d'insérer une autre valeur, il faut placer l'insertion avant l'effacement.

Quelques exemples :

Exemples avec setcookie

```
<?php
 setcookie("TestCookie","Valeur de test");
 setcookie("TestCookie",$value,time()+3600);  /* expire dans une heure */
```

```
  setcookie("TestCookie",$value,time()+3600,"/~rasmus/",".utoronto.ca",1);
?>
```

Notez que la partie "valeur" du cookie sera automatiquement encodée URL lorsque vous envoyez le cookie, et lorsque vous le recevez, il sera automatiquement décodé, et affecté à la variable du même nom que le cookie. Pour voir le résultat, essayez les scripts suivants :

Affectation des valeurs de cookie

```
 <?php
// Afficher un cookie
 echo $_COOKIE["TestCookie"];
echo $HTTP_COOKIE_VARS["TestCookie"];

// Une autre méthode pour afficher tous les cookies
print_r($_COOKIE);
?>
```

Lorsque vous effacez un cookie, vous devriez toujours vous assurer que la date d'expiration du cookie est déjà passée, pour déclencher le mécanisme de votre navigateur. Voici comment procéder :

Exemple d'effacement de cookies avec setcookie

```
 <?php
// utilisation de la date moins une heure
setcookie ("TestCookie", "", time() − 3600);
setcookie ("TestCookie", "", time() − 3600, "/~rasmus/", ".exemple.com", 1);
?>
```

Vous pouvez aussi utiliser les cookies avec des tableaux, en utilisant la notation des tableaux. Cela a pour effet de créer autant de cookies que votre tableau a d'éléments, mais lorsque les cookies seront reçus par PHP, les valeurs seront placées dans un tableau :

Utilisation des tableaux avec setcookie

```
 <?php
  setcookie( "cookie[three]", "cookiethree" );
  setcookie( "cookie[two]", "cookietwo" );
  setcookie( "cookie[one]", "cookieone" );
  if ( isset( $cookie ) ) {
    while( list( $name, $value ) = each( $cookie ) ) {
      echo "$name == $value<br>\n";
    }
  }
?>
```

Note

Pour plus d'informations sur les cookies, voyez les spécifications de cookies de Netscape à http://www.netscape.com/newsref/std/cookie_spec.html et RFC 2965 .

Vous pourrez noter que le paramètre expire prend un timestamp unique, et non pas la date au format Jour, JJ−Mois−AAAA HH:MM:SS GMT , car PHP fait la conversion en interne.

Microsoft Internet Explorer 4 utilisé avec le Service Pack 1 ne gère pas bien les cookies qui possèdent un paramètre path .

Netscape Communicator 4.05 et Microsoft Internet Explorer 3.x semblent ne pas gérer correctement les cookies lorsque path et time> ne sont pas fournis.

Voir aussi header .

# 6.14 Images

## 6.14.1 Introduction

PHP ne se limite pas à la génération de pages HTML. Il peut aussi servir à créer et manipuler des images, dans un grand choix de formats, comme gif, png, jpg, wbmp et xpm. Et PHP peut même générer directement des images pour le navigateur, avec la librarie GD. GD et PHP auront aussi besoin d'autres librairies, en fonction des formats que vous souhaitez utiliser.

Vous pouvez utiliser les fonctions PHP pour obtenir les tailles des images aux formats JPEG , GIF , PNG , SWF , TIFF et JPEG2000 .

| Note |
| --- |
| Lisez la section sur les pré−requis pour savoir comment améliorer les possibilités de lecture, modification et écriture d'images, ainsi que les possibilités de lecture d'images prises par des appareils numériques. |

## 6.14.2 Pré−requis

Si vous disposez de la librairie GD (disponible à http://www.boutell.com/gd/ ) vous pourrez aussi créer et manipuler ces images.

Les formats des images que vous pourrez manipuler dépendent de la version de GD que vous installerez, et de toute autre librairie dont GD a besoin pour traiter à ces images. Les versions antérieures à la version 1.6 supportent le GIF , mais pas le PNG . Pour les versions plus récentes, c'est le contraire.

| Note |
| --- |
| Depuis PHP 4.3, il existe une version de GD qui est distribuée avec PHP. Cette version contient des fonctionnalités supplémentaires, comme les canaux alpha, et il est recommandé de l'utiliser de préférence à la librairie externe, car elle est mieux supportée, et bien plus stable. |

Vous pouvez aussi améliorer GD en lui ajoutant des formats d'images supplémentaires.

| Format d'image | Librairie à télécharger | Notes |
| --- | --- | --- |
| gif | | Uniquement supporté en versions de GD antérieures à gd−1.6. Le support des images GIF est disponible **en lecture seule** depuis PHP 4.3.0, et dans la version de la librairie qui est fournie avec la distribution de PHP . |
| jpeg−6b | ftp://ftp.uu.net/graphics/jpeg/ | |
| png | http://www.libpng.org/pub/png/libpng.html | Uniquement supporté avec GD plus récente que gd−1.6. |
| xpm | ftp://metalab.unc.edu/pub/Linux/libs/X/!INDEX.html | Il est probable que vous ayez déjà cette librairie disponible si votre système dispose d'un environnement X. |

Vous pouvez aussi améliorer GD en lui ajoutant des fonctionalités de manipulation de polices. Les librairies suivantes sont supportées :

| Librarie de police | Téléchargement | Notes |
| --- | --- | --- |
| FreeType 1.x | http://www.freetype.org/ | |
| FreeType 2 | http://www.freetype.org/ | |
| T1lib | ftp://sunsite.unc.edu/pub/Linux/libs/graphics/) | Support des polices de Type 1. |

Si vous avez compilé PHP avec l'option −−enable−exif vous serez capable d'accéder aux données stockées dans les entêtes des images JPEG et TIFF . De cette façon, vous pouvez lire les méta données générées par les appareils numériques. Ces fonctions ne requièrent par la librairie GD .

| Note |
| --- |
| PHP ne requiert aucune librairie supplémentaire pour gérer les données EXIF. |

## 6.14.3 Installation

Pour activer le support de GD, vous devez compiler PHP avec l'option −−with−gd[=DIR] , où DIR est le dossier d'installation de GD. Il est recommandé d'utiliser la version de GD qui est distribué avec PHP, en utilisant simplement l'option −−with−gd . Sous Windows, vous devez inclure la librairie php_gd2.dll comme extension dans le fichier php.ini . Il y a aussi la librairie php_gd.dll pour GD version 1.x mais elle n'est pas recommandée.

Pour désactiver le support de GD en PHP 3, ajoutez simplement l'option −−without−gd dans votre ligne de configuration.

Augmentez les possibilités des GD de gérer d'autres formats d'images en spécifiant les options de compilations suivantes :

| Format d'image | Option de compilation |
| --- | --- |
| jpeg−6b | Pour activer le support de la librairie JPEG−6b, ajouter l'option −−with−jpeg−dir=DIR . |
| png | Pour activer le support de la librairie PNG, ajouter l'option −−with−png−dir=DIR . Notez que libpng requiert la librairie zlib, et donc, il vous faudra ajouter aussi −−with−zlib−dir[=DIR] dans votre ligne de compilation. |
| xpm | Pour activer le support de la librairie XPM, ajoutez l'option −−with−xpm−dir=DIR . Si le script de compilation n'est pas capable de trouver les librairies nécessaires, il vous faudra ajouter le chemin vers les librairies X11. |

Augmentez les possibilités de GD pour qu'elle manipule différentes types de polices de caractères en ajoutant les options de compilations suivantes :

| Librairie | Option de configuration |
| --- | --- |
| FreeType 1.x | Pour activer le support de FreeType 1.x, ajoutez l'option −−with−ttf[=DIR] . |
| FreeType 2 | Pour activer le support de FreeType 2, ajoutez l'option −−with−freetype−dir=DIR . |
| T1lib | Pour activer le support de T1lib (Type 1 fonts), ajoutez l'option −−with−t1lib[=DIR] . |
| Native TrueType string function | Pour activer le support des chaînes de caractères TrueType, ajoutez l'option −−enable−gd−native−ttf . |

## 6.14.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

Exif supporte automatiquement la conversion en Unicode et JIS pour les commentaires des utilisateurs, lorsque le module mbstring est disponible. Cela se fait en décodant le commentaire avec le bon jeu de caractères. Le résultat peut alors être encodé dans le jeu de caractères de votre serveur HTTP.

| Nom | Par défaut | Modifiable |
| --- | --- | --- |
| exif.encode_unicode | "ISO−8859−15" | PHP_INI_ALL |
| exif.decode_unicode_motorola | "UCS−2BE" | PHP_INI_ALL |
| exif.decode_unicode_intel | "UCS−2LE" | PHP_INI_ALL |
| exif.encode_jis | "" | PHP_INI_ALL |
| exif.decode_jis_motorola | "JIS" | PHP_INI_ALL |
| exif.decode_jis_intel | "JIS" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*exif.encode_unicode string*

exif.encode_unicode définit le jeu de caractères UNICODE utilisé pour les commentaires. Par défaut, c'est ISO−8859−15 , qui devrait fonctionner pour les pays non asiatiques. Cette option peut être vide, ou bien contenir un jeu de caractères qui est supporté par mbstring. Si cette option est vide, la configuration de mbstring sera utilisée.

*exif.decode_unicode_motorola string*
: exif.decode_unicode_motorola définit le jeu de caractères interne pour les utilisateurs Unicode, si le commentaire de l'image est en big−endian (ordre de motorola). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par mbstring. Par défaut, c'est UCS−2BE .

*exif.decode_unicode_intel string*
: exif.decode_unicode_intel définit le jeu de caractères interne pour les utilisateurs Unicode, si le commentaire de l'image est en little−endian (ordre d'intel). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par mbstring. Par défaut, c'est UCS−2LE .

*exif.encode_jis string*
: exif.encode_jis définit le jeux de caractères pour les commentaires en JIS. Par défaut, c'est une valeur vide, ce qui force la fonction à utiliser le jeu de caractères interne de mbstring.

*exif.decode_jis_motorola string*
: exif.decode_jis_motorola définit le jeu de caractères interne pour les utilisateurs JIS, si le commentaire de l'image est en big−endian (ordre de motorola). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par mbstring. Par défaut, c'est JIS .

*exif.decode_jis_intel string*
: exif.decode_jis_intel définit le jeu de caractères interne pour les utilisateurs JIS, si le commentaire de l'image est en little−endian (ordre d'intel). Cette option ne peut être vide, mais vous pouvez y spécifier une liste de jeux de caractères utilisés par mbstring. Par défaut, c'est JIS .

## 6.14.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.14.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*IMG_GIF (entier)*
*IMG_JPG (entier)*
*IMG_JPEG (entier)*
*IMG_PNG (entier)*
*IMG_WBMP (entier)*
*IMG_XPM (entier)*
*IMG_COLOR_TILED (entier)*
*IMG_COLOR_STYLED (entier)*
*IMG_COLOR_BRUSHED (entier)*
*IMG_COLOR_STYLEDBRUSHED (entier)*
*IMG_COLOR_TRANSPARENT (entier)*
*IMG_ARC_ROUNDED (entier)*
*IMG_ARC_PIE (entier)*

*IMG_ARC_CHORD (entier)*
*IMG_ARC_NOFILL (entier)*
*IMG_ARC_EDGED (entier)*
*IMAGETYPE_GIF (entier)*
*IMAGETYPE_JPG (entier)*
*IMAGETYPE_PNG (entier)*
*IMAGETYPE_SWF (entier)*
*IMAGETYPE_PSD (entier)*
*IMAGETYPE_BMP (entier)*
*IMAGETYPE_TIFF_II (entier)*
*IMAGETYPE_TIFF_MM (entier)*
*IMAGETYPE_JPC (entier)*
*IMAGETYPE_JP2 (entier)*
*IMAGETYPE_JPX (entier)*
*IMAGETYPE_SWC (entier)*

## 6.14.7 Exemples

Création d'une image PNG avec PHP

```php
<?php
   header("Content-type: image/png");
   $string = $_GET['text'];
   $im     = imagecreatefrompng("images/button1.png");
   $orange = imagecolorallocate($im, 220, 210, 60);
   $px     = (imagesx($im) - 7.5 * strlen($string)) / 2;
   imagestring($im, 3, $px, 9, $string, $orange);
   imagepng($im);
   imagedestroy($im);
?>
```

Cet exemple doit être appelé depuis une page HTML avec une balise image telle que : <img src="button.php?text"> . Le script ci-dessus, button.php, prend la chaîne "text" et l'inscrit sur le fond d'image appelé "images/button1.png" et l'affiche. C'est une méthode très pratique pour éviter de redessiner un nouveau bouton, dès que vous changez le texte d'un bouton. De cette façon, ils sont générés dynamiquement.


- Introduction
- Pré-requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- exif_imagetype
- exif_read_data
- exif_thumbnail
- gd_info
- getimagesize
- image_type_to_mime_type
- image2wbmp
- imagealphablending
- imageantialias
- imagearc
- imagechar
- imagecharup
- imagecolorallocate
- imagecolorallocatealpha

- imagecolorat
- imagecolorclosest
- imagecolorclosestalpha
- imagecolorclosesthwb
- imagecolordeallocate
- imagecolorexact
- imagecolorexactalpha
- imagecolormatch
- imagecolorresolve
- imagecolorresolvealpha
- imagecolorset
- imagecolorsforindex
- imagecolorstotal
- imagecolortransparent
- imagecopy
- imagecopymerge
- imagecopymergegray
- imagecopyresampled
- imagecopyresized
- imagecreate
- imagecreatefromgd
- imagecreatefromgd2
- imagecreatefromgd2part
- imagecreatefromgif
- imagecreatefromjpeg
- imagecreatefrompng
- imagecreatefromstring
- imagecreatefromwbmp
- imagecreatefromxbm
- imagecreatefromxpm
- imagecreatetruecolor
- imagedashedline
- imagedestroy
- imageellipse
- imagefill
- imagefilledarc
- imagefilledellipse
- imagefilledpolygon
- imagefilledrectangle
- imagefilltoborder
- imagefontheight
- imagefontwidth
- imageftbbox
- imagefttext
- imagegammacorrect
- imagegd
- imagegd2
- imagegif
- imageinterlace
- imageistruecolor
- imagejpeg
- imageline
- imageloadfont
- imagepalettecopy

- imagepng
- imagepolygon
- imagepsbbox
- imagepscopyfont
- imagepsencodefont
- imagepsextendfont
- imagepsfreefont
- imagepsloadfont
- imagepsslantfont
- imagepstext
- imagerectangle
- imagerotate
- imagesavealpha
- imagesetbrush
- imagesetpixel
- imagesetstyle
- imagesetthickness
- imagesettile
- imagestring
- imagestringup
- imagesx
- imagesy
- imagetruecolortopalette
- imagettfbbox
- imagettftext
- imagetypes
- imagewbmp
- iptcembed
- iptcparse
- jpeg2wbmp
- png2wbmp
- read_exif_data

## 6.14.9 exif_read_data() : Lit les en−têtes EXIF des images JPEG et TIFF

array **exif_read_data** ( string filename , *string sections* , *bool arrays* , *bool thumbnail* )

exif_read_data lit les en−têtes EXIF des images JPEG et TIFF. Elle retourne un tableau associatif avec comme index les noms des en−têtes, et en valeur, les valeurs des en−têtes. Si aucune donnée n'a pu être lue, elle retourne FALSE .

filename est le nom du fichier à lire. Il ne peut pas être une URL.

sections est une liste de valeur séparée par des virgules des sections qui devront être présentées dans le tableau de résultat.

| | |
|---|---|
| FILE | FileName (nom du ficher), FileSize(taille du fichier), FileDateTime (date de modification du fichier), SectionsFound (sections trouvées) |
| COMPUTED | html (attributs HTML), Width (largeur), Height (Hauteur), IsColor (couleur ou noir et blanc) et quelques autres si ils sont disponibles. |
| ANY_TAG | Toutes les informations concernant cette balise, comme par exemple IFD0, EXIF, ... |
| IFD0 | Toutes les balises IFD0. Dans les images normales, ils contiennent les dimensions de l'images, etc... |
| THUMBNAIL | Un fichier qui contient une minitaure, si il y a un second IFD. Toutes les informations mises en balises à propos de cette miniature seront stockées dans cette section. |
| COMMENT | En−tête de commentaire des images JPEG. |
| EXIF | La section EXIF est une sous section de la section IFD0. Elle contient des informations plus détaillées sur les images. La plupart de ces index sont reliés aux appareils numériques. |

arrays spécifie si chaque section doit être un tableau ou non. Les sections **FILE** , **COMPUTED** et **THUMBNAIL** seront toujours transformées en tableau, car elle contiennent des noms qui risquent d'être en conflit.

thumbnail indique si il faut lire la miniature elle−même, et pas seulement les données en balises.

Note

Les en−têtes Exif sont souvent présents dans les en−têtes d'images JPEG/TIFF, générées par les appareils numériques, mais malheureusement, chaque fabriquant d'appareils ont tous une idée différente pour étiquetter leurs images, et vous ne pourrez pas compter sur des en−têtes communs.

Exemple avec exif_read_data

```
<?php
echo "test1.jpg : <br />\n";
$exif = exif_read_data ('tests/test1.jpg','IFD0');
echo $exif===false ? "No header data found.<br />\n" : "Image contains headers<br />";
$exif = exif_read_data ('tests/test2.jpg',0,true);
echo "test2.jpg : <br />\n";
foreach($exif as $key=>$section) {
    foreach($section as $name=>$val) {
        echo "$key.$name: $val<br />\n";
    }
}?>
```

La première commande échoue, car l'image n'a pas d'en−têtes d'informations.
test1.jpg:
No header data found.
test2.jpg:
FILE.FileName: test2.jpg
FILE.FileDateTime: 1017666176
FILE.FileSize: 1240
FILE.FileType: 2
FILE.SectionsFound: ANY_TAG, IFD0, THUMBNAIL, COMMENT
COMPUTED.html: width="1" height="1"
COMPUTED.Height: 1
COMPUTED.Width: 1
COMPUTED.IsColor: 1
COMPUTED.ByteOrderMotorola: 1
COMPUTED.UserComment: Exif test image.
COMPUTED.UserCommentEncoding: ASCII
COMPUTED.Copyright: Photo (c) M.Boerger, Edited by M.Boerger.
COMPUTED.Copyright.Photographer: Photo (c) M.Boerger
COMPUTED.Copyright.Editor: Edited by M.Boerger.
IFD0.Copyright: Photo (c) M.Boerger
IFD0.UserComment: ASCII
THUMBNAIL.JPEGInterchangeFormat: 134
THUMBNAIL.JPEGInterchangeFormatLength: 523
COMMENT.0: Comment #1.
COMMENT.1: Comment #2.
COMMENT.2: Comment #3end
THUMBNAIL.JPEGInterchangeFormat: 134
THUMBNAIL.Thumbnail.Height: 1
THUMBNAIL.Thumbnail.Height: 1

Note

Si l'image contient des données IFD0, alors la section COMPUTED contient l'entrée ByteOrderMotorola qui vaut 0 pour little−endian (intel) et 1 pour big−endian (motorola). Ceci a été ajoué en PHP 4.3.

Lorsqu'un en−tête Exif contient une note de Copyright, cet en−tête peut alors contenir lui même deux valeurs. Comme cette solution est incohérente avec les standards Exif 2.10, la section COMPUTED retournera les deux en−têtes, **Copyright.Photographer** et **Copyright.Editor** , tandis que les sections IFD0 contiennent le tableau d'octets avec des caractères NULL pour séparer les deux entrées. Ou bien, juste la première entére si le type de données était erroné (comportement par défaut de Exif). La section COMPUTED va aussi contenir une entrée **Copyright** , qui sera soit la chaîne originale de copyright, soit une liste de valeur séparée par des virgules de photos et de copyright de l'auteur.

Note

La balise UserComment présente le même problème que la balise Copyright. Elle peut stocker deux valeurs : en premier, le jeu de caractères utilisé, puis la valeur elle−même. Si c'est le cas, la section IFD contiendra uniquement le jeu de caractères, ou bien un tableau d'octets. La section COMPUTED va stocker les deux entrées **UserCommentEncoding** et **UserComment** . L'index **UserComment** est disponbile dans les deux cas, et il est préférable de l'utiliser, plutôt que la valeur de la section IFD0.

Si les commentaires utilisateurs utilisent l'Unicode ou bien le JIS et que le module mbstring est accessible, ces jeux de caractères seront automatiquement changés en fonction la configuration de EXIF dans le fichier php.ini . Cette possibilité a été ajoutée en PHP 4.3.

Note

La hauteur (Height) et la largeur (Width) sont calculés de la même façon que la fonction getimagesize le fait, et leurs valeurs ne devraient jamais différer. De même, l'index html est la représentation textuelle utilisée dans une balise image HTML classique.

Note

Depuis PHP 4.3, cette fonction peut aussi lire les sections IFD incluent dans les tableaux, et retournée sous cette forme. De plus, la taille d'une miniature intégrée est retournée dans le sous tableau **THUMBNAIL** et la fonction exif_read_data peut retourner les miniatures au format TIFF . Enfin, il n'y a plus de longueur maximale pour les valeurs retournées (hormis la place en mémoire).

Note

Cette fonction est uniquement disponible en PHP 4, si il a été compilé avec l'option −−enable−exif . Ses fonctionnalités et comportement ont changés en PHP 4.2. Les premières versions étaient très instables.

Depuis PHP 4.3, les commentaires utilisateurs peuvent être automatiquement convertis d'un jeu de caractères à l'autre avec l'extension −−enable−mbstring .

Cette fonction ne requiert pas la librairie GD.

Voir aussi exif_thumbnail et getimagesize .

## 6.14.10 exif_thumbnail() : Récupère la miniature d'une image TIFF ou JPEG

string **exif_thumbnail** ( string filename , *int &width* , *int &height* , *int &imagetype* )

exif_thumbnail lit la miniature de l'image TIFF ou JPEG, dans le fichier filename . Si il n'y a pas de miniature, FALSE sera retourné.

Les paramètres width , height et imagetype sont disponibles depuis PHP 4.3 et retourne les dimensions de la miniature, ainsi que son type. Il est possible que exif_thumbnail ne puisse pas créer une image, mais en détermine la taille. Dans ce cas, la valeur retournée par la fonction est FALSE , mais width et height se voient assignés une valeur.

Si vous voulez exploiter les miniatures avec cette fonction, il faut nevoyer les en−têtes HTTP avec la fonction header . L'exemple ci−dessous montre comment faire :

Exemple d'utilisation de miniature avec exif_thumbnail

```
<?php
if (array_key_exists('file',$_REQUEST)) {
    $image = exif_thumbnail($_REQUEST['file'], $width, $height, $type);
} else {
    $image = false;
}
if ($image!==false) {
    header("Content−type: ".image_type_to_mime_type($type));
    echo $image;
    exit;
} else {
    // aucune miniature, gestion d'erreur
    echo "Pas de miniature disponible";
}
?>
```

Depuis PHP 4.3, la fonction exif_thumbnail lit aussi les miniatures dans les images au format TIFF .

Note

Cette fonction n'est disponible que si vous avez compilé PHP avec l'option −−enable−exif . Les fonctionnalités et le comportement de cette fonction ont changé en PHP 4.2

Cette fonction ne requiert pas la librairie GD.

Voir aussi exif_read_data et image_type_to_mime_type .

## 6.14.11 gd_info() : Retourne des informations à propos de la librairie GD installée

array **gd_info** ( void )

Retourne un tableau associatif décrivant la version et les possibilités de la librairie GD installée.

| Attribut | Signification |
|---|---|
| GD Version | chaîne de caractères décrivant la version de libgd qui est installée. |
| Freetype Support | booléen . TRUE si le support Freetype est installé. |
| Freetype Linkage | chaîne de caractères décrivant la façon de laquelle Freetype a été lié. Les valeurs attendues sont : 'with freetype', 'with TTF library', et 'with unknown library'. Cet élément ne sera défini que si Freetype Support est évalué TRUE . |
| T1Lib Support | booléen . TRUE si le support T1Lib est inclus. |
| GIF Read Support | booléen . TRUE si le support pour la **lecture** des images GIF est inclus. |
| GIF Create Support | booléen . TRUE si le support pour la **création** des images GIF est inclus. |
| JPG Support | booléen . TRUE si le support de JPG est inclus. |
| PNG Support | booléen . TRUE si le support de PNG est inclus. |
| WBMP Support | booléen . TRUE si le support de WBMP est inclus. |
| XBM Support | booléen . TRUE si le support de XBM est inclus. |

Utilisation de gd_info

```
<?php
var_dump(gd_info());

/* Sortie typique
   ==============

array(9) {
  ["GD Version"]=>
```

```
  string(24) "bundled (2.0 compatible)"
  ["FreeType Support"]=>
  bool(false)
  ["T1Lib Support"]=>
  bool(false)
  ["GIF Read Support"]=>
  bool(true)
  ["GIF Create Support"]=>
  bool(false)
  ["JPG Support"]=>
  bool(false)
  ["PNG Support"]=>
  bool(true)
  ["WBMP Support"]=>
  bool(true)
  ["XBM Support"]=>
  bool(false)
}

*/
?>
```

Voir aussi imagepng , imagejpeg , imagegif , imagewbmp , et imagetypes .

## 6.14.12 getimagesize() : Retourne la taille d'une image

array **getimagesize** ( string filename , *array imageinfo* )

getimagesize va déterminer la taille des images de type GIF , JPG , PNG , SWF , PSD ou BMP et en retourner les dimensions, le type d'image, et une chaîne type "height/width", à placer dans une balise HTML ou IMG normale.

Si l'accès au fichier filename est impossible, ou si ce n'est pas un format d'image supporté, getimagesize retournera FALSE et générera une alerte.

getimagesize retourne un tableau de 4 éléments. L'index 0 contient la longueur. L'index 1 contient la largeur. L'index 2 contient le type de l'image : 1 = GIF , 2 = JPG , 3 = PNG , 5 = PSD , 6 = BMP , 7 = TIFF (Ordre des octets Intel), 8 = TIFF (Ordre des octets Motorola), 9 = JPC , 10 = JP2 , 11 = JPX , 12 = JB2 , 13 = SWC , 14 = IFF . Ces valeurs correspondent aux constantes IMAGETYPE qui ont été ajoutée en PHP 4.3. L'index 3 contient la chaîne à placer dans les balises HTML : "height=xxx width=xxx" .

Exemple avec getimagesize

```
 <?php
list($width, $height, $type, $attr) = getimagesize("img/flag.jpg");
echo "<img src=\"img/flag.jpg\" $attr>";
?>>
```

getimagesize avec une URL

```
 <?php
  $size = getimagesize("http://www.php.net/gifs/logo.gif");
?>
```

Avec les images JPEG , deux en−têtes supplémentaires sont retournés : channel et bits . channel vaudra 3 avec les images RGB, et 4 avec les images CMYK. bits est le nombre de bits de chaque couleur.

Depuis PHP 4.3, bits et channels sont présent pour tous les types d'images. Cependant, la présence de ces valeurs peut mener à la confusion. Par exemple, une image GIF utilise toujours trois canaux par pixel, mais le nombre de bits par pixel ne peut être calculé dans le cas d'une image animée GIF ayant une table de couleur globale.

Certains formats peuvent ne contenir aucune image, ou bien plusieurs images. Dans ces cas, getimagesize peut ne pas être capable de déterminer correctement la taille de l'image. getimagesize retourne alors zéro comme taille de hauteur et largeur.

Depuis PHP 4.3, getimagesize retourne aussi un paramètre additionnel, mime , qui correspond au type MIME d'une image. Cette information peut être utilisée pour envoyer l'en−tête HTTP correct.

getimagesize et types MIME

```
<?php
$size = getimagesize ($filename);
$fp=fopen($filename, "rb");
if ($size && $fp) {
  header("Content-type: {$size['mime']}");
  fpassthru($fp);
  exit;
} else {
  // error
}
?>
```

Si l'accès à filename est impossible, ou si ce n'est pas une image valide, getimagesize retournera NULL et générera une alerte.

Le paramètre optionnel imageinfo permet d'extraire des informations supplémentaires du fichier image. Actuellement, cette option va retourner différents marqueurs JPG APP dans un tableau associatif. Certains programmes utilisent ces marqueur APP pour préciser les informations dans les balises HTML. Un marqueur commun est le marqueur APP13, décrit à http://www.iptc.org/ . Vous pouvez utiliser la fonction iptcparse pour analyser ce marqueur, et obtenir des informations intelligibles.

getimagesize qui retourne IPTC

```
<?php
 $size = getimagesize("testimg.jpg", $info);
 if (isset($info["APP13"])) {
   $iptc = iptcparse($info["APP13"]);
   var_dump($iptc);
 }
?>
```

Note

getimagesize ne requiert pas la bibliothèque GD.

Le support URL a été ajouté en PHP 4.0.5.

Voir aussi image_type_to_mime_type , exif_imagetype , exif_read_data et exif_thumbnail .

Le support du format TIFF a été ajouté en PHP 4.2.

## 6.14.13 image_type_to_mime_type() : Lit le Mime−Type d'un type d'image

string **image_type_to_mime_type** ( int imagetype )

image_type_to_mime_type détermine le type Mime (Mime−Type), à utiliser dans l'en−tête HTTP Content−type . imagetype est une constante, représentant un type d'image, et retournée par getimagesize , exif_read_data , exif_thumbnail ou exif_imagetype .

image_type_to_mime_type (file)

```
<?php
header ("Content-type: ".image_type_to_mime_type (IMAGETYPE_PNG));
?>
```

Note

Cette fonction ne requiert par la librairie GD.

Voir ausi getimagesize , exif_imagetype , exif_read_data et exif_thumbnail .

## 6.14.14 image2wbmp() : Crée une image WBMP

int **image2wbmp** ( resource image , *string filename* , *int threshold* )

image2wbmp crée une image WBMP à partir de l'image image . Si le paramètre filename est fourni, l'image sera créée dans ce fichier, sinon, elle sera envoyée au navigateur. image est une image valide, créée avec la fonction imagecreate .

Le nom de fichier filename est optionnel, et si il est omis, l'image sera renvoyée directement au navigateur. En retournant un en−tête HTTP Content−Type : IMAGE/VND.WAP.WBMP avec la fonction header , vous pouvez créer des images WBMP avec vos scripts PHP.

Exemple avec image2wbmp

```
<?php

 $file = 'php.jpg';

 header('Content-type: ' . image_type_to_mime_type(IMAGETYPE_WBMP));
 image2wbmp($file); // output the stream directly

?>
```

Note

Le support WBMP n'est disponible qu'avec GD−1.8 ou plus récent.

Voir aussi imagewbmp .

## 6.14.15 imagealphablending() : Modifie le mode de blending d'une image

bool **imagealphablending** ( resource image , bool blendmode )

imagealphablending fournit deux modes de dessins des images en vraies couleurs (truecolors). En mode "blending", le canal alpha de chaque couleur est fournie à chaque fonction de dessin, tel que imagesetpixel peut déterminer sa transparence. GD va alors automatiquement mixer la couleur à ce point, et stocker le résultat dans l'image. Le pixel résultant est alors opaque. En mode non−mixant, la couleur est copiée littéralement avec ses informations de canal alpha, et remplace le pixel de destination. Le mixage n'est pas disponible avec les images à palette. Si blendmode vaut TRUE , alors le mode de mixage sera activé, sinon il sera désactivé.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

imagealphablending a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

## 6.14.16 imageantialias() : Utiliser ou non les fonctions d'antialias

bool **imageantialias** ( resource im , bool on )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

Voir aussi imagecreatetruecolor .

## 6.14.17 imagearc() : Dessine une ellipse partielle

int **imagearc** ( resource image , int cx , int cy , int w , int h , int s , int e , int color )

imagearc dessine une ellipse partielle, centrée sur cx , cy (le coin en haut à gauche est l'origine (0,0)) dans l'image référencée par image . w et h spécifient la largeur et la hauteur de l'ellipse, tandis que le début et la fin de l'arc sont donnés en degrés, par les arguments s et e . color est la couleur utilisée pour dessiner l'ellipse.

Dessin d'un cercle avec imagearc

```
<?php

// Création d'une image 200*200
$img = imagecreate(200, 200);

// Allocation de couleurs
$white = imagecolorallocate($img, 255, 255, 255);

// Dessin d'un cercle blanc
imagearc($img, 100, 100, 150, 150, 0, 360, $white);

// Affichage au navigateur
header("Content-type: image/png");
imagepng($img);

// Libération de la mémoire
imagedestroy($img);

?>
```

Voir aussi imageellipse , imagefilledellipse et imagefilledarc .

## 6.14.18 imagechar() : Dessine un caractère horizontalement

int **imagechar** ( resource image , int font , int x , int y , string c , int color )

imagechar dessine le premier caractère de la chaîne c dans l'image id avec le coin supérieur gauche placé à la position x , y (le coin en haut à gauche est l'origine (0,0)) avec la couleur color . Si la police est 1, 2, 3, 4 ou 5, une police intégrée sera utilisée (plus le chiffre est grand, plus grande est la police).

Voir aussi imageloadfont .

## 6.14.19 imagecharup() : Dessine un caractère verticalement

int **imagecharup** ( resource image , int font , int x , int y , string c , int color )

imagecharup dessine le premier caractère de la chaîne c dans l'image id avec le coin supérieur gauche placé à la position ( x , y ) (le coin en haut à gauche est l'origine (0,0)), avec la couleur color . Si la police est 1, 2, 3, 4 ou 5, une police intégrée sera utilisée (plus le chiffre est grand, plus grande est la police).

Voir aussi imageloadfont .

## 6.14.20 imagecolorallocate() : Alloue une couleur pour une image

int **imagecolorallocate** ( resource image , int red , int green , int blue )

imagecolorallocate retourne un identifiant de couleur, représentant la couleur composée avec les couleurs RGB ( red , green , blue ). L'argument image est le résultat de la fonction imagecreate . imagecolorallocate doit être appelée pour créer chaque couleur qui sera représentée par image .

Exemple avec imagecolorallocate

```
<?php
 $white = imagecolorallocate($im, 255,255,255);
 $black = imagecolorallocate($im, 0,0,0);
?>
```

Voir aussi imagecolorallocatealpha et imagecolordeallocate .

## 6.14.21 imagecolorallocatealpha() : Alloue une couleur à une image

int **imagecolorallocatealpha** ( resource image , int red , int green , int blue , int alpha )

imagecolorallocatealpha se comporte comme imagecolorallocate avec en plus le paramètre de transparence alpha qui prend une valeur entre 0 et 127 . 0 indique une opacité complète tandis que 127 indique une transparence complète.

imagecolorallocatealpha retourne FALSE si l'allocation a échoué.

Voir aussi imagecolorallocate et imagecolordeallocate .

## 6.14.22 imagecolorat() : Retourne l'index de la couleur d'un pixel donné

int **imagecolorat** ( resource image , int x , int y )

imagecolorat retourne l'index de la couleur du pixel situé aux coordonnées ( x , y ), dans l'image image .

Si PHP supporte la librairie GD library 2.0 ou plus récent, et que l'image est une image en truecolor, cette fonction retourne la valeur RGB du pixel, sous forme d'un entier. Utiliser les opérateurs de bits et les masques pour distinguer le rouge, du vert du bleu :

Accès aux valeurs RGB

```
<?php
$im = ImageCreateFromPng("rockym.png");
$rgb = ImageColorAt($im, 100, 100);
$r = ($rgb >> 16) & 0xFF;
$g = ($rgb >> 8) & 0xFF;
$b = $rgb & 0xFF;
?>
```

Voir aussi imagecolorset et imagecolorsforindex .

## 6.14.23 imagecolorclosest() : Retourne l'index de la couleur la plus proche d'une couleur donnée.

int **imagecolorclosest** ( resource image , int red , int green , int blue )

imagecolorclosest retourne l'index de la couleur de la palette qui est la plus proche de la valeur RGB passée.

La "distance" entre la couleur souhaitée et les couleurs de la palette est calculée en considérant l'espace RGB comme un espace à 3 dimensions.

Voir aussi imagecolorexact .

## 6.14.24 imagecolorclosestalpha() : Retourne la couleur la plus proche, en tenant compte du canal alpha

int **imagecolorclosestalpha** ( resource image , int red , int green , int blue , int alpha )

imagecolorclosestalpha retourne l'index de la couleur, dans la palette de l'image image , la plus proche de la couleur spécifiée par les autres paramètres, au format RGB et de canal alpha alpha .

Voir aussi imagecolorexactalpha .

| Note |
| --- |
| imagecolorclosestalpha a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1. |

## 6.14.25 imagecolorclosesthwb() : Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir (hue, white and blackness)

int **imagecolorclosesthwb** ( resource image , int red , int green , int blue )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.14.26 imagecolordeallocate() : Supprime une couleur d'une image

int **imagecolordeallocate** ( resource image , int color )

imagecolordeallocate désalloue la couleur color précédemment allouée avec la fonction imagecolorallocate , pour l'image image .

Exemple avec imagecolordeallocate

```
<?php
 $white = imagecolorallocate($im, 255, 255, 255);
 imagecolordeallocate($im, $white);
?>
```

Voir aussi imagecolorallocate et imagecolorallocatealpha .

## 6.14.27 imagecolorexact() : Retourne l'index de la couleur donnée

int **imagecolorexact** ( resource image , int red , int green , int blue )

imagecolorexact retourne l'index de la couleur spécifiée dans la palette de l'image image .

Si la couleur n'existe pas dans cette palette, imagecolorexact retourne −1.

Voir aussi imagecolorclosest .

## 6.14.28 imagecolorexactalpha() : Retourne l'index d'une couleur avec son canal alpha

int **imagecolorexactalpha** ( resource image , int red , int green , int blue , int alpha )

imagecolorexactalpha retourne l'index de la couleur fournie au format RGB et son canal alpha alpha , dans l'image image .

Si la couleur n'existe pas dans la palette de l'image, imagecolorexactalpha retourne −1.

Voir aussi imagecolorclosestalpha .

## 6.14.29 imagecolormatch() : Fait correspondre un peu plus les couleurs de la version palette d'une image aux couleurs de sa version truecolor

bool **imagecolormatch** ( resource image1 , resource image2 )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

image1 doit être Truecolor, image2 doit être Palette, et image1 et image2 doivent avoir la même taille.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi imagecreatetruecolor .

## 6.14.30 imagecolorresolve() : Retourne l'index de la couleur donnée, ou la plus proche possible.

int **imagecolorresolve** ( resource image , int red , int green , int blue )

imagecolorresolve retourne un index de couleur à tous les coups. Soit il arrive à trouver la couleur demandée dans la palette, soit il recherche la couleur la plus proche.

Voir aussi imagecolorclosest .

## 6.14.31 imagecolorresolvealpha() : Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha

int **imagecolorresolvealpha** ( resource image , int red , int green , int blue , int alpha )

imagecolorresolvealpha retourne toujours un index de couleur, disponible dans la palette de l'image image : soit c'est la couleur exacte, soit c'est la meilleure approximation.

Voir aussi imagecolorclosestalpha .

| Note |
| --- |
| imagecolorresolvealpha a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1. |

## 6.14.32 imagecolorset() : Change la couleur dans une palette à l'index donné.

bool **imagecolorset** ( resource image , int index , int red , int green , int blue )

imagecolorset permet d'attribuer à un index d'une palette une couleur spécifique. C'est une fonction très pratique pour effectuer du remplissage de couleur sans le faire réellement.

Voir aussi imagecolorat .

## 6.14.33 imagecolorsforindex() : Retourne la couleur associée à un index

array **imagecolorsforindex** ( resource image , int index )

imagecolorsforindex retourne un tableau associatif avec les couleurs rouge (red) , vert (green), bleu (blue) qui contiennent les valeurs de la couleur correspondante.

Voir aussi imagecolorat et imagecolorexact .

## 6.14.34 imagecolorstotal() : Calcule le nombre de couleurs d'une palette

int **imagecolorstotal** ( resource image )

imagecolorstotal retourne le nombre de couleurs de la palette pour l'image image .

Voir aussi imagecolorat et imagecolorsforindex .

## 6.14.35 imagecolortransparent() : Définit la couleur transparente

int **imagecolortransparent** ( resource image , *int color* )

imagecolortransparent permet de choisir la couleur transparente d'une image, et de lui donner la valeur de color . image est un identifiant d'image, retourné par imagecreate et color est un identifiant de couleur retourné par imagecolorallocate .

L'identifiant de la nouvelle (ou courante) couleur transparente est retourné.

## 6.14.36 imagecopy() : Copie une partie d'une image

int **imagecopy** ( resource dst_im , resource src_im , int dst_x , int dst_y , int src_x , int src_y , int src_w , int src_h )

Copie une partie de l'image src_im sur l'image de destination dst_im , en commençant aux coordonnées src_x , src_y et sur la largeur de src_w et la hauteur de src_h . La portion ainsi définie sera copiée et placée aux coordonnées dst_x et dst_y .

## 6.14.37 imagecopymerge() : Copie et fusionne une partie d'une image

int **imagecopymerge** ( resource dst_im , resource src_im , int dst_x , int dst_y , int src_x , int src_y , int src_w , int src_h , int pct )

imagecopymerge copie une partie de l'image src_im dans l'image de destination dst_im en commençant aux coordonnées ( src_x , src_y ), avec la largeur src_w et la hauteur src_h . La zone de l'image ainsi définie sera copiée aux coordonnées ( dst_x , dst_y ), dans l'image de destination. Les deux images seront fusionnées suivant le paramètre pct , qui peut valoir de 0 à 100. Si pct = 0, aucune action n'est faite, alors que si pct = 100, imagecopymerge se comporte exactement comme imagecopy .

| Note |
| --- |
| imagecopymerge a été ajoutée en PHP 4.0.6. |

## 6.14.38 imagecopymergegray() : Copie et fusionne une partie d'une image en niveaux de gris

int **imagecopymergegray** ( resource dst_im , resource src_im , int dst_x , int dst_y , int src_x , int src_y , int src_w , int src_h , int pct )

imagecopymergegray copie une partie de l'image src_im dans l'image de destination dst_im commençant aux coordonnées ( src_x , src_y ), avec la largeur src_w et la hauteur src_h . La zone de l'image ainsi définie sera copiée aux coordonnées ( dst_x , dst_y ), dans l'image de destination. Les deux images seront fusionnées suivant le paramètre pct , qui peut valoir de 0 à 100. Si pct = 0, aucune action n'est faite, alors que si pct = 100, imagecopymerge se comporte exactement comme imagecopy .

imagecopymergegray est identique à la fonction imagecopymerge , hormis le fait que lors de la fusion, le "hue" de l'image sera conservé grâce à la conversion de la zone dans l'image de destination en gris, avant l'opération de copie.

| Note |
| --- |
| imagecopymergegray a été ajoutée en PHP 4.0.6. |

## 6.14.39 imagecopyresampled() : Copie, redimensionne, rééchantillonne une image

bool **imagecopyresampled** ( resource dst_im , resource src_im , int dstx , int dsty , int srcx , int srcy , int dstw , int dsth , int srcw , int srch )

imagecopyresampled copie une zone rectangulaire de l'image src_im vers l'image dst_im . Durant la copie, la zone est rééchantillonnée de manière à conserver la clarté de l'image durant une réduction. dst_im est l'image de destination, src_im est l'image source. Si les hauteurs et largeur des source et destination diffèrent, l'image copiée sera étirée de manière appropriée. Les coordonnées sont celles du coin supérieur gauche. imagecopyresampled peut servir à copier des zones d'une image vers elle−même, mais si les régions se chevauchent, les résultats sont imprévisibles.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Il y a un problème du aux limitations de la taille de la palette (255 + 1 couleurs différentes). Filtrer ou rééchantillonner une image demande plus de 255 couleurs, une approximation est alors utilisée pour calculer le nouveau nombre de couleurs. Avec une palette, si une nouvelle couleur ne peut être alloueé, la couleur la plus proche (en théorie) est utilisée. Ce n'est pas toujours la couleur la plus proche visuellement. Cela peut générer des problèmes étranges, comme des images blanches. Pour éviter ce problème, passez en image true color, comme celles générée par la fonction imagecreatetruecolor .

Note

imagecopyresampled a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

Voir aussi imagecopyresized .

## 6.14.40 imagecopyresized() : Copie et redimensionne une partie d'une image

int **imagecopyresized** ( resource dst_im , resource src_im , int dstx , int dsty , int srcx , int srcy , int dstw , int dsth , int srcw , int srch )

imagecopyresized copie une partie rectangulaire d'une image dans une autre image de destination. dst_im est l'image de destination, src_im est l'image source. Si les dimensions de la source et de la destination ne sont pas égales, un étirement adéquat est effectué pour faire correspondre les deux. Les coordonnées fournies sont définies par rapport au coin supérieur gauche. Cette fonction peut être utilisée pour recopier des régions à l'intérieur d'une même image, si dst_im et src_im sont identiques : mais si les régions se chevauchent, le résultat risque d'être incohérent.

Note

Il y a un problème du aux limitations de la taille de la palette (255 + 1 couleurs différentes). Filtrer ou rééchantillonner une image demande plus de 255 couleurs, une approximation est alors utilisée pour calculer le nouveau nombre de couleurs. Avec une palette, si une nouvelle couleur ne peut être alloueé, la couleur la plus proche (en théorie) est utilisée. Ce n'est pas toujours la couleur la plus proche visuellement. Cela peut générer des problèmes étranges, comme des images blanches. Pour éviter ce problème, passez en image true color, comme celles générée par la fonction imagecreatetruecolor .

Voir aussi imagecopyresampled .

## 6.14.41 imagecreate() : Crée une nouvelle image à palette

resource **imagecreate** ( int x_size , int y_size )

imagecreate retourne un identifiant d'image représentant une image vide, de largeur x_size et longueur y_size .

Nous recommandons l'utilisation de imagecreatetruecolor à la place de imagecreate .

Création d'une image GD et affichage de cette image

```
<?php
 header ("Content-type: image/png");
 $im = @imagecreate (50, 100)
   or die ("Impossible d'initialiser la librairie GD");
 $background_color = imagecolorallocate ($im, 255, 255, 255);
 $text_color = imagecolorallocate ($im, 233, 14, 91);
 imagestring ($im, 1, 5, 5,  "A Simple Text String", $text_color);
```

```
   imagepng ($im);
?>
```

Voir aussi imagedestroy et imagecreatetruecolor .

## 6.14.42 imagecreatefromgd() : Crée une nouvelle image à partir d'un fichier GD ou d'une URL

resource **imagecreatefromgd** ( string filename )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Array

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

## 6.14.43 imagecreatefromgd2() : Crée une nouvelle image à partir d'un fichier GD2 ou d'une URL

resource **imagecreatefromgd2** ( string filename )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Array

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

## 6.14.44 imagecreatefromgd2part() : Crée une nouvelle image à partir d'une partie de fichier GD2 ou d'une URL

resource **imagecreatefromgd2part** ( string filename , int srcx , int srcy , int width , int height )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Array

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

## 6.14.45 imagecreatefromgif() : Crée une nouvelle image à partir d'un fichier ou d'une URL.

resource **imagecreatefromgif** ( string filename )

imagecreatefromgif retourne un identifiant d'image qui représente l'image obtenue à partir du fichier dont le nom est donné.

imagecreatefromgif retourne une chaîne vide en cas d'échec. Il va aussi retourner une erreur qui va afficher un lien brisé dans un navigateur. Pour simplifier le débuggage, utilisez le code suivant, qui retourne une erreur GIF :

Exemple de gestion des erreurs durant la création d'image (gracieusement offert par vic@zymsys.com )

```
<?php
function LoadGif ($imgname) {
    $im = @imagecreatefromgif ($imgname); /* Tentative d'ouverture */
    if (!$im) { /* Test d'échec */
        $im = imagecreate (150, 30); /* Création d'une image vide */
        $bgc = imagecolorallocate ($im, 255, 255, 255);
        $tc = imagecolorallocate ($im, 0, 0, 0);
        imagefilledrectangle ($im, 0, 0, 150, 30, $bgc);
        /* Affichage d'un message d'erreur */
        imagestring ($im, 1, 5, 5, "Erreur au chargement de l'image $imgname", $tc);
    }
    return $im;
}
?>
```

Note

Etant donné que toutes les fonctions de gestion des GIF ont été supprimées de la bibliothèque GD version 1.6, cette fonction n'est pas disponible si vous utilisez cette version de la librairie.

Array

Attention

La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

## 6.14.46 imagecreatefromjpeg() : Crée une nouvelle image JPEG à partir d'un fichier ou d'une URL

resource **imagecreatefromjpeg** ( string filename )

imagecreatefromjpeg retourne un identifiant d'image représentant une image obtenue à partir du fichier filename .

imagecreatefromjpeg retourne une chaîne vide en cas d'échec. Elle affiche aussi un message d'erreur, qui s'affiche comme un lien brisé dans un navigateur web. Pour faciliter le débuggage, voici une erreur JPEG :

Exemple de gestion d'erreur lors de la création d'image (gracieusement offert par vic@zymsys.com )

```
<?php
function loadjpeg($imgname) {
  $im = @imagecreatefromjpeg($imgname); /* Tentative d'ouverture */
  if (!$im) { /* Vérification */
    $im = imagecreate(150, 30); /* Création d'une image blanche */
    $bgc = imagecolorallocate($im, 255, 255, 255);
    $tc  = imagecolorallocate($im, 0, 0, 0);
    imagefilledrectangle($im, 0, 0, 150, 30, $bgc);
// Affichage d'un message d'erreur
    imagestring($im, 1, 5, 5, "Erreur de chargement de l'image $imgname", $tc);
  }
  return $im;
}
?>
```

Array

Attention

La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

## 6.14.47 imagecreatefrompng() : Crée une nouvelle image PNG à partir d'un fichier ou d'une URL

resource **imagecreatefrompng** ( string filename )

imagecreatefrompng retourne un identifiant d'image représentant une image obtenue à partir du fichier filename .

imagecreatefromjpeg retourne une chaîne vide en cas d'échec. Elle affiche aussi un message d'erreur, qui s'affiche comme un lien brisé dans un navigateur web. Pour faciliter le débuggage, voici une erreur PNG :

Exemple de gestion d'erreur lors de la création d'image (gracieusement offert par vic@zymsys.com )

```
<?php
function LoadPNG($imgname) {
  $im = @imagecreatefrompng($imgname); /* Tentative d'ouverture */
  if (!$im) { /* Vérification */
    $im = imagecreate(150, 30); /* Création d'une image blanche */
        $bgc = imagecolorallocate($im, 255, 255, 255);
        $tc  = imagecolorallocate($im, 0, 0, 0);
        imagefilledrectangle($im, 0, 0, 150, 30, $bgc);
        /* Affichage d'un message d'erreur */
        imagestring($im, 1, 5, 5, "Erreur de chargement de l'image $imgname", $tc);
    }
    return $im;
}
?>
```

Array

Attention

La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé.

## 6.14.48 imagecreatefromstring() : Crée une image à partir d'une chaîne

resource **imagecreatefromstring** ( string image )

imagecreatefromstring retourne un identifiant d'image représentant la chaîne image .

## 6.14.49 imagecreatefromwbmp() : Crée une image depuis un fichier WBMP

resource **imagecreatefromwbmp** ( string filename )

imagecreatefromwbmp retourne une ressource d'image PHP, représentant l'image filename .

imagecreatefromwbmp retourne une chaîne vide en cas d'erreur. Il retourne aussi un message d'erreur qui s'affiche comme un lien mort dans un navigateur. Pour aider au débuggage, l'exemple suivant va produire une erreur WBMP :

Exemple de gestion des erreurs durant la création d'une image WBMP (gracieusement proposé par vic@zymsys.com)

```
<?php
function loadwbmp($imgname) {
  $im = @imagecreatefromwbmp($imgname); /* Tentative d'ouverture */
  if (!$im) { /* Vérification que cela s'est bien passé */
    $im  = imagecreate(20, 20); /* Crée une image blanche */
    $bgc = imagecolorallocate($im, 255, 255, 255);
    $tc  = imagecolorallocate($im, 0, 0, 0);
    imagefilledrectangle($im, 0, 0, 10, 10, $bgc);
// Affiche le message d'erreur
    imagestring($im, 1, 5, 5, "Erreur de chargement de $imgname", $tc);
  }
```

```
  return $im;
}
?>
```

| Note |
| --- |
| Le support WBMP n'est disponible qu'avec GD−1.8 ou plus récent. |

Array

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

## 6.14.50 imagecreatefromxbm() : Crée une image à partir d'un fichier XBM

resource **imagecreatefromxbm** ( string filename )

imagecreatefromxbm retourne un identifiant d'image représentant l'image obtenue à partir du fichier XBM filename .

Array

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

## 6.14.51 imagecreatefromxpm() : Crée une image à partir d'un fichier XPM

resource **imagecreatefromxpm** ( string filename )

imagecreatefromxpm retourne un identifiant d'image représentant l'image obtenue à partir du fichier XPM filename .

Array

| Attention |
| --- |
| La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. |

## 6.14.52 imagecreatetruecolor() : Crée une nouvelle image en vraies couleurs

resource **imagecreatetruecolor** ( int x_size , int y_size )

imagecreatetruecolor retourne une ressource représentant une image noire de largeur x_size , et de hauteur y_size .

Création d'un flux d'image GD, et affichage

```
 <?php
header ("Content−type: image/png");
$im = @imagecreatetruecolor (50, 100)
      or die ("Impossible de crée un flux d'image GD");
$text_color = imagecolorallocate ($im, 233, 14, 91);
imagestring ($im, 1, 5, 5,  "Une simple chaîne de texte", $text_color);
imagepng ($im);
imagedestroy ($im);
?>
```

| Note |
| --- |
| imagecreatetruecolor a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1. |

| Note |
| --- |
| imagecreatetruecolor ne fonctionne pas avec le format GIF . |

Voir aussi imagedestroy et imagecreate .

## 6.14.53 imagedashedline() : Dessine une ligne pointillée

int **imagedashedline** ( resource image , int x1 , int y1 , int x2 , int y2 , int color )

imagedashedline est obsolète. UTilisez plutôt une combinaison des fonctions imagesetstyle et imageline à la place.

## 6.14.54 imagedestroy() : détruit une image

int **imagedestroy** ( resource image )

imagedestroy libère toute la mémoire associée à l'image image . image est un identifiant d'image valide retourné par imagecreate .

## 6.14.55 imageellipse() : Dessine une ellipse

int **imageellipse** ( resource image , int cx , int cy , int w , int h , int color )

imageellipse dessine une ellipse centrée sur le point ( cx , cy ). Le coin supérieur gauche est aux coordonnées (0,0). L'image de dessin est image . w et h spécifient respectivement la largeur et la hauteur de l'ellipse. La couleur de dessin de l'ellipse est color .

Exemple avec imageellipse

```
<?php

// nouvelle image
$image = imagecreate(400,300);

// remplissage du fond
$bg = imagecolorallocate($image,0,0,0);

// couleur de l'ellipse
$col_ellipse = imagecolorallocate($image,255,255,255);

// on dessine l'ellipse
imageellipse($image, 200, 150, 300, 200, $col_ellipse);

// on affiche l'image
header("Content-type: image/png");
imagepng($image);

?>
```

| Note |
| --- |
| imageellipse a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1. |

Voir aussi imagefilledellipse et imagearc .

## 6.14.56 imagefill() : Remplit

int **imagefill** ( resource image , int x , int y , int color )

imagefill effectue un remplissage avec la couleur color , dans l'image image , à partir du point de coordonnées ( x , y ) (le coin supérieur gauche est l'origine (0,0)).

## 6.14.57 imagefilledarc() : Dessine une ellipse partielle et la remplit

bool **imagefilledarc** ( resource image , int cx , int cy , int w , int h , int s , int e , int color , int style )

imagefilledarc dessine une ellipse partielle, centrée sur le point ( cx , cy ). Le coin supérieur gauche est (0, 0), dans l'image image . w et h spécifient respectivement la largeur et la hauteur de l'ellipse, tandis que les points de début et de fin sont représentés par s et e , en degrés. L'argument style est un champ de bits, combiné avec l'opérateur OR :

1. IMG_ARC_PIE
2. IMG_ARC_CHORD
3. IMG_ARC_NOFILL
4. IMG_ARC_EDGED

IMG_ARC_PIE et IMG_ARC_CHORD sont mutellemement exclusives; IMG_ARC_CHORD ne fait que connecter les angles de début et de fin avec une ligne droite, tandis que IMG_ARC_PIE produit une ligne courbe. IMG_ARC_NOFILL indique que l'arc (ou corde) doit être dessiné mais pas rempli. IMG_ARC_EDGED , utilisé conjointement avec IMG_ARC_NOFILL , indique que les angles de début et de fin doivent être connectés au centre. Cette fonction est recommandée pour faire les graphiques de type camembert.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| imagefilledarc a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1. |

## 6.14.58 imagefilledellipse() : Dessine une ellipse pleine

bool **imagefilledellipse** ( resource image , int cx , int cy , int w , int h , int color )

imagefilledellipse dessine une ellipse centrée sur le point ( cx , cy ). Le coin supérieur gauche est aux coordonnées (0,0). L'image de dessin est image . w et h spécifient respectivement la largeur et la hauteur de l'ellipse. La couleur de remplissage de l'ellipse est color .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec imagefilledellipse

```
 <?php

// nouvelle image
$image = imagecreate(400,300);

// couleur de fond
$bg = imagecolorallocate($image,0,0,0);

// couleur de remplissage de l'ellipse
$col_ellipse = imagecolorallocate($image,255,255,255);

// on dessine l'ellipse blanche
```

```
imagefilledellipse($image, 200, 150, 300, 200, $col_ellipse);

// on affiche l'image
header("Content-type: image/png");
imagepng($image);

?>
```

| Note |
| --- |
| imagefilledellipse a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1. |

Voir aussi imagefilledarc .

## 6.14.59 imagefilledpolygon() : Dessine un polygone rempli

int **imagefilledpolygon** ( resource image , array points , int num_points , int color )

imagefilledpolygon dessine un polygone rempli dans l'image image . points est un tableau PHP qui contient les sommets des polygones sous la forme :. points[0] = x0, points[1] = y0, points[2] = x1, points[3] = y1, etc. num_points est le nombre total de sommets.

## 6.14.60 imagefilledrectangle() : Dessine un rectangle rempli

int **imagefilledrectangle** ( resource image , int x1 , int y1 , int x2 , int y2 , int color )

imagefilledrectangle dessine un rectangle de couleur color dans l'image image , en commençant par le sommet supérieur gauche ( x1 , y1 ) et finissant au sommet inférieur droit ( x2 , y2 ). Le coin supérieur gauche est l'origine (0, 0).

## 6.14.61 imagefilltoborder() : Remplit avec une région avec une couleur spécifique

int **imagefilltoborder** ( resource image , int x , int y , int border , int color )

imagefilltoborder remplit avec la couleur color toute la région à l'intérieur de la région limitée par la couleur border . Le point de départ est ( x , y ) (le coin supérieur gauche est l'origine (0,0)).

## 6.14.62 imagefontheight() : Retourne la hauteur de la police

int **imagefontheight** ( int font )

imagefontheight retourne la hauteur de la police font en pixels.

Voir aussi imagefontwidth et imageloadfont .

## 6.14.63 imagefontwidth() : Retourne la largeur de la police

int **imagefontwidth** ( int font )

imagefontwidth retourne la largeur de la police font en pixels.

Voir aussi imagefontheight et imageloadfont .

## 6.14.64 imageftbbox() : Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2

array **imageftbbox** ( int size , int angle , string font_file , string text , *array extrainfo* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.14.65 imagefttext() : Ecrit du texte dans une image avec la police courante et FreeType 2

array **imagefttext** ( resource image , int size , int angle , int x , int y , int col , string font_file , string text , *array extrainfo* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.14.66 imagegammacorrect() : Applique une correction gamma à l'image

int **imagegammacorrect** ( resource image , float inputgamma , float outputgamma )

imagegammacorrect applique une correction gamma à l'image GD image . Le facteur d'entrée est inputgamma , et le facteur de sortie est outputgamma .

## 6.14.67 imagegd() : Génère une image au format GD, vers le navigateur ou un fichier

int **imagegd** ( resource image , *string filename* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.14.68 imagegd2() : Génère une image au format GD2, vers le navigateur ou un fichier

int **imagegd2** ( resource image , *string filename* , *int chunk_size* , *int type* )

imagegd2 génère une image au format GD2, vers le navigateur ou un fichier.

Le paramètre optionnel type vaut soit raw (brut) soit compressed (compressé).

| Note |
| --- |
| Les paramètres optionnels chunk_size et type ont été ajouté en PHP 4.3.1. |

## 6.14.69 imagegif() : Envoie une image GIF vers un navigateur ou un fichier

int **imagegif** ( resource image , *string filename* )

imagegif crée un fichier image GIF avec le nom filename d'après l'image image . L'argument image est un identifiant valide retourné par la fonction imagecreate .

Le format de l'image sera GIF87a, à moins que l'image n'ait une couleur transparente (mise en place grâce à la fonction imagecolortransparent )), ce qui fera qu'elle sera au format GIF89a.

Le nom du fichier est optionnel, et dans ce cas, l'image sera transmise directement à la sortie standard. En envoyant une en−tête de type image/gifcontent−type , (grâce à la fonction header ), vous pouvez créer des images avec des scripts PHP.

Note

Etant donné que toutes les fonctions GIF ont été supprimées de la bibliothèque GD version 1.6, cette fonction ne sera pas accessible si vous avez cette version de la librairie.

Le code suivant vous permet d'écrire des scripts PHP plus portables : le type de GD est automatiquement détecté. Il remplace la séquence Header("Content−type: image/gif"); ImageGif($im); par un code plus souple :

Détection automatique des formats d'image supportés

```
<?php
 if (function_exists("imagegif")) {
   header("Content-type: image/gif");
   imagegif($im);
 }
 elseif (function_exists("imagejpeg")) {
   header("Content-type: image/jpeg");
   imagejpeg($im, "", 0.5);
 }
 elseif (function_exists("imagepng")) {
   header("Content-type: image/png");
   imagepng($im);
 } elseif (function_exists("imagewbmp")) {
   header("Content-type: image/vnd.wap.wbmp");
   imagewbmp($im);
 }  else {
   die("Pas de support graphique avec PHP sur ce serveur");
 }
?>
```

Note

Depuis les versions 3.0.18 et 4.0.2, vous pouvez utiliser la fonction imagetypes au lieu de function_exists pour vérifier la présence des différents formats d'images supportés.

Détection automatique des formats d'image supportés avec imagetypes

```
 if (imagetypes() & IMG_GIF) {
    header ("Content-type: image/gif");
    imagegif ($im);
}
elseif (imagetypes() & IMG_JPG) {
        ... etc.
```

Voir aussi imagepng , imagewbmp , imagejpeg et imagetypes .

## 6.14.70 imageinterlace() : Active ou désactive l'entrelacement

int **imageinterlace** ( resource image , *int interlace* )

imageinterlace active ou désactive le bit d'entrelacement.

Si l'entrelacement est à 1, l'image image sera interlacée, et sinon, elle ne le sera pas. Si le format d'affichage de l'image est JPEG , l'image créée sera un JPEG progressif.

imageinterlace retourne l'état courant d'entrelacement de l'image.

## 6.14.71 imageistruecolor() : Détermine si une image est une image truecolor.

bool **imageistruecolor** ( resource image )

imageistruecolor détermine si l'image image est une image truecolor.

Voir aussi imagecreatetruecolor .

## 6.14.72 imagejpeg() : Envoie une image JPEG vers un navigateur ou un fichier.

int **imagejpeg** ( resource image , *string filename* , *int quality* )

imagejpeg envoie l'image GD ( image ) au format JPEG sur la sortie standard (typiquement, le navigateur web), ou si filename est fourni, l'envoie dans un fichier. image a été créé par imagecreate .

Le nom du fichier est optionnel, et dans ce cas, l'image sera transmise directement à la sortie standard. En envoyant une image de type image/jpeg content−type (grâce à la fonction header ), vous pouvez créer des images JPEG avec des scripts PHP.

| Note |
| --- |
| Le support JPEG n'est disponible que si PHP est compilé avec GD−1.8 ou plus récent. |

quality est optionnel, et prend des valeurs entières de 0 (pire qualité, petit fichier) et 100 (meilleure qualité, gros fichier). Par défaut, la valeur est à 100.

Si vous voulez générer des images JPEG progressive, vous devez choisir l'entrelacement à l'aide de la fonction imageinterlace .

Voir aussi imagepng , imagewbmp , imagegif , imageinterlace , imagegif et imagetypes .

## 6.14.73 imageline() : Dessine une ligne

int **imageline** ( resource image , int x1 , int y1 , int x2 , int y2 , int color )

imageline dessine une ligne depuis le point ( x1 , y1 ) jusqu'au point ( x2 , y2 ) (le coin supérieur gauche est l'origine (0,0)) dans l'image image et avec la couleur color .

Voir aussi imagecreate et imagecolorallocate .

## 6.14.74 imageloadfont() : Charge une nouvelle police

int **imageloadfont** ( string file )

imageloadfont charge une nouvelle police utilisateur et retourne un identifiant sur cette police. Cet identifiant sera toujours supérieur à 5, pour éviter les conflits avec les polices standard PHP.

Le format des polices dépend actuellement du système d'exploitation. Ce qui signifie qu'il vous faut générer des fichiers de polices pour la machine qui fait tourner PHP.

| Position | Type de données C | Description |
|---|---|---|
| Octets 0−3 | int | Nombre de caractères de la police |
| Octets 4−7 | int | Valeur du premier caractère de la police (souvent 32 pour espace) |
| Octets 8−11 | int | Largeur en pixels des caractères |
| Octets 12−15 | int | Hauteur en pixels des caractères |
| Octets 16− | char | Tableau avec les données des caractères, un octet par pixel pour chaque caractère, avec un total de (nombre_caractères*largeur*hauteur) octets. |

Voir aussi imagefontwidth et imagefontheight .

## 6.14.75 imagepalettecopy() : Copie la palette d'une image à l'autre

int **imagepalettecopy** ( resource destination , resource source )

imagepalettecopy copie la palette de l'image source dans l'image destination .

## 6.14.76 imagepng() : Envoie une image PNG vers un navigateur ou un fichier.

int **imagepng** ( resource image , *string filename* )

imagepng envoie l'image GD ( image ) au format PNG sur la sortie standard (typiquement, le navigateur web), ou si filename est fourni, l'envoie dans un fichier.

Exemple avec imagepng

```
<?php
  $im = imagecreatefrompng("test.png");
  imagepng($im);
?>
```

Le nom du fichier est optionnel, et dans ce cas, l'image sera transmise directement à la sortie standard. En envoyant une image de type image/png content−type (grâce à la fonction header ), vous pouvez créer des images PNG avec des scripts PHP.

Voir aussi imagegif , imagewbmp , imagejpeg et imagetypes .

## 6.14.77 imagepolygon() : Dessine un polygone

int **imagepolygon** ( resource image , array points , int num_points , int color )

imagepolygon dessine un polygone dans l'image image . points est un tableau PHP qui contient les sommets du polygone sous la forme : points[0] = x0, points[1] = y0, points[2] = x1, points[3] = y1,

etc. num_points est le nombre de sommets.

Exemple avec imagepolygon

```
<?php
// nouvelle image
$image = imagecreate(400,300);

// couleur de fond
$bg = imagecolorallocate($image,0,0,0);

// couleur pour les bords du polygone
$col_poly = imagecolorallocate($image,255,255,255);

// on dessine le polygone
imagepolygon($image,
             array (
                    0, 0,
                    100, 200,
                    300, 200
             ),
             3,
             $col_poly);

// on affiche l'image
header("Content-type: image/png");
imagepng($image);

?>
```

Voir aussi imagecreate et imagecreatetruecolor .

## 6.14.78 imagepsbbox() : Retourne le rectangle entourant un texte et dessiné avec une police PostScript Type1.

array **imagepsbbox** ( string text , int font , int size , *int space* , *int tightness* , *float angle* )

size est exprimé en pixels.

space permet de changer la valeur par défaut du charactère espace. Cette valeur est ajoutée lors des dessins, et donc, peut être négative.

tightness permet de contrôler la quantité d'espace entre les caractères. Cette quantité est ajoutée lors des dessins, et peut donc être négative.

angle est en degrés.

Les paramètres space et tightness sont exprimés en unité d'espacement de caractères, avec 1 unité vaut 1/1000 d'un em carré (un em : Unité de mesure représentée par un carré dont la dimension horizontale est la même que le corps du caractère).

Les paramètres space , tightness et angle sont optionnels.

Le rectangle entourant est calculé en utilisant les informations disponibles sur les tailles de caractères, et, malheureusement, il a tendance à être légèrement différent du résultat réel final. Si l'angle est de 0 degré, vous pouvez−vous attendre à avoir besoin d'un rectangle d'au moins un pixel plus grand dans toutes les directions.

Note

Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib .

imagepsbbox retourne un tableau contenant les éléments suivants :

| | |
|---|---|
| 0 | Abscisse inférieure gauche |

| | |
|---|---|
| 1 | Ordonnée inférieure gauche |
| 2 | Abscisse supérieure droite |
| 3 | Ordonnée supérieure droite |

Voir aussi imagepstext .

## 6.14.79 imagepscopyfont() : Fait une copie d'une police chargée pour modification ultérieure.

int **imagepscopyfont** ( int fontindex )

Utilisez cette fonction si vous devez faire des modifications dans une polie, comme par exemple l'extension ou la condensation, ou encore, modifier son vecteur d'encodage, tout en conservant l'original. Notez que la police que vous copiez doit être obtenue par imagepsloadfont , et non pas une police qui a elle−même été copiée. Vous pouvez aussi faire des modifications avant de la copier.

Si vous utilisez cette fonction, vous **DEVEZ** libérer les polices obtenues vous même, et dans l'ordre inverse. Sinon, votre script sera **suspendu** .

Si tout a bien été fait, un index valide de police sera retourné, et pourra être utilisé ultérieurement. Sinon, la fonction retournera FALSE , et affichera un message décrivant l'erreur.

| Note |
|---|
| Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib . |

Voir aussi imagepsloadfont .

## 6.14.80 imagepsencodefont() : Change le codage vectoriel d'un caractère dans une police

int **imagepsencodefont** ( int font_index , string encodingfile )

imagepsencodefont charge le codage vectoriel d'un caractère depuis un fichier et change le codage vectoriel de la police correspondante. Etant donné que les polices PostScript ne disposent pas des caractères au−delà de 127, vous aurez sûrement besoin de les changer si vous utilisez une autre langue que l'anglais. Le format exact est décrit dans la documentation T1libs. T1lib est disponible en deux formes : IsoLatin1.enc et IsoLatin2.enc.

Si vous commencez à utiliser cette fonction régulièrement, une meilleure solution est de définir un encodage, et de l'utiliser avec ps.default_encoding dans le fichier de configuration pour utiliser par défaut l'encodage correct.

| Note |
|---|
| Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib . |

## 6.14.81 imagepsextendfont() : Etend ou condense une police de caractères

bool **imagepsextendfont** ( int font_index , float extend )

imagepsextendfont étend ou condense la police de caractères font . Si la valeur de extend est inférieure à 1, ce sera une condensation.

| Note |
| --- |
| Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib . |

## 6.14.82 imagepsfreefont() : Libère la mémoire occupée par une police PostScript Type 1

void **imagepsfreefont** ( int fontindex )

imagepsfreefont libère la mémoire occupée par une police PostScript Type 1.

| Note |
| --- |
| Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib . |

Voir aussi imagepsloadfont .

## 6.14.83 imagepsloadfont() : Charge une police PostScript Type 1 depuis un fichier

int **imagepsloadfont** ( string filename )

En cas de succès, imagepsloadfont retourne un index de police, qui pourra être utilisé pour des opérations ultérieures. Sinon, imagepsloadfont retourne FALSE et affiche un message décrivant ce qui est arrivé.

Exemple avec imagepsloadfont

```
<?php
header ("Content-type: image/jpeg");
$im = imagecreate (350, 45);
$black = imagecolorallocate ($im, 0, 0, 0);
$white = imagecolorallocate ($im, 255, 255, 255);
$font = imagepsloadfont ("bchbi.pfb"); // ou bien situez un de vos fichiers .pfb
imagepstext ($im, "Test ... Ca marche!", $font, 32, $white, $black, 32, 32);
imagepsfreefont ($font);
imagejpeg ($im, "", 100); //Meilleure qualité...
imagedestroy ($im);
?>
```

| Note |
| --- |
| Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib . |

Voir aussi imagepsfreefont .

## 6.14.84 imagepsslantfont() : Incline une police de caractères PostScript

bool **imagepsslantfont** ( int font_index , float slant )

imagepsslantfont met en italique la police de caractères font avec le coefficient slant .

| Note |
| --- |
| Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib . |

## 6.14.85 imagepstext() : Dessine un texte sur une image avec une police PostScript Type1

array **imagepstext** ( resource image , string text , int font , int size , int foreground , int background , int x , int y , *int space* , *int tightness* , *float angle* , *int antialias_steps* )

foreground est la couleur dans laquelle le texte va être dessiné. background est la couleur d'anti aliasing. Aucun pixel avec la couleur background n'est dessiné, ce qui fait que l'arrière−plan n'a pas besoin d'être dans une couleur fixe.

Les coordonnées données ( x , y ) définissent l'origine du premier caractère (grossièrement, le coin inférieur gauche du caractère). Ceci est différent de la fonction imagestring , où (x, y) définissait le coin supérieur gauche du premier caractère. Reportez−vous à la documentation PostScript pour avoir des détails à propos des polices et de leurs tailles.

space permet de changer la taille par défaut du caractère d'espacement. Cette valeur peut être négative.

tightness permet de contrôler la quantité d'espace entre deux caractères. Cette valeur peut être négative.

angle est en degrés.

size est exprimé en pixels.

antialias_steps permet de contrôler le nombre de couleurs du texte anti−aliasé. Les valeurs autorisées sont 4 et 16. 16 est recommandé pour les polices de moins de 20 pixels, car l'effet est alors visible. Avec les tailles plus grandes, utilisez de préférence 4, qui est moins gourmande en ressources.

Les paramètres space et tightness sont exprimés en unité d'espaces caractère, ce qui vaut 1/1000ème d'un em−carré (un em : Unité de mesure représentée par un carré dont la dimension horizontale est la même que le corps du caractère).

Les paramètres space , tightness , angle et antialias sont optionnels.

imagepstext retourne un tableau contenant les éléments suivants :

| | |
|---|---|
| 0 | Abscisse inférieure gauche |
| 1 | Ordonnée inférieure gauche |
| 2 | Abscisse supérieure droite |
| 3 | Ordonnée supérieure droite |

| Note |
|---|
| Cette fonction n'est disponible que si PHP est compilé en utilisant −−enable−t1lib . |

Voir aussi imagepsbbox .

## 6.14.86 imagerectangle() : Dessine un rectangle

int **imagerectangle** ( resource image , int x1 , int y1 , int x2 , int y2 , int col )

imagerectangle dessine un rectangle dans la couleur color , dans l'image image , et en commençant au point supérieur gauche ( x1 , y1 ), et en finissant au point inférieur droit ( x2 , y2 ).

Le coin supérieur gauche est l'origine (0,0).

## 6.14.87 imagerotate() : Fait tourner une image d'un angle

resource **imagerotate** ( resource src_im , float angle , int bgd_color )

imagerotate fait tourner l'image src_im d'un angle de angle degrés bgd_color spécifie la couleur des zones qui seront découvertes après la rotation.

## 6.14.88 imagesavealpha() : Définit l'option pour enregistrer les informations complètes du canal alpha lors de la sauvegarde d'images PNG.

bool **imagesavealpha** ( resource image , bool saveflag )

imagesavealpha définit l'option pour essayer de sauvegarder toutes les informations du canal alpha (en opposition à la transparence à couleur unique) lors de la sauvegarde d'images PNG.

Vous devez désactiver le alphablending ( imagealphablending ($im, FALSE )) pour l'utiliser.

Le canal alpha n'est pas supporté par tous les naviguateurs, si vous avez des problèmes avec le votre, essayez de charger le script avec un naviguateur compatible avec les canaux alpha, c'est à dire le dernier Mozilla.

Voir aussi imagealphablending .

## 6.14.89 imagesetbrush() : Modifie la brosse pour le dessin des lignes

int **imagesetbrush** ( resource image , resource brush )

imagesetbrush remplace la brosse courante pour le dessin des lignes par brush . Cette brosse sera alors utilisée avec les fonctions imageline et imagepolygon .

| Note |
| --- |
| Vous n'avez rien à faire lorsque vous en avez terminé avec une brosse, mais si vous détruisez l'image de brosse, vous ne DEVEZ plus utiliser les options IMG_COLOR_BRUSHED et IMG_COLOR_STYLEDBRUSHED des fonctions imageline et imagepolygon , avant d'avoir créé une nouvelle brosse. |

| Note |
| --- |
| imagesetbrush a été ajoutée en PHP 4.0.6. |

## 6.14.90 imagesetpixel() : Dessine un pixel

int **imagesetpixel** ( resource image , int x , int y , int color )

imagesetpixel dessine un pixel au point ( x , y ) (le coin supérieur gauche est l'origine (0,0)) dans l'image image , et avec la couleur color .

Voir aussi imagecreate et imagecolorallocate .

## 6.14.91 imagesetstyle() : Configure le style pour le dessin des lignes

bool **imagesetstyle** ( resource image , array style )

imagesetstyle permet de choisir le style à utiliser lors du dessin des lignes (comme avec les fonctions imageline and imagepolygon ) lors de l'utilisation de la couleur spéciale IMG_COLOR_STYLED ou bien lors du dessin de lignes avec la couleur IMG_COLOR_STYLEDBRUSHED .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre style est un tableau de pixels. L'exemple suivant dessine une ligne pointillée depuis le coin supérieur gauche vers le coin inférieur droit de l'image :

Exemple avec imagesetstyle

```
 <?php
header ("Content-type: image/jpeg");
$im  = imagecreate (100, 100);
$w   = imagecolorallocate ($im, 255, 255, 255);
$red = imagecolorallocate ($im, 255, 0, 0);

/* Dessine une ligne pointillée de 5 pixels rouges, 5 pixels blancs */
$style = array ($red,$red,$red,$red,$red,$w,$w,$w,$w,$w);
imagesetstyle ($im, $style);
imageline ($im, 0, 0, 100, 100, IMG_COLOR_STYLED);

/* Dessine une lgine avec des smileys, en utilisant imagesetbrush() et imagesetstyle */
$style = array ($w,$w,$w,$w,$w,$w,$w,$w,$w,$w,$w,$w,$w,$red);
imagesetstyle ($im, $style);

$brush = imagecreatefrompng ("http://www.libpng.org/pub/png/images/smile.happy.png");
$w2 = imagecolorallocate($brush,255,255,255);
imagecolortransparent ($brush, $w2);
imagesetbrush ($im, $brush);
imageline ($im, 100, 0, 0, 100, IMG_COLOR_STYLEDBRUSHED);

imagejpeg ($im);
imagedestroy ($im);
?>
```

Note

Cette fonction a été ajoutée en PHP 4.0.6.

Voir aussi imagesetbrush et imageline .

## 6.14.92 imagesetthickness() : Modifie l'épaisseur d'un trait

bool **imagesetthickness** ( resource image , int thickness )

imagesetthickness modifie l'épaisseur du trait des lignes de l'image image . Cette épaisseur intervient dans les dessins de polygones, ellipses, cercles, rectangles, etc... thickness est en pixels.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

imagesetthickness a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

## 6.14.93 imagesettile() : Modifie l'image utilisée pour le carrelage

int **imagesettile** ( resource image , resource tile )

imagesettile remplace l'image de pavement courante par l'image tile , à utiliser dans tous les remplissages (comme avec les fonctions imagefill et imagefilledpolygon ) lors des remplissages avec l'option IMG_COLOR_TILED .

Une image de carrelage est une image utilisée pour remplir une zone, de manière répétitive. N'importe quelle image GD peut servir d'image de remplissage. L'utilisation de la couleur transparente (gérée avec la fonction imagecolortransparent ) permet à certaines zones d'apparaître à travers le carrelage.

| Note |
| --- |
| Vous n'avez rien à faire lorsque vous en avez terminé avec une brosse, mais si vous détruisez l'image de brosse, vous ne DEVEZ plus utiliser l'option IMG_COLOR_TILED des fonctions imagefill et imagefilledpolygon , avant d'avoir créé une nouvelle brosse. |

## 6.14.94 imagestring() : Dessine une chaîne horizontale

int **imagestring** ( resource image , int font , int x , int y , string s , int col )

imagestring dessine une la chaîne sur une ligne horizontale, dans l'image image , aux coordonnées ( x , y ) (le coin supérieur gauche est l'origine (0,0)) dans la couleur color . Si l'argument de police vaut 1, 2, 3, 4 ou 5, une des polices par défaut sera utilisée).

Exemple avec imagestring

```
<?php

// nouvelle iamge 100*30
$im = imagecreate(100, 30);

// fond blanc et texte bleu
$bg = imagecolorallocate($im, 255, 255, 255);
$textcolor = imagecolorallocate($im, 0, 0, 255);

// ajout de la phrase en haut &grave; gauche
imagestring($im, 5, 0, 0, "Hello world!", $textcolor);

// afichage de l'image
header("Content-type: image/jpg");
imagejpeg($im);
?>
```

Voir aussi imageloadfont .

## 6.14.95 imagestringup() : Dessine une chaîne verticale

int **imagestringup** ( resource image , int font , int x , int y , string s , int col )

imagestringup dessine une chaîne sur une ligne verticale dans l'image image aux coordonnées ( x , y ) (l'origine est le coin supérieur gauche (0,0)) dans la couleur color . Si la police utilisée est 1, 2, 3, 4 ou 5, une police par défaut sera utilisée.

Voir aussi imageloadfont .

## 6.14.96 imagesx() : Retourne la largeur d'une image

int **imagesx** ( resource image )

imagesx retourne la largeur de l'image référencée par image .

Exemple avec imagesx

```
<?php

// Création d'une image 300*200 image
$img = imagecreate(300, 200);

echo imagesx($img); // 300

?>
```

Voir aussi getimagesize , imagecreate et imagesy .

## 6.14.97 imagesy() : Retourne la hauteur de l'image

int **imagesy** ( resource image )

imagesy retourne la hauteur de l'image référencée par image .

Exemple avec imagesy

```
<?php

// Création d'une image 300*200
$img = imagecreate(300, 200);

echo imagesy($img); // 200

?>
```

Voir aussi getimagesize , imagecreate et imagesx .

## 6.14.98 imagetruecolortopalette() : Convertit une image en vraies couleurs en image à palette

void **imagetruecolortopalette** ( resource image , bool dither , int ncolors )

imagetruecolortopalette convertit l'image en vraies couleurs image en image à palette. Le code de cette fonction est directement tiré de la librairie du "Independent JPEG Group", qui est tout simplement génial. Le code a été modifié pour préserver l'essentiel du canal alph dans la nouvelle palette, en plus de conserver les couleurs du mieux possible. Mais cela ne fonctionne pas toujours comme voulu. Il est alors préférable de générer un résultat en vraies couleurs, ce qui a toujours le meilleur rendu.

Si dither vaut TRUE , cela indique que l'image doit être ditherée : l'image sera un peu plus granuleuse, mais l'approximation des couleurs sera meilleure.

ncolors est le nombre maximal de couleurs dans la palette finale.

Note

imagetruecolortopalette a été ajoutée en PHP 4.0.6 et nécessite GD 2.0.1.

## 6.14.99 imagettfbbox() : Retourne le rectangle entourant un texte et dessiné avec une police TrueType.

array **imagettfbbox** ( int size , int angle , string fontfile , string text )

imagettfbbox calcule et retourne le rectangle entourant le texte text , écrit avec une police truetype.

*text*
La chaîne à mesurer.
*size*
La taille de la police en pixel.
*fontfile*
Le nom de la police TrueType (peut aussi être une URL.)
*angle*
Angle en degré dans lequel le texte text va être mesuré.

imagettfbbox retourne une tableau avec 8 éléments, représentant les 4 sommets du rectangle ainsi défini.

| | |
|---|---|
| 0 | Coin inférieur gauche, abscisse |
| 1 | Coin inférieur gauche, ordonnée |
| 2 | Coin inférieur droit, abscisse |
| 3 | Coin inférieur droit, ordonnée |
| 4 | Coin supérieur droit, abscisse |
| 5 | Coin supérieur droit, ordonnée |
| 6 | Coin supérieur gauche, abscisse |
| 7 | Coin supérieur gauche, ordonnée |

Les positions des points sont relatives au texte **text** , indépendamment de l'angle : coin supérieur gauche faire référence au coin supérieur gauche du texte écrit horizontalement.

imagettfbbox requiert les bibliothèques GD et Freetype.

Voir aussi imagettftext .

## 6.14.100 imagettftext() : Dessine un texte avec une police TrueType

array **imagettftext** ( resource image , int size , int angle , int x , int y , int color , string fontfile , string text )

imagettftext dessine la chaîne text dans l'image image , en commençant aux coordonnées ( x , y ) (le coin supérieur gauche est l'origine (0,0)), avec un angle de angle , et dans la couleur color , en utilisant la police TrueType identifiée par fontfile . Suivant la version de GD dont vous disposez, lorsque le paramètre fontfile ne commence pas par un slash '/', l'extension '.ttf' sera ajoutée au nom du fichier et la librairie essaiera de trouver un fichier avec ce nom la, dans les chemins prédéfinies de la librairie pour les polices.

Les coordonnées ( x , y ) serviront de référence pour le premier caractère (en gros, le coin inférieur gauche du caractère). C'est différent de imagestring , qui utilise le coin supérieur droit.

angle est donné en degrés, avec degré 0 pour un texte horizontal, et en comptant les angles dans le sens inverse des aiguilles d'une montre (sens direct).

fontfile est le chemin jusqu'à la police TrueType à utiliser.

text est le texte à dessiner, incluant aussi des séquences de caractères UTF−8 (de la forme: { ) pour générer des caractères au−delà de 255.

color est l'index de la couleur dans la palette. Utiliser des index négatifs, revient à supprimer l'anti−aliasing.

imagettftext retourne un tableau de 8 éléments représentant les 4 points marquants les limites du texte. L'ordre des points est: supérieur gauche, supérieur droit, inférieur droit, inférieur gauche. Les points sont nommés relativement au texte à l'horizontale imagecolorexact .

Cet exemple va générer une image JPEG noire de 400x30 pixels, avec les mots "Test en cours...Oméga: Ω" en police blanche, type Arial.

Exemple avec imagettftext

```
<?php
 header("Content−type: image/jpeg");
 $im = imagecreate(400,30);
 $white = imagecolorallocate($im, 255,255,255);
 $black = imagecolorallocate($im, 0,0,0);

 // Remplacez path par votre chemin d'accès aux polices
 imagettftext($im, 20, 0, 10, 20, $white, "/path/arial.ttf",
 "Test en cours... Omega: &#937;");
 imagejpeg($im);
 imagedestroy($im);
?>
```

imagettftext requiert les bibliothèques GD ainsi que FreeType .

Voir aussi imagettfbbox .

## 6.14.101 imagetypes() : Retourne les types d'images supportés par la version courante de PHP

int **imagetypes** ( void )

imagetypes retourne un champ de bits correspondant aux formats d'images supportés par la version de GD utilisée. Les valeurs suivantes sont valables : IMG_GIF | IMG_JPG | IMG_PNG | IMG_WBMP . Pour vous assurer du support PNG , faites ceci :

Exemple avec imagetypes

```
<?php
if (imagetypes() & IMG_PNG) {
    echo "Le type PNG est supporté";
}
?>
```

## 6.14.102 imagewbmp() : Affiche une image WBMP

int **imagewbmp** ( resource image , *string filename* , *int foreground* )

imagewbmp crée l'image WBMP dans le fichier filename , à partir de l'image image . Le paramètre image a été créé avec la fonction imagecreate .

filename est optionnel, et s'il est omis, l'image sera envoyée directement au client. En plaçant l'en−tête IMAGE/VND.WAP.WBMP , dans le champ "content−type", vous pourrez afficher une image WBMP .

| Note |
| --- |
| Le support WBMP n'est disponible que si PHP a été compilé avec GD−1.8 ou plus récent. |

En passant le paramètre optionnel foreground , vous pouvez choisir la couleur de fond. Utilisez l'identifiant retourné par imagecolorallocate comme valeur de ce paramètre. La couleur de fond par défaut est noir.

Voir aussi image2wbmp , imagepng , imagegif , imagejpeg et imagetypes .

## 6.14.103 iptcembed() : Intègre des données binaires IPTC dans une image JPEG

array **iptcembed** ( string iptcdata , string jpeg_file_name , *int spool* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.14.104 iptcparse() : Analyse un bloc binaire IPTC http://www.iptc.org/ et recherche les balises simples.

array **iptcparse** ( string iptcblock )

iptcparse analyse un bloc binaire IPTC et recherche les balises simples. iptcparse retourne un tableau avec les balises comme index et les valeurs de ces balises IPTC dans les valeurs de tableau correspondantes. En cas d'erreur, ou si aucune balise IPTC n'a été trouvée, retourne FALSE .

Voir getimagesize pour un exemple.

## 6.14.105 jpeg2wbmp() : Convertit une image JPEG en image WBMP

int **jpeg2wbmp** ( string jpegname , string wbmpname , int d_height , int d_width , int threshold )

jpeg2wbmp convertit l'image JPEG du fichier jpegname au format WBMP, et la sauve dans le fichier wbmpname . Les paramètres d_height et d_width vous permettent de spécifier la hauteur et la largeur (respectivement) de l'image de destination.

| Note |
| --- |
| Le support WBMP n'est disponible que si PHP a été compilé avec GD−1.8 ou plus récent. |

Voir aussi png2wbmp .

## 6.14.106 png2wbmp() : Convertit une image PNG en image WBMP

int **png2wbmp** ( string pngname , string wbmpname , int d_height , int d_width , int threshold )

png2wbmp convertit l'image PNG du fichier pngname au format WBMP, et la sauve dans le fichier wbmpname . Les paramètres d_height et d_width vous permettent de spécifier la hauteur et la largeur (respectivement) de l'image de destination.

| Note |
| --- |
| Le support WBMP n'est disponible que si PHP a été compilé avec GD−1.8 ou plus récent. |

Voir aussi jpeg2wbmp .

## 6.14.107 read_exif_data() : Alias de exif_read_data

read_exif_data est un alias de exif_read_data .

# 6.15 IMAP

## 6.15.1 Introduction

These functions are not limited to the IMAP protocol, despite their name. The underlying c−client library also supports NNTP , POP3 and local mailbox access methods.

## 6.15.2 Pré−requis

Cette extension requiert la librairie C cliente. Téléchargez cette dernière version à ftp://ftp.cac.washington.edu/imap/ et compilez la.

Puis, copiez le fichier c−client/c−client.a dans /usr/local/lib ou n'importe quel autre dossier qui soit dans le chemin de link. Enfin, copiez les fichiers c−client/rfc822.h , mail.h et linkage.h dans /usr/local/include ou n'importe quel autre dossier qui soit dans le chemin d'inclusion.

| Note |
| --- |
| Assurez vous que c−client.a est bien dans votre chemin de link, sous le nom libc−client.a . |

Copiez les fichiers suivants dans le dossier /usr/local/include ou dans n'importe quel dossier qui soit dans votre chemin d'inclusion.

- c−client.h
- imap4r1.h
- rfc822.h
- mail.h
- linkage.h
- env.h
- env_unix.h
- fs.h
- ftl.h
- misc.h
- nntp.h
- nl.h
- osdep.h
- smtp.h
- tcp.h

| Note |
| --- |
| Le nom de env_unix.h dépend de votre OS. Assurez vous que ces fichiers sont bien copiés avec l'option de suivi des liens. Pour compiler la librairie C cliente avec SSL et/ou avec le support Kerberos, lisez la doc fournie dans la distribution. |

## 6.15.3 Installation

Pour avoir accès à ces fonctions, vous devez compiler PHP avec l'option −−with−imap . Dans nos exemples, vous pourriez utiliser −−with−imap=/usr/local/imap−2000b . Le nom du dossier d'installation dépend de l'emplacement où vous avez créé le dossier, en fonction de la description ci−dessus. Les utilisateurs Windows peuvent inclure la librairie DLL php_imap.dll dans le php.ini

| Note |
| --- |

Suivant le la librairie C cliente qui a été configurée, vous pouvez aussi avoir besoion d'ajouter −−with−imap−ssl=/path/to/openssl/ et/ou −−with−kerberos=/path/to/kerberos dans la ligne de configuration de PHP.

| Attention |
| --- |
| L'extension IMAP ne peut être utilisée simultanément avec l'extension recode ou l'extension YAZ . Ceci est dû au fait que ces deux librairies utilisent un symbole interne commun. |

## 6.15.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.15.5 Types de ressources

## 6.15.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*NIL (entier)*
*OP_DEBUG (entier)*
*OP_READONLY (entier)*
*OP_ANONYMOUS (entier)*
*OP_SHORTCACHE (entier)*
*OP_SILENT (entier)*
*OP_PROTOTYPE (entier)*
*OP_HALFOPEN (entier)*
*OP_EXPUNGE (entier)*
*OP_SECURE (entier)*
*CL_EXPUNGE (entier)*
*FT_UID (entier)*
*FT_PEEK (entier)*
*FT_NOT (entier)*
*FT_INTERNAL (entier)*
*FT_PREFETCHTEXT (entier)*
*ST_UID (entier)*
*ST_SILENT (entier)*
*ST_SET (entier)*
*CP_UID (entier)*
*CP_MOVE (entier)*
*SE_UID (entier)*
*SE_FREE (entier)*
*SE_NOPREFETCH (entier)*
*SO_FREE (entier)*
*SO_NOSERVER (entier)*
*SA_MESSAGES (entier)*
*SA_RECENT (entier)*
*SA_UNSEEN (entier)*
*SA_UIDNEXT (entier)*
*SA_UIDVALIDITY (entier)*

*SA_ALL (entier)*
*LATT_NOINFERIORS (entier)*
*LATT_NOSELECT (entier)*
*LATT_MARKED (entier)*
*LATT_UNMARKED (entier)*
*SORTDATE (entier)*
*SORTARRIVAL (entier)*
*SORTFROM (entier)*
*SORTSUBJECT (entier)*
*SORTTO (entier)*
*SORTCC (entier)*
*SORTSIZE (entier)*
*TYPETEXT (entier)*
*TYPEMULTIPART (entier)*
*TYPEMESSAGE (entier)*
*TYPEAPPLICATION (entier)*
*TYPEAUDIO (entier)*
*TYPEIMAGE (entier)*
*TYPEVIDEO (entier)*
*TYPEOTHER (entier)*
*ENC7BIT (entier)*
*ENC8BIT (entier)*
*ENCBINARY (entier)*
*ENCBASE64 (entier)*
*ENCQUOTEDPRINTABLE (entier)*
*ENCOTHER (entier)*

## 6.15.7 Voir aussi

Ce document ne peut entrer dans les détails de toutes les sujets abordés. Plus d'informations sont disponibles avec la documentation de la librairie C ( docs/internal.txt ) ainsi que les RFC suivantes : documents:

- RFC821 : Simple Mail Transfer Protocol ( SMTP ).
- RFC822 : Standard for ARPA internet text messages.
- RFC2060 : Internet Message Access Protocol ( IMAP ) Version 4rev1.
- RFC1939 : Post Office Protocol Version 3 ( POP3 ).
- RFC977 : Network News Transfer Protocol ( NNTP ).
- RFC2076 : Common Internet Message Headers.
- RFC2045 , RFC2046 , RFC2047 , RFC2048 & RFC2049 : Multipurpose Internet Mail Extensions (MIME).

Une étude approfondie est aussi disponible dans les livres suivants (en anglais): Programming Internet Email par David Wood et Managing IMAP par Dianna Mullet & Kevin Mullet.

| Attention |
| --- |
| Des crashs et des problèmes de démarrage de PHP sont possibles lorsque vous chargez cette extension en même temps que l'extension recode. Voyez l'extension recode pour plus d'informations. |

- Introduction
- Pré−requis
- Installation

- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Voir aussi
- imap_8bit
- imap_alerts
- imap_append
- imap_base64
- imap_binary
- imap_body
- imap_bodystruct
- imap_check
- imap_clearflag_full
- imap_close
- imap_createmailbox
- imap_delete
- imap_deletemailbox
- imap_errors
- imap_expunge
- imap_fetch_overview
- imap_fetchbody
- imap_fetchheader
- imap_fetchstructure
- imap_get_quota
- imap_get_quotaroot
- imap_getacl
- imap_getmailboxes
- imap_getsubscribed
- imap_header
- imap_headerinfo
- imap_headers
- imap_last_error
- imap_list
- imap_listmailbox
- imap_listscan
- imap_listsubscribed
- imap_lsub
- imap_mail_compose
- imap_mail_copy
- imap_mail_move
- imap_mail
- imap_mailboxmsginfo
- imap_mime_header_decode
- imap_msgno
- imap_num_msg
- imap_num_recent
- imap_open
- imap_ping
- imap_qprint
- imap_renamemailbox
- imap_reopen
- imap_rfc822_parse_adrlist
- imap_rfc822_parse_headers
- imap_rfc822_write_address

- imap_scanmailbox
- imap_search
- imap_set_quota
- imap_setacl
- imap_setflag_full
- imap_sort
- imap_status
- imap_subscribe
- imap_thread
- imap_timeout
- imap_uid
- imap_undelete
- imap_unsubscribe
- imap_utf7_decode
- imap_utf7_encode
- imap_utf8

## 6.15.9 imap_alerts() : Retourne toutes les alertes

array **imap_alerts** ( void )

imap_alerts retourne tous les messages d'alerte IMAP générés depuis le dernier appel à imap_alerts ou depuis le début de la page. Lorsque imap_alerts est appelé, la pile d'alertes est vidée.

## 6.15.10 imap_append() : Ajoute une chaîne dans une boîte aux lettres.

bool **imap_append** ( resource imap_stream , string mbox , string message , *string options* )

imap_append ajoute un message dans la boîte aux lettres mbox . Si l'option flags est utilisée, flags sera aussi écrit dans la boîte aux lettres.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Lors des échanges avec le serveur Cyrus IMAP, vous devrez utiliser "\r\n" comme terminaison de ligne, à la place de "\n" ou l'opération échouera.

Exemple avec imap_append

```
 <?php
$stream = imap_open("{votre.hote.imap}INBOX.Drafts","username", "password");
$check = imap_check($stream);
print "Nombre de message avant ajout : ". $check->Nmsgs."\n";
imap_append($stream,"{votre.hote.imap}INBOX.Drafts"
                  ,"From: moi@mon.hote\r\n"
                  ."To: toi@ton.hote\r\n"
                  ."Subject: test\r\n"
                  ."\r\n"
                  ."Ceci est un message de test. Ignorez le\r\n"
                  );
$check = imap_check($stream);
print "Nombre de messages après ajout : ". $check->Nmsgs."\n";
imap_close($stream);
?>
```

## 6.15.11 imap_base64() : Décode un texte encodé en BASE64

string **imap_base64** ( string text )

imap_base64 décode un texte encodé en BASE64. Le texte décodé est retourné sous la forme d'une chaîne (selon la RFC 2045 , Section 6.8).

Voir aussi imap_binary , base64_encode et base64_decode .

## 6.15.12 imap_binary() : Convertit une chaîne à 8 bits en une chaîne à base64.

string **imap_binary** ( string string )

imap_binary convertit la chaîne à 8 bits string en une chaîne à base64 (selon la RFC 2045 , Section 6.8).

imap_binary retourne la chaîne codée.

Voir aussi imap_base64 .

## 6.15.13 imap_body() : Lit le corps d'un message

string **imap_body** ( resource imap_stream , int msg_number , *int options* )

imap_body retourne le corps du message numéro msg_number de la boîte aux lettres courante. L'option flags est un masque qui peut contenir les valeurs suivantes :

- FT_UID − msgno est un UID
- FT_PEEK − Ne pas lever le drapeaux \Seen (Message lu) s'il n'est pas déjà levé.
- FT_INTERNAL − La chaîne renvoyée est au format interne, et ne va pas canoniser les CRLF.

imap_body va retourner une copie brute du corps du message. Pour extraire les sous parties MIME du message, utilisez imap_fetchstructure pour analyser la structure, et imap_fetchbody pour extraire une copie d'une des sous−partie.

## 6.15.14 imap_bodystruct() : Lit la structure d'une section du corps d'un mail

object **imap_bodystruct** ( resource stream_id , int msg_no , int section )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.15.15 imap_check() : Vérifie le courrier de la boîte aux lettres courante.

object **imap_check** ( resource imap_stream )

imap_check retourne les informations à propos de la boîte aux lettres courante. imap_check retourne FALSE en cas d'échec.

imap_check vérifie le statut de la boîte aux lettres courante, sur le serveur imap_stream , et retourne les informations dans un objet avec les membres suivants :

- Date − Date de dernière modification du contenu de la boîte aux lettres
- Driver − protocole utilisé pour accéder à la boîte aux lettres: POP3 , IMAP , NNTP .
- Mailbox − nom de la boîte aux lettres
- Nmsgs − nombre de messages de la boîte aux lettres
- Recent − nombre de messages récents de la boîte aux lettres

## 6.15.16 imap_clearflag_full() : Supprime un flag sur un message

bool **imap_clearflag_full** ( resource stream , string sequence , string flag , string options )

imap_clearflag_full efface le flag flag dans les messages de la séquence sequence , du flux imap stream . Les flags flag que vous pouvez effacer sont "\\Seen", "\\Answered", "\\Flagged", "\\Deleted" et "\\Draft" (tels que définis dans la RFC2060).

Les options sont un masque de bit, qui accepte les valeurs suivantes :

| Constante | Signification |
|---|---|
| ST_UID | la séquence contient des UIDs au lieu de numéro de séquence. |

## 6.15.17 imap_close() : Termine un flux IMAP

bool **imap_close** ( resource imap_stream , *int flag* )

imap_close termine un flux IMAP . imap_close prend un argument optionnel flag , CL_EXPUNGE, qui va retirer automatiquement de la liste la boîte aux lettres.

## 6.15.18 imap_createmailbox() : Crée une nouvelle boîte aux lettres

bool **imap_createmailbox** ( resource imap_stream , string mbox )

imap_createmailbox crée une nouvelle boîte aux lettres nommée mbox . Les noms contenant des caractères spéciaux doivent être encodés.

imap_createmailbox retourne TRUE en cas de succès, et FALSE en cas d'erreur.

Exemple avec imap_createmailbox

```
 <?php
$mbox = imap_open("{your.imap.host}","utilisateur","mot_de_passe",OP_HALFOPEN)
    or die("connexion impossible: ".imap_last_error());
$name1 = "nouvellephpbox";
$name2 = imap_utf7_encode("nouvellephpboxéx");
$newname = $name1;
echo "Le nouveau nom sera '$name1'<br />\n";
// Nous allons créer maintenant une nouvelle boîte aux lettres "phptestbox"
// dans votre dossier inbox, vérifier son état et finalement, la supprimer
// pour remettre votre inbox dans son état initial.
if (@imap_createmailbox($mbox,imap_utf7_encode("{your.imap.host}INBOX.$newname"))){
 $status = @imap_status($mbox,"{your.imap.host}INBOX.$newname",SA_ALL);
 if ($status) {
```

```
   echo "Votre nouvelle boîte '$name1' est dans l'état suivant :<br />\n";
   echo "Messages : "            . $status->messages . "<br />\n";
   echo "Récent : "       . $status->recent . "<br />\n";
   echo "Non lus : "             . $status->unseen . "<br />\n";
   echo "UID suivant : "         . $status->uidnext . "<br />\n";
   echo "UID validité : " . $status->uidvalidity . "<br />\n";

   if(imap_renamemailbox($mbox,"{your.imap.host}INBOX.$newname","{your.imap.host}INBOX.$name2")) {
     echo "renommage de la boîte aux lettres '$name1' en '$name2'<br />\n";
     $newname=$name2;
   } else {
     echo "imap_renamemailbox sur la nouvelle boîte aux lettres a échoué : " . imap_last_error() . "<br />\n";
   }
 } else {
   echo "imap_status sur la nouvelle boîte aux lettres a échoué : " . imap_last_error() . "<br />\n";
 }
 if(@imap_deletemailbox($mbox,"{your.imap.host}INBOX.$newname")) {
   echo "new mailbox supprimée pour remettre tout en état<br />\n";
 } else {
   echo "imap_deletemailbox ur la nouvelle boîte aux lettres a échoué : " . implode("<br />\n", imap_errors()) . "<br />\n";
 }
} else {
 print  "Impossible de créer une nouvelle boîte aux lettres :  " . implode("<br />\n", imap_errors()) . "<br />\n";
}
imap_close($mbox);
?>
```

Voir aussi imap_renamemailbox , imap_deletemailbox et imap_open pour connaître le format des noms de mbox .

## 6.15.19 imap_delete() : Marque le fichier pour l'effacement, dans la boîte aux lettres courante

bool **imap_delete** ( int imap_stream , int msg_number , *int options* )

imap_delete marque le fichier msg_number pour l'effacement, dans la boîte aux lettres courante. Le paramètre optionnel flags ne prend qu'une seule valeur, ft_uid , qui indique à PHP qu'il faut traiter msg_number comme un uid . L'effacement réel n'interviendra que lors de l'appel de la fonction imap_expunge .

imap_delete retourne TRUE .

Exemple avec imap_delete

```
 <?php
$mbox = imap_open ("{your.imap.host}INBOX", "utilisateur", "mot_de_passe")
   or die ("connexion impossible: " . imap_last_error());
$check = imap_mailboxmsginfo ($mbox);
print "Nombre de messages avant effacement  : " . $check->Nmsgs . "<BR>\n" ;
imap_delete ($mbox, 1);
$check = imap_mailboxmsginfo ($mbox);
print "Nombre de messages après effacement: " . $check->Nmsgs . "<BR>\n" ;
imap_expunge ($mbox);
$check = imap_mailboxmsginfo ($mbox);
print "Nombre de messages après imap_expunge: " . $check->Nmsgs . "<BR>\n" ;
imap_close ($mbox);
?>
```

## 6.15.20 imap_deletemailbox() : Efface une boîte aux lettres

bool **imap_deletemailbox** ( resource imap_stream , string mbox )

imap_deletemailbox efface la boîte aux lettres (voir imap_open pour connaître le format des noms de mbox ).

imap_deletemailbox retourne TRUE en cas de succès, et FALSE en cas d'erreur.

Voir aussi imap_createmailbox , imap_renamemailbox et imap_open pour le format du paramètre

mbox .

## 6.15.21 imap_errors() : Retourne toutes les erreurs

array **imap_errors** ( void )

imap_errors retourne tous les messages d'erreurs IMAP générés depuis le dernier appel à imap_errors , ou depuis le début de la page. Lorsque imap_errors est appelés, la pile d'erreur est vidée.

## 6.15.22 imap_expunge() : Efface tous les messages marqués pour l'effacement.

bool **imap_expunge** ( resource imap_stream )

imap_expunge efface tous les messages marqués pour l'effacement par imap_delete .

imap_expunge retourne TRUE .

## 6.15.23 imap_fetch_overview() : Lit un sommaire des en−têtes de messages

array **imap_fetch_overview** ( resource imap_stream , string sequence , *int options* )

imap_fetch_overview lit les en−têtes des courriers électroniques de la séquence sequence et retourne un sommaire de leur contenu. sequence va contenir une séquence d'indice de message ou d'UIDs, si flags cotient FT_UID. La valeur retournée est un tableau d'objets, un par message d'en−tête décrit :

- subject − Le sujet du message
- from − Expéditeur
- date − Date d'expédition
- message_id − Identification du message
- references − est une référence sur l'id de ce message
- size − taille en octets
- uid − UID du message dans la boîte aux lettres
- msgno − numéro de séquence du message dans la boîte
- recent − Ce message est récent
- flagged − Ce message est marqué
- answered − Ce message a donné lieu à une réponse
- deleted − Ce message est marqué pour l'effacement
- seen − Ce message est déjà lu
- draft − Ce message est un brouillon

Exemple avec imap_fetch_overview

```
 <?php
$mbox = imap_open("{your.imap.host:143}","username","password")
    or die("connexion impossible : ".imap_last_error());

 $overview = imap_fetch_overview($mbox,"2,4:6",0);

 if(is_array($overview)) {
        reset($overview);
        while( list($key,$val) = each($overview)) {
```

```
                print     $val->msgno
                . " - " . $val->date
                . " - " . $val->subject
                . "\n";
        }
}

 imap_close($mbox);
?>
```

## 6.15.24 imap_fetchbody() : Retourne une section extraite du corps d'un message

string **imap_fetchbody** ( resource imap_stream , int msg_number , string part_number , *flags options* )

imap_fetchbody va rechercher une section du corps du message, et la retourne sous la forme d'une chaîne. La section est une chaîne d'entiers, séparés par des virgules, qui servent d'index dans le corps du message, comme spécifié dans la norme IMAP4. Le texte n'est alors pas décodé par imap_fetchbody .

L'option imap_fetchbody est un masque qui peut contenir les valeurs suivantes :

- FT_UID − msgono est un UID
- FT_PEEK − Ne pas lever le drapeau \Seen (Message lu) s'il n'est pas déjà levé.
- FT_INTERNAL − La chaîne renvoyée est au format interne, et ne va pas canoniser les CRLF.

Voir aussi imap_fetchstructure .

## 6.15.25 imap_fetchheader() : Retourne l'en−tête d'un message

string **imap_fetchheader** ( resource imap_stream , int msgno , int options )

imap_fetchheader retourne l'en−tête brute et complète RFC2822 du message msgno , et le retourne sous la forme d'une chaîne.

Les options sont :

```
       FT_UID          L'argument
msgno
 est un UID
       FT_INTERNAL     la chaîne renvoyée est au format "internal" ,
                       c'est-à-dire sans canonisation des CRLF
       FT_PREFETCHTEXT RFC822.TEXT doit être pré
       téléchargé en même temps que l'en-tête.
       Cela réduit le RTT sur une connexion
IMAP
, si
       le message complet est souhaité. (e.g. dans une opération
       de sauvegarde dans un fichier).
```

## 6.15.26 imap_fetchstructure() : Lit la structure d'un message.

object **imap_fetchstructure** ( resource imap_stream , int msg_number , *int options* )

imap_fetchstructure la structure du message msg_number . imap_fetchstructure dispose d'une option *flags* , qui une seule valeur, ft_uid , pour indiquer que l'argument msg_number est un uid . imap_fetchstructure retourne un objet avec des propriétés d'enveloppe, de date interne, de taille, de structure de flags et de corps, ainsi qu'un objet pour chaque attachement. La structure est la suivante :

| | |
|---|---|
| type | Type primaire de corps |
| encoding | Codage de transfert du corps |
| ifsubtype | TRUE s'il y a une chaîne de sous type |
| subtype | sous type MIME |
| ifdescription | TRUE s'il y au ne chaîne de description |
| description | Chaîne de description du contenu |
| ifid | TRUE s'il y a une chaîne d'identification |
| id | Chaîne d'identification |
| lines | Nombre de lignes |
| bytes | Nombre d'octets |
| ifdisposition | TRUE s'il y a une chaîne de disposition |
| disposition | Chaîne de disposition |
| ifdparameters | TRUE s'il y a un tableau de paramètres dparameters |
| dparameters | tableau de disposition |
| ifparameters | TRUE si le tableau de paramètres existe |
| parameters | Tableau de paramètres MIME |
| parts | Tableau d'objet décrivant chaque partie du message |

| Note |
|---|
| 1. dparameters est un tableau d'objet où chaque objet à un "attribut" et une "valeur".<br>2. parameter est un tableau d'objet où chaque objet à un "attribut" et une "valeur".<br>3. parts est un tableau d'objets de même structure que l'objet supérieur, mais qui ne contient pas d'autres objets de même sorte. |

| | |
|---|---|
| 0 | text |
| 1 | multipart |
| 2 | message |
| 3 | application |
| 4 | audio |
| 5 | image |
| 6 | vidéo |
| 7 | autre |

| | |
|---|---|
| 0 | 7BIT |
| 1 | 8BIT |
| 2 | BINARY |
| 3 | BASE64 |
| 4 | QUOTED−PRINTABLE |
| 5 | OTHER |

Voir aussi imap_fetchbody .

## 6.15.27 imap_get_quota() : Lit les quotas des boîtes aux lettres

array **imap_get_quota** ( resource imap_stream , string quota_root )

imap_get_quota retourne un tableau contenant les valeurs de quota et courante de la boîte aux lettres quota_root . Le quota représente la taille maximale de votre boîte aux lettres. La valeur courante est l'espace actuellement utilisé par votre boîte aux lettres. imap_get_quota retournera FALSE en cas d'échec.

imap_get_quota ne fonctionne actuellement qu'avec les librairies c−client2000.

imap_stream doit avoir été créé avec la fonction imap_open . Ce flux est nécessairement ouvert en tant qu'administrateur du serveur, pour que les droits nécessaires lui soit alloué. quota_root doit être de la forme : " user.nom ", où "nom" est le nom de la boîte aux lettres que vous souhaitez analyser.

Exemple avec imap_get_quota

```
 <?php
$mbox = imap_open("{votre.hote.imap}","mailadmin","mot de passe",OP_HALFOPEN)
     or die("Connexion impossible : ".imap_last_error());
$quota_value = imap_get_quota($mbox, "user.toto");
if(is_array($quota_value)) {
   print "Utilisation actuelle : " . $quota_value['usage'];
   print "Quota : " . $quota_value['limit'];
}
imap_close($mbox);
?>
```

Depuis PHP 4.3, la fonction reflète plus fidèlement les fonctionnalités édictée par la RFC 2087. Le tableau retourné a changé pour supporter un nombre illimité de ressources retournées (i.e. messages ou sous−dossiers) avec chaque ressource nommée qui est identifiée par une clé. Chaque clé contient alors un autre tableau avec l'utilisation et le quota. L'exemple ci−dessous montre comment l'utiliser.

Pour des raisons de compatibilité, la méthode d'accès originale est toujours disponible, mais il est recommandé de l'abandonner.

Exemple avec imap_get_quota pour PHP 4.3

```
 $mbox = imap_open("{your.imap.host}","mailadmin","password",OP_HALFOPEN)
      or die("Impossible de se connecter : ".imap_last_error());

$quota_values = imap_get_quota($mbox, "user.kalowsky");
if(is_array($quota_values)) {
   $storage = $quota_values['STORAGE'];
   print "Utilisation actuelle de la capacité de stockage : " .  $storage['usage'];
   print "Quota actuel de stockage : " .  $storage['limit'];

   $message = $quota_values['MESSAGE'];
    print "Niveau d'utilisation de MESSAGE : " .  $message['usage'];
   print "Quota de MESSAGE : " .  $message['limit'];

   /* ...  */
 }

 imap_close($mbox);
```

Voir aussi imap_open , imap_set_quota et imap_get_quotaroot .

## 6.15.28 imap_get_quotaroot() : Lit les quotas de chaque utilisateur

array **imap_get_quotaroot** ( resource imap_stream , string quota_root )

imap_get_quotaroot retourne un tableau d'entiers, contenant les quotas de la boîte à lettres de l'utilisateur. Toutes les valeurs sont représentées par une clé basée sur le nom de la boîte, et par un tableau représentant le niveau d'utilisation et les limites.

La valeur limite représente l'espace maximal alloué à l'utilisateur. Le niveau d'utilisation représente l'occupation actuelle de la boîte. Cette fonction retournera FALSE si une erreur est survenue, et un tableau de données si la réponse du serveur n'a pu être comprise.

Cette fonction est uniquement accessible aux utilisateurs de la librairie c−client2000 ou plus récent.

imap_stream est une ressource de connexion, obtenue grâce à imap_open . Cette connexion doit être ouverte avec l 'identité de l'utilisateur que l'on étudie. quota_root doit être un nom de boîte aux lettres (i.e. INBOX).

Exemple avec imap_get_quotaroot

```
 $mbox = imap_open("{your.imap.host}","kalowsky","password",OP_HALFOPEN)
      or die("can't connect: ".imap_last_error());

 $quota = imap_get_quotaroot($mbox, "INBOX");
if(is_array($quota)) {
   $storage = $quota_values['STORAGE'];
   print "STORAGE usage level is: " .  $storage['usage'];
   print "STORAGE limit level is: " .  $storage['limit'];

   $message = $quota_values['MESSAGE'];
    print "MESSAGE niveau d'utilisation : " .  $message['usage'];
   print "MESSAGE  niveau d'utilisation : " .  $message['limit'];

   /* ...  */

 }

 imap_close($mbox);
```

Voir aussi imap_open , imap_set_quota et imap_get_quota .

## 6.15.29 imap_getacl() : Retourne le ACL pour la boîte aux lettres

array **imap_getacl** ( resource stream_id , string mailbox )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.15.30 imap_getmailboxes() : Liste les boîtes aux lettres, et retourne le détail pour chacune.

array **imap_getmailboxes** ( resource imap_stream , string ref , string pattern )

imap_getmailboxes retourne un tableau d'objets contenant les informations sur les boîtes aux lettres. Chaque objet a les attributs de name , qui contient le nom complet de la boîte aux lettres; delimiter , qui est le délimiteur hiérarchique; et attributes . attributes est un masque de bits, qui contient :

- LATT_NOINFERIORS − Cette boîte aux lettres n'a pas d'"enfants" (il n'y a plus de boîtes aux lettres en dessous de celle−ci).
- LATT_NOSELECT − Ceci est juste un container, pas une boîte aux lettres (vous ne pouvez pas l'ouvrir).
- LATT_MARKED − Cette boîte aux lettres est marquée. Utilisé uniquement avec UW−IMAPD.
- LATT_UNMARKED − Cette boîte aux lettres n'est pas marquée. Utilisé uniquement avec UW−IMAPD.

ref ne devrait être que le serveur IMAP sous la forme {imap_server:imap_port} , et pattern spécifie la position dans la hiérarchie des boîtes aux lettres, où il faut commencer à charcher. Si vous voulez passer en revue toute la hiérarchier, passez '*' comme pattern .

Il y a deux caractères spéciaux que vous pouvez utiliser dans pattern : '*' et '%'. '*' signifie : toutes les boîtes aux lettres. Si vous passez pattern comme '*', vous obtiendrez la liste complète des boîtes aux lettres de la hiérarchie. '%' signifie qu'on ne s'interesse qu'au niveau courant. '%' passé à pattern ne retournera que les boîtes aux lettres de niveau supérieur; '~/mail/%'.Sous UW_IMAPD retournera toutes les boîtes aux lettres du dossier ~/mail directory , mais pas leurs enfants.

Exemple avec imap_getmailboxes

```
<?php
$mbox = imap_open("{your.imap.host}","utilisateur","mot_de_passe",OP_HALFOPEN)
     or die("connexion impossible : ".imap_last_error());
$list = imap_getmailboxes($mbox,"{votre.hote.imap}","*");
if(is_array($list)) {
  reset($list);
  while (list($key, $val) = each($list)) {
   print "($key) ";
   print imap_utf7_decode($val->name).",";
   print "'".$val->delimiter."',";
   print $val->attributes."<BR>\n";
  }
} else {
  print "imap_getmailboxes a échoué : ".imap_last_error()."\n";
}
imap_close($mbox);
?>
```

Voir aussi imap_getsubscribed .

## 6.15.31 imap_getsubscribed() : Liste toutes les boîtes aux lettres souscrites

array **imap_getsubscribed** ( resource imap_stream , string ref , string pattern )

imap_getsubscribed est identique à imap_getmailboxes , mais ne retourne que les boîtes aux lettres auxquelles l'utilisateur est inscrit.

## 6.15.32 imap_header() : Lit l'en−tête d'un message

object **imap_header** ( resource imap_stream , int msg_number , *int fromlength* , *int subjectlength* , *string defaulthost* )

imap_header est un alias de imap_headerinfo et lui est identique en tous points.

## 6.15.33 imap_headerinfo() : Lit l'en−tête du message

object **imap_headerinfo** ( resource imap_stream , int msg_number , *int fromlength* , *int subjectlength* , *string defaulthost* )

imap_headerinfo retourne un objet contenant divers éléments d'en−tête.

```
      remail, date, Date, subject, Subject, in_reply_to, message_id,
      newsgroups, followup_to, references
éléments d'en−tête :
  Recent −  'R' si récent et lu
            'N' si récent et non lu
            ' ' si non récent
  Unseen −  'U' si non lu ET non récent
            ' ' si lu OU non lu et récent
  Answered −'A' si répondu,
            ' ' si non répondu
  Deleted − 'D' si effacé,
            ' ' si non effacé
  Draft −   'X' si brouillon,
            ' ' si non brouillon
  Flagged − 'F' si marqué,
            ' ' si non marqué
Notez bien que le comportement récent/non lu est un peu particulier :
si vous voulez savor si un message est non lu, vous devez le vérifier
avec
```

```
Unseen == 'U' || Recent == 'N'
toaddress (toute la ligne d'en-tête To: jusqu'à 1024 caractères)
to[] (retourne un objet avec tout l'en-tête To, contenant):
  personal
  adl
  mailbox
  host
fromaddress (toute la ligne d'en-tête from: jusqu'à 1024 caractères)
from[] (retourne un objet avec tout l'en-tête From, contenant):
  personal
  adl
  mailbox
  host
ccaddress (toute la ligne d'en-tête CC: jusqu'à 1024 caractères)
cc[] (retourne un objet avec tout l'en-tête CC, contenant):
  personal
  adl
  mailbox
  host
bccaddress (toute la ligne d'en-tête BCC: jusqu'à 1024 caractères)
bcc[] (retourne un objet avec tout l'en-tête BCC, contenant):
  personal
  adl
  mailbox
  host
reply_toaddress (oute la ligne d'en-tête Reply_to: jusqu'à 1024 caractères)
reply_to[] (retourne un objet avec tout l'en-tête Reply_to, contenant)
  personal
  adl
  mailbox
  host
senderaddress (toute la ligne d'en-tête Sender: jusqu'à 1024 caractères)
sender[] (retourne un objet avec tout l'en-tête Sender, contenant)
  personal
  adl
  mailbox
  host
return_path (toute la ligne d'en-tête Return-path: jusqu'à 1024 caractères)
return_path[] (retourne un objet avec tout l'en-tête Return-path, contenant)
  personal
  adl
  mailbox
  host
udate (Date du mail, au format UNIX)
fetchfrom (Ligne d'en-tête from formatée pour tenir dans
fromlength

caractères)
fetchsubject (Ligne d'en-tête subject formatée pour tenir dans
subjectlength
 caractères)
```

## 6.15.34 imap_headers() : Retourne les en−têtes de tous les messages d'une boîte aux lettres.

array **imap_headers** ( resource imap_stream )

imap_headers retourne un tableau de chaîne contenant les en−tête des messages. Une chaîne par message.

## 6.15.35 imap_last_error() : Retourne la dernière erreur (si elle existe) qui est survenu lors de la dernière requête.

string **imap_last_error** ( void )

imap_last_error retourne le texte complet de la dernière erreur IMAP (si elle existe) qui est survenu lors de la dernière requête. La pile d'erreur n'est pas touchée. Appeler imap_last_error successivement dans nouvelles erreurs retournera la même erreur.

## 6.15.36 imap_list() : Lit la liste des boîtes aux lettres

array **imap_list** ( resource imap_stream , string ref , string pattern )

imap_list retourne un tableau contenant la liste des noms des boîtes aux lettres. Voyez imap_getmailboxes pour une description des paramètres ref et pattern .

Exemple avec imap_list

```
 <?php
$mbox = imap_open("{your.imap.host}","username","password",OP_HALFOPEN)
      or die("can't connect: ".imap_last_error());

 $list = imap_list($mbox,"{your.imap.host}","*");
if(is_array($list)) {
  reset($list);
  while (list($key, $val) = each($list))
    print imap_utf7_decode($val)."<br>\n";
} else
  print "imap_list a échoué : ".imap_last_error()."\n";

 imap_close($mbox);
?>
```

## 6.15.37 imap_listmailbox() : Liste les boîtes aux lettres

array **imap_listmailbox** ( resource imap_stream , string ref , string pattern )

imap_listmailbox est un alias de imap_list .

## 6.15.38 imap_listscan() : Lit la liste des boîtes aux lettres, et y recherche une chaîne

array **imap_listscan** ( resource imap_stream , string ref , string pattern , string content )

imap_listscan retourne un tableau contenant les noms des boîtes aux lettres qui contiennent la chaîne content dans leurs messages. Cette fonction est similaire à imap_listmailbox , mais va aussi vérifiéer la présence de la chaîne content dans les messages de la boîte aux lettres. Voyez aussi imap_getmailboxes pour une description des paramètres ref et pattern .

## 6.15.39 imap_listsubscribed() : Liste les boîtes aux lettres souscrites

array **imap_listsubscribed** ( resource imap_stream , string ref , string pattern )

imap_listsubscribed retourne un tableau avec toutes les boîtes aux lettres auxquelles vous avez souscrit. Les arguments ref et pattern indiquent respectivement, le dossier où chercher et le nom

des boîtes recherchées, sous la forme d'un masque.

## 6.15.40 imap_lsub() : Liste toutes les boîtes aux lettres enregistrées

array **imap_lsub** ( resource imap_stream , string ref , string pattern )

**imap_lsub** retourne un tableau contenant toutes les boîtes aux lettres que vous avez enregistré. C'est presque le même résultat que imap_listmailbox , mais cette fonction ne va retourner que des boîtes aux lettres que l'utilisateur avec lequelqe vous vous êtes connecté sur le serveur imap a enregistré.

## 6.15.41 imap_mail_compose() : Crée un message MIME

string **imap_mail_compose** ( array envelope , array body )

Exemple avec imap_mail_compose

```
<?php
   $envelope["from"]="musone@afterfive.com";
   $envelope["to"]="musone@darkstar.com";
   $envelope["cc"]="musone@edgeglobal.com";
   $part1["type"]=TYPEMULTIPART;
   $part1["subtype"]="mixed";
   $filename="/tmp/imap.c.gz";
   $fp=fopen($filename,"r");
   $contents=fread($fp,filesize($filename));
   fclose($fp);
   $part2["type"]=TYPEAPPLICATION;
   $part2["encoding"]=ENCBINARY;
   $part2["subtype"]="octet-stream";
   $part2["description"]=basename($filename);
   $part2["contents.data"]=$contents;
   $part3["type"]=TYPETEXT;
   $part3["subtype"]="plain";
   $part3["description"]="description3";
   $part3["contents.data"]="contents.data3\n\n\n\t";
   $body[1]=$part1;
   $body[2]=$part2;
   $body[3]=$part3;
   echo nl2br(imap_mail_compose($envelope,$body));
?>
```

## 6.15.42 imap_mail_copy() : Copie les messages spécifiés dans une boîte aux lettres.

bool **imap_mail_copy** ( resource imap_stream , string msglist , string mbox , *int options* )

imap_mail_copy copie les messages email spécifiés par msglist dans la boîte aux lettres nommée mbox . msglist est un intervalle, et pas seulement une liste numéros de message (comme décrit dans la RFC2060 ).

imap_mail_copy retourne TRUE en cas de succès et FALSE en cas d'erreur.

flags est un masque, qui peut contenir une ou plusieurs des valeurs suivantes :

- CP_UID − la séquence de nombre contient des UIDS
- CP_MOVE − Efface les messages après copie.

## 6.15.43 imap_mail_move() : Déplace des messages dans une boîte aux lettres

bool **imap_mail_move** ( resource imap_stream , string msglist , string mbox , *int options* )

imap_mail_move déplace les messages spécifiés par msglist dans la boîte aux lettres mbox . msglist est un intervalle, et pas seulement une liste de messages (comme décrit dans la RFC2060 ).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

flags est un champ de bit et peut contenir une seule valeur :

- CP_UID − La séquence de nombres contient UIDS

## 6.15.44 imap_mail() : Envoie un message mail

string **imap_mail** ( string to , string subject , string message , *string additional_headers* , *string cc* , *string bcc* , *string rpath* )

imap_mail est uniquement disponible sous PHP 3.

## 6.15.45 imap_mailboxmsginfo() : Lit les informations à propos de la boîte aux lettres courante.

object **imap_mailboxmsginfo** ( resource imap_stream )

imap_mailboxmsginfo retourne les informations à propos de la boîte aux lettres courante. imap_mailboxmsginfo retourne FALSE en cas d'échec.

imap_mailboxmsginfo vérifie le statut courant de la boîte aux lettres sur le serveur, et retourne un objet avec les propriétés suivantes :

| | |
|---|---|
| Date | Date de dernière modification du contenu de la boîte aux lettres |
| Driver | Pilote |
| Mailbox | Nom de la boîte aux lettres |
| Nmsgs | Nombre de messages |
| Recent | Nombre de messages récents |
| Unread | Nombre de messages non lus |
| Deleted | Nombre de messages effacés |
| Size | Taille de la boîte aux lettres |

Exemple avec imap_mailboxmsginfo

```
 <?php
$mbox = imap_open("{your.imap.host}INBOX","utilisateur", "mot_de_passe")
     or die("conexion impossible: ".imap_last_error());
$check = imap_mailboxmsginfo($mbox);
if($check) {
   print "Date: "    . $check->Date    ."<BR>\n" ;
   print "Pilote: "  . $check->Driver  ."<BR>\n" ;
   print "Mailbox: " . $check->Mailbox ."<BR>\n" ;
   print "Messages: ". $check->Nmsgs   ."<BR>\n" ;
   print "Récent: "  . $check->Recent  ."<BR>\n" ;
   print "Non lus: "  . $check->Unread  ."<BR>\n" ;
   print "Effacés: " . $check->Deleted ."<BR>\n" ;
   print "Taille: "    . $check->Size    ."<BR>\n" ;
} else {
   print "imap_check() a échoué: ".imap_last_error(). "<BR>\n";
}
imap_close($mbox);
?>
```

## 6.15.46 imap_mime_header_decode() : Décode les éléments MIME d'une en−tête

array **imap_mime_header_decode** ( string text )

imap_mime_header_decode décode un message MIME qui contient des données non ASCII (Voir RFC2047 ) Les éléments décodés sont retournés dans un tableau d'objets. Chacun de ces objets a deux propriétés : "charset" & "text". Si l'élément n'a pas été encodé, ou, en d'autres termes, sil il est en clair (plain US_ASCII), la propriété "charset" est mise à "default".

Exemple imap_mime_header_decode

```
 <?php
$text="=?ISO−8859−1?Q?Keld_J=F8rn_Simonsen?= <keld@dkuug.dk>";
$elements=imap_mime_header_decode($text);
for($i=0;$i<count($elements);$i++) {
   echo "Charset: {$elements[$i]−>charset}\n";
   echo "Texte: {$elements[$i]−>text}\n\n";
}
?>
```

Dans l'exemple ci−dessus, on trouve deux éléments : le premier a été encodé en ISO−8859−1, et le second est en clair.

## 6.15.47 imap_msgno() : Retourne le numéro de séquence de message pour un UID donné.

int **imap_msgno** ( resource imap_stream , int uid )

imap_msgno retourne le numéro de séquence de message pour l'UID uid . C'est la fonction contraire de imap_uid .

## 6.15.48 imap_num_msg() : Retourne le nombre de message dans la boîte aux lettres courante.

int **imap_num_msg** ( resource imap_stream )

imap_num_msg retourne le nombre de message dans la boîte aux lettres courante.

Voir aussi imap_num_recent et imap_status .

## 6.15.49 imap_num_recent() : Retourne le nombre de messages récents dans la boîte aux lettres courante.

int **imap_num_recent** ( resource imap_stream )

imap_num_recent retourne le nombre de message récents dans la boîte aux lettres courante.

Voir aussi imap_num_msg et imap_status .

## 6.15.50 imap_open() : Ouvre un flux IMAP vers une boîte aux lettres

resource **imap_open** ( string mailbox , string username , string password , *int options* )

imap_open retourne un flux IMAP en cas de succès, et FALSE en cas d'erreur. imap_open peut aussi être utilisée pour ouvrir des flots sur des serveurs POP3 et NNTP .

Un nom de boîte aux lettres est constitué d'une adresse de serveur, et d'une adresse de boîte sur ce serveur. Le mot réservé INBOX représente la boîte aux lettres de l'utilisateur courant. L'adresse du serveur, mise entre accolades '{' et '}', est constitué du nom du serveur ou de son adresse IP, d'une spécification de protocole (commençant par '/') et d'un port optionnel (spécifié avec ':'). Cette partie est obligatoire dans les paramètres de la boîte aux lettres. Les noms de boîtes aux lettres qui contiennent des caractères spéciaux (en dehors de l'espace ASCII) doivent être encodés avec imap_utf7_encode .

Les options sont un masque de bit, qui peut prendre une ou plusieurs des valeurs suivantes :

- OP_READONLY − Ouvre une boîte aux lettres en lecture seule
- OP_ANONYMOUS − Ne pas utiliser, ou modifier le fichier .newsrc pour les news.
- OP_HALFOPEN − Pour les noms IMAP et NNTP , ouvre une connexion mais n'ouvre pas une boîte aux lettres.
- CL_EXPUNGE − Supprime automatiquement la boîte aux lettres de la liste, lors de la terminaison du flux.

Pour se connecter à un serveur IMAP , on peut utiliser la commande suivante :

Connexion à un serveur IMAP avec imap_open

```
<?php
$mbox  = imap_open("{localhost:143}INBOX","user_id","password");
?>
```

Pour se connecter à un serveur POP3 qui fonctionne sur le port 110 de la machine locale on peut utiliser la commande suivante :

Connexion à un serveur POP3 avec imap_open

```
<?php
$mbox = imap_open("{localhost:110/pop3}INBOX","user_id","password");
?>
```

Pour se connecter à un serveur IMAP SSL ou POP3 SSL, ajoutez /ssl après le protocole :

Connexion à un serveur IMAP SSL ou POP3 SSL avec imap_open

```
<?php
$mbox = imap_open ("{localhost:993/imap/ssl}INBOX", "user_id", "password");
?>
```

Pour se connecter à un serveur SSL IMAP ou POP3 avec un certificat ajoutez /ssl/novalidate−cert après le protocole :

Connexion à un serveur IMAP SSL ou POP3 SSL avec imap_open et un certificat

```
<?php
$mbox = imap_open ("{localhost:995/pop3/ssl/novalidate-cert}", "user_id", "password");
?>
```

Pour se connecter à un serveur NNTP qui fonctionne sur le port 119 de la machine locale on peut utiliser la commande:

Connexion à un serveur NNTP avec imap_open

```
<?php
$nntp = imap_open("{localhost:119/nntp}comp.test","","");
?>
```

Pour se connecter à un serveur distant, remplacez "localhost" par le nom ou l'adresse IP de la machine.

Exemple avec imap_open

```
<?php
$mbox = imap_open ("{votre.hote.imap:143}", "nom_utilisateur", "mot de passe");
echo "<p><h1>Mailboxes</h1>\n";
```

```
$folders = imap_listmailbox ($mbox, "{votre.hote.imap:143}", "*");
if ($folders ==
 <font face="courrier">FALSE</font>
) {
   echo "Appel échoué<BR>\n";
} else {
   while (list ($key, $val) = each ($folders)) {
       echo $val."<BR>\n";
   }
}
echo "<p><h1>en-têtes dans INBOX</h1>\n";
$headers = imap_headers ($mbox);
if ($headers ==
 <font face="courrier">FALSE</font>
) {
   echo "Appel échoué<BR>\n";
} else {
   while (list ($key,$val) = each ($headers)) {
       echo $val."<BR>\n";
   }
}
imap_close($mbox);
?>
```

## 6.15.51 imap_ping() : Vérifie que le flux IMAP est toujours actif

bool **imap_ping** ( resource imap_stream )

imap_ping retourne TRUE si le flux imap_stream existe toujours, et FALSE sinon.

imap_ping vérifie que le flux IMAP est toujours actif, en lui envoyant un ping. Cette fonction permet de se rendre compte que du mail est arrivé : c'est même la méthode préconisée pour des tests périodiques de vérification du courrier. Cette fonction peut aussi servir à garder une connexion ouverte, avec les serveurs dotés d'un délai d'expiration.

## 6.15.52 imap_qprint() : Convertit une chaîne à guillemets en une chaîne à 8 bits.

string **imap_qprint** ( string string )

imap_qprint convertit la chaîne à guillemets string en une chaîne à 8 bits (selon la RFC2045 , section 6.7).

imap_qprint retourne une chaîne 8 bits (binaire).

Voir aussi imap_8bit .

## 6.15.53 imap_renamemailbox() : Renomme une boîte aux lettres

bool **imap_renamemailbox** ( resource imap_stream , string old_mbox , string new_mbox )

imap_renamemailbox renomme la boîte aux lettres old_mbox en new_mbox .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi imap_createmailbox , imap_deletemailbox et imap_open pour le format de mbox .

## 6.15.54 imap_reopen() : Ouvre un flux IMAP vers une nouvelle boîte aux lettres.

bool **imap_reopen** ( resource imap_stream , string mailbox , *string options* )

imap_reopen réouvre la connexion spécifiée au serveur IMAP ou NNTP , avec une nouvelle boîtes aux lettres.

Les options sont des masques de bit, qui peuvent contenir les valeurs suivantes :

- OP_READONLY − Ouvre une boîte aux lettres en lecture seule
- OP_ANONYMOUS − Ne pas utiliser, ou modifier le fichier .newsrc pour les news
- OP_HALFOPEN − Pour les noms IMAP et NNTP , ouvre une connexion mais n'ouvre pas une boîte aux lettres.
- CL_EXPUNGE − Supprime automatiquement la boîte aux lettres de la liste, lors de la terminaison du flux. (voir imap_delete et imap_expunge ).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.15.55 imap_rfc822_parse_adrlist() : Analyse une adresse email

array **imap_rfc822_parse_adrlist** ( string address , string default_host )

imap_rfc822_parse_adrlist analyse la chaîne address , tel que définie dans la RFC2822 et essaie, pour chaque adresse, de retourner un tableau d'objets. Les 4 membres de ces objets sont :

- mailbox − Le nom de la boîte aux lettres
- host − le nom de l'hôte
- personal − Le nom personnel
- adl − at domain source route (NDT : ???).

Exemple avec imap_rfc822_parse_adrlist

```
<?php
 $address_string = "Hartmut Holzgraefe <hartmut@cvs.php.net>, postmaster@somedomain.net, root";
 $address_array  = imap_rfc822_parse_adrlist($address_string,"somedomain.net");
 if(! is_array($address_array)) die("Erreur!\n");

  reset($address_array);
 while(list($key,$val)=each($address_array)){
   print "boîte    : ".$val->mailbox."<br>\n";
   print "hôte     : ".$val->host."<br>\n";
   print "nom      : ".$val->personal."<br>\n";
   print "adl      : ".$val->adl."<p>\n";
 }
?>
```

## 6.15.56 imap_rfc822_parse_headers() : Analyse un en−tête mail

object **imap_rfc822_parse_headers** ( string headers , *string defaulthost* )

imap_rfc822_parse_headers analyse la chaîne headers , et retourne un objet contenant differents éléments, similaires à la fonction imap_header , hormis les flags, et autres éléments liés au serveur IMAP .

## 6.15.57 imap_rfc822_write_address() : Retourne une adresse email formatée correctement

string **imap_rfc822_write_address** ( string mailbox , string host , string personal )

mailbox retourne une adresse email formatée correctement selon la RFC2822 , à partir du nom de la boîte aux lettres de l'hôte host , et des informations personnelles personal .

Exemple avec imap_rfc822_write_address

```
<?php
 print imap_rfc822_write_address("hartmut","cvs.php.net","Hartmut Holzgraefe")."\n";
?>
```

## 6.15.58 imap_scanmailbox() : Lit la liste des boîtes aux lettres, et y recherche une chaîne

array **imap_scanmailbox** ( resource imap_stream , string ref , string pattern , string content )

imap_scanmailbox est un alias de imap_listscan .

## 6.15.59 imap_search() : Retourne un tableau de message après recherche.

array **imap_search** ( resource imap_stream , string criteria , int options )

imap_search effectue une recherche dans la boîte aux lettres courante, sur le flux IMAP courant. criteria est une chaîne, délimitée par des espaces, dans laquelle les mots−clés suivants sont acceptés. Tous les arguments multi−mots doivent être entre guillemets :

- ALL − retourne tous les message qui vérifie le reste du critère.
- ANSWERED − tous les messages avec le flag \\ANSWERED
- BCC "string" − tous les messages avec la chaîne "string" dans le champ Bcc :
- BEFORE "date" − tous les messages avec Date : avant "date"
- BODY "string" − tous les messages avec "string" dans le corps
- CC "string" − tous les messages avec "string" dans le champ Cc :
- DELETED − tous les messages effacés
- FLAGGED − tous les messages avec le flag \\FLAGGED (parfois interprété comme Important ou Urgent)
- FROM "string" − tous les messages avec la chaîne "string" dans le champ From :
- KEYWORD "string" − tous les messags avec la chaîne "string" comme mot clé
- NEW − tous les nouveaux messages
- OLD − tous les anciens messages
- ON "date" − tous les messages avec la date "date" comme champ Date :
- RECENT − tous les messages avec le flag \\RECENT
- SEEN − tous les messages lus (avec le flag\\SEEN flag)
- SINCE "date" − tous les messages avec la date Date: après "date"
- SUBJECT "string" − tous les messages avec la chaîne "string" dans le champ Subject :
- TEXT "string" − tous les messages avec le texte "string"
- TO "string" − tous les messages avec la chaîne "string" dans le champ To :
- UNANSWERED − tous les messages non répondus
- UNDELETED − tous les messages non effacés
- UNFLAGGED − tous les messages non flaggés

- UNKEYWORD "string" − tous les messages dans le mot clés "string"
- UNSEEN − tous les messages non lus

Par exemple, pour rechercher les messages non répondus, envoyés par maman, vous pouvez utiliser : "UNANSWERED FROM maman". Les recherches semblent insensibles à la casse. Cette liste de critères est issue du code d'un client C UW et peut être incomplète ou inprécise. (voir aussi RFC2060, section 6.4.4).

Les valeurs pour les flags sont SE_UID, qui fait que le tableau réponse contient les UIDs plutôt que les numéros de séquence.

## 6.15.60 imap_set_quota() : Modifie le quota d'une boîte aux lettres

bool **imap_set_quota** ( resource imap_stream , string quota_root , int quota_limit )

imap_set_quota modifie le quota de la boîte aux lettres quota_root , en la fixant à quota_limit . imap_set_quota requiert que imap_stream ait été ouvert avec un compte d'administrateur, pour avoir les droits nécessaires : elle ne fonctionnera avec aucun autre utilisateur.

imap_get_quota ne fonctionne actuellement qu'avec les librairies c−client2000.

imap_stream doit avoir été créé avec la fonction imap_open . Ce flux est nécessairement ouvert en tant qu'administrateur du serveur, pour que les droits nécessaires lui soit alloué. quota_root doit être de la forme : " user.nom ", où "nom" est le nom de la boîte aux lettres que vous souhaitez analyser. quota_limit est la nouvelle taille maximum (en ko) de la boîte quota_root .

imap_set_quota retourne TRUE en cas de succès, et FALSE sinon.

Exemple avec imap_set_quota

```
 <?php
$mbox = imap_open ("{your.imap.host:143}", "mailadmin", "password");

if(!imap_set_quota($mbox, "user.kalowsky", 3000)) {
    print "Error in setting quota\n";
    return;
}

imap_close($mbox);
?>
```

Voir aussi imap_open et imap_set_quota .

## 6.15.61 imap_setacl() : Modifie le ACL de la boîte aux lettres

bool **imap_setacl** ( resource stream_id , string mailbox , string id , string rights )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.15.62 imap_setflag_full() : Positionne un flag sur un message

bool **imap_setflag_full** ( resource stream , string sequence , string flag , string options )

imap_setflag_full affecte le flag spécifié aux messages de la sequence donné.

Les flags que vous pouvez modifier sont "\\Seen", "\\Answered", "\\Flagged", "\\Deleted", "\\Draft" et "\\Recent" (comme défini dans la RFC2060).

Les options sont un masque de bits, et peuvent contenir les valeurs suivantes :

| Constante | Signification |
|---|---|
| ST_UID | la séquence contient des UIDs au lieu de numéro de séquence. |

Exemple avec imap_setflag_full

```
 <?php
$mbox = imap_open("{your.imap.host:143}","username","password")
     or die("can't connect: ".imap_last_error());

 $status = imap_setflag_full($mbox,"2,5","\\Seen \\Flagged");

 print gettype($status)."\n";
print $status."\n";

 imap_close($mbox);
?>
```

## 6.15.63 imap_sort() : Trie des messages

array **imap_sort** ( resource stream , int criteria , int reverse , *int options* , *string search_criteria* )

imap_sort retourne un tableau de nombre de message, triés suivant les paramètres suivants :

reverse vaut 1 pour signifier : tri inverse.

Les critères peuvent être un (et un seul) parmi les suivants :

```
SORTDATE        Date du message
SORTARRIVAL     Date d'arrivée
SORTFROM        Nom de la première boîte aux lettres de l'adresse d'origine (From address)
SORTSUBJECT     Sujet du message
SORTTO          Nom de la première boîte aux lettres de destination (To address)
SORTCC          Nom de la boîte aux lettres de copie cachée (cc address)
SORTSIZE        Taille du message en octets
```

Les flags dont des masques de bits, d'un ou plusieurs des éléments suivants :

```
SE_UID          Retourne l'UIDs à la place d'une séquence
de nombres.
SE_NOPREFETCH   Ne pas pré-télécharger les messages
trouvés.
```

## 6.15.64 imap_status() : Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante.

object **imap_status** ( resource imap_stream , string mailbox , int options )

imap_status retourne un objet contenant les informations de statut. Les options valables sont :

- SA_MESSAGES − met la valeur de status−>messages au nombre de messages dans la boîtes aux lettres.

- SA_RECENT − met la valeur destatus−>recent au nombre de messages récents dans la boîte aux lettres.
- SA_UNSEEN − met la valeur de status−>unseen au nombre de messages non lus dans la boîte aux lettres.
- SA_UIDNEXT − met la valeur de status−>uidnext à la prochaine valeur d'uid qui sera utilisée.
- SA_UIDVALIDITY − met la valeur de status−>uidvalidity à une constante, qui change lorsque l'uid de la boîte aux lettres n'est plus valide.
- SA_ALL − fixe les valeurs de toutes les précédents.

status−>flags est aussi fixé : c'est un masque de bit qui peut contenir tous les flags ci−dessus.

Exemple imap_status

```
<?php
 $mbox = imap_open("{your.imap.host}","utilisateur","mot_de_passe",OP_HALFOPEN)
    or die("can't connect: ".imap_last_error());
 $status = imap_status($mbox,"{your.imap.host}INBOX",SA_ALL);
 if($status) {
   echo"Messages:   ". $status->messages    . "<br />\n";
   echo "Récents:     ". $status->recent      . "<br />\n";
   echo "Non lus:     ". $status->unseen      . "<br />\n";
   echo "UIDnext:    ". $status->uidnext     . "<br />\n";
   echo "UIDvalidité:". $status->uidvalidity . "<br />\n";
 } else {
   echo "imap_status a échoué : " . imap_last_error() . "\n";
 }
 imap_close($mbox);
?>
```

## 6.15.65 imap_subscribe() : Souscrit à une boîte aux lettres

bool **imap_subscribe** ( resource imap_stream , string mbox )

imap_subscribe souscrit à la boîte aux lettres mbox .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.15.66 imap_thread() : Retourne l'arbre de REFERENCE organisé par thread

array **imap_thread** ( resource stream_id , *int options* )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.15.67 imap_timeout() : Configure ou retourne le timeout

mixed **imap_timeout** ( int timeout_type , *int timeout* )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.15.68 imap_uid() : Retourne l'UID d'un message.

int **imap_uid** ( resource imap_stream , int msgno )

imap_uid retourne l'UID pour le message msgno . Un UID est un identifant unique que ne change jamais, alors que le numéro du message dans la lsite des messages peut changer à toute modification de la boîte aux lettres. C'est la fonction contraire de imap_msgno .

| Note |
| --- |
| Cette fonctionnalité n'est pas supportées par les boîtes aux lettres POP3. |

## 6.15.69 imap_undelete() : Enlève la marque d'effacement d'un message

bool **imap_undelete** ( resource imap_stream , int msg_number )

imap_undelete enlève la marque d'effacement du message msg_number , placée avec imap_delete .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.15.70 imap_unsubscribe() : Termine la souscription à une boîte aux lettres

bool **imap_unsubscribe** ( string imap_stream , string mbox )

imap_unsubscribe termine la souscription à la boîte aux lettres mbox .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.15.71 imap_utf7_decode() : Décode une chaîne modifiée UTF−7.

string **imap_utf7_decode** ( string text )

imap_utf7_decode décode la chaîne UTF−7 text en données 8 bits.

imap_utf7_decode retourne les données 8bits décodées, ou FALSE si la chaîne text n'est pas au format UTF−7. Cette fonction sert à décoder des noms de boîtes aux lettres qui contiennent des caractères internationaux hors de l'espace ASCII. Le standard UTF−7 est défini dans la RFC 2060 , section 5.1.3 (l'original UTF−7 a été défini dans RFC1642 ).

## 6.15.72 imap_utf7_encode() : Convertit des données 8bit en texte UTF−7.

string **imap_utf7_encode** ( string data )

imap_utf7_encode retourne les données data 8bits encodées, ou FALSE si une erreur est survenue. Cette fonction sert à encoder des noms de boîtes aux lettres qui contiennent des caractères internationaux hors de l'espace ASCII. Le standard UTF−7 est défini dans la RFC 2060 , section 5.1.3 (l'original UTF−7 a été défini dans RFC1642 ).

imap_utf7_encode retourne un texte UTF−7.

## 6.15.73 imap_utf8() : Convertit du texte au format MIME en UTF8

string **imap_utf8** ( string mime_encoded_text )

imap_utf8 convertit le texte mime_encoded_text en UTF8. Les spécifications de MIME et UTF8 sont décrites dans les RFC2047 et RFC2044 .

# 6.16 LDAP

## 6.16.1 Introduction

LDAP siginifie : Lightweight Directory Access Protocol (Protocole léger d'accès aux annuaires). C'est un protocole utilisé pour accéder aux "serveurs de dossiers". Ces serveurs sont des bases de données particulières, qui stockent les informations sous forme d'arborescence.

Le concept d'arborescence est similaire à celui de la structure de votre système de fichier, hormis le fait que dans ce contexte, la racine s'appelle "le monde", et que le premier niveau de sous−dossier s'appelle "pays". Les niveaux encore en dessous sont des "compagnies" "organisation" ou "places", et encore plus bas, vous trouverez des "personnes" et même, des "équipements" et "documents".

Pour identifier un fichier dans votre disque, vous utilisez un chemin tel que

```
/usr/local/mon_application/documents
```

Le slash indique une division dans la référence, et la séquence est lue de gauche à droite.

l'équivalent d'une référence globale en LDAP s'appelle un "nom distingué" ("distinguished name"), aussi appelé "dn". Un exemple de dn serait :

```
cn=Jean Dupond,ou=Comptabilité,o=Ma Compagnie,c=FR
```

La virgule marque la séparation de chaque division comme référence, et la séquence est lue de droite à gauche. Il faut donc lire :

```
country = FR
organization = Ma Compagnie
organizationalUnit = Comptabilité
commonName = Jean Dupond
```

De la même façon qu'il n'y a pas de règle obligatoire sur comment organiser les fichiers sur un disque dur, un responsable de serveur de dossiers peut organiser le serveur comme cela lui semble le plus pratique. Cependant, il y a des conventions à utiliser. Le principe est que vous ne pouvez pas écrire accéder à un serveur de dossier à moins que vous ne connaissiez sa structure, de même que vous ne pouvez une base de données sans en connaître les tables et les bases.

Bien plus d'informations sont disponibles aux URL suivantes (en anglais) :

- Netscape
- OpenLDAP Project
- LDAP World

Le SDK Netscape contient un guide du programmeur ( Programmer's Guide ) très utile, au format HTML (et en anglais).

## 6.16.2 Pré−requis

Vous devez télécharger et compiler les librairies clients LDAP, soit celles de University of Michigan ldap−3.3 package , soit le Netscape Directory SDK 3.0 ou OpenLDAP , pour compiler le support LDAP.

## 6.16.3 Installation

Le support LDAP de PHP n'est pas activé par défaut. Vous devez utiliser l'option de configuration −−with−ldap lorsque vous compilez PHP.

| Note |
| --- |
| Note aux utilisateurs Win32<br><br>Afin d'activer ce module dans l'environnement Windows, vous devez copier les librairies **libsasl.dll** depuis le dossier DLL de PHP/Win32 dans le dossier système SYSTEM32 de votre machine (par exemple : C:\WINNT\SYSTEM32 or C:\WINDOWS\SYSTEM32). |

## 6.16.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
| --- | --- | --- |
| ldap.max_links | "−1" | PHP_INI_SYSTEM |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

## 6.16.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.16.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*LDAP_DEREF_NEVER (entier)*
*LDAP_DEREF_SEARCHING (entier)*
*LDAP_DEREF_FINDING (entier)*
*LDAP_DEREF_ALWAYS (entier)*
*LDAP_OPT_DEREF (entier)*
*LDAP_OPT_SIZELIMIT (entier)*
*LDAP_OPT_TIMELIMIT (entier)*
*LDAP_OPT_PROTOCOL_VERSION (entier)*
*LDAP_OPT_ERROR_NUMBER (entier)*
*LDAP_OPT_REFERRALS (entier)*
*LDAP_OPT_RESTART (entier)*
*LDAP_OPT_HOST_NAME (entier)*
*LDAP_OPT_ERROR_STRING (entier)*
*LDAP_OPT_MATCHED_DN (entier)*

*LDAP_OPT_SERVER_CONTROLS (entier)*
*LDAP_OPT_CLIENT_CONTROLS (entier)*
*LDAP_OPT_DEBUG_LEVEL (entier)*
*GSLC_SSL_NO_AUTH (entier)*
*GSLC_SSL_ONEWAY_AUTH (entier)*
*GSLC_SSL_TWOWAY_AUTH (entier)*

## 6.16.7 Exemples

Lit les informations sur toutes les entrées dont le nom commence par "S" sur le serveur de dossier, puis affiche le nom et l'adresse email.

Recherche avec LDAP

```
 <?php
// La séquence de base avec LDAP est
 // connexion, liaison, recherche, interprétation du résultat
// déconnexion

echo "<h3>requête de test de LDAP</h3>";
echo "Connexion ...";
$ds=ldap_connect("localhost");  // doit être un serveur LDAP valide!
echo "Le résultat de connexion est ".$ds."<p>";

if ($ds) {
     echo "Liaison ...";
     $r=ldap_bind($ds);     // this is an "anonymous" bind, typically
                           // read-only access
    echo "Le résultat de connexion est ".$r."<p>";

    echo "Recherchons (sn=S*) ...";
    // Recherche par nom
    $sr=ldap_search($ds,"o=My Company, c=US", "sn=S*");
    echo "Le résultat de la recherche est ".$sr."<p>";

    echo "Le nombre d'entrées retournées est ".ldap_count_entries($ds,$sr)."<p>";

    echo "Lecture des entrées ...<p>";
    $info = ldap_get_entries($ds, $sr);
    echo "Données pour ".$info["count"]." entrées:<p>";

    for ($i=0; $i<$info["count"]; $i++) {
        echo "dn est : ". $info[$i]["dn"] ."<br>";
        echo "premiere entree cn : ". $info[$i]["cn"][0] ."<br>";
        echo "premier email : ". $info[$i]["mail"][0] ."<p>";
    }

    echo "Fermeture de la connexion";
    ldap_close($ds);

} else {
    echo "<h4>Impossible de se connecter au serveur LDAP.</h4>";
}
?>
```

## 6.16.8 Utiliser les fonctions LDAP de PHP

Avant d'utiliser les fonctions LDAP, vous devez connaître

- Le nom ou l'adresse du serveur de dossiers que vous voudrez utiliser
- Le "base dn" du serveur (la partie du dossier monde qui est disponible sur ce serveur, ce qui peut être "o=Ma Compagnie,c=FR")
- Le mot de passe éventuel d'accès au serveur (de nombreux serveurs fournissent un accès anonyme en lecture, mais requièrent des mots de passe pour tout le reste.

La séquence LDAP typique que vous exécuterez sera la suivante :

```
ldap_connect()    // établit une connexion au serveur
    |
ldap_bind()       // connexion anonyme ou identifiée
```

```
   |
réalisation de commadnes comme des recherches ou des
modifications, puis affichage du résultat.
   |
ldap_close()      // déconnexion
```


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- Utiliser les fonctions LDAP de PHP
- ldap_8859_to_t61
- ldap_add
- ldap_bind
- ldap_close
- ldap_compare
- ldap_connect
- ldap_count_entries
- ldap_delete
- ldap_dn2ufn
- ldap_err2str
- ldap_errno
- ldap_error
- ldap_explode_dn
- ldap_first_attribute
- ldap_first_entry
- ldap_first_reference
- ldap_free_result
- ldap_get_attributes
- ldap_get_dn
- ldap_get_entries
- ldap_get_option
- ldap_get_values_len
- ldap_get_values
- ldap_list
- ldap_mod_add
- ldap_mod_del
- ldap_mod_replace
- ldap_modify
- ldap_next_attribute
- ldap_next_entry
- ldap_next_reference
- ldap_parse_reference
- ldap_parse_result
- ldap_read
- ldap_rename
- ldap_search
- ldap_set_option
- ldap_set_rebind_proc
- ldap_sort
- ldap_start_tls
- ldap_t61_to_8859

- ldap_unbind

## 6.16.10 ldap_add() : Ajoute une entrée dans un dossier LDAP

bool **ldap_add** ( resource link_identifier , string dn , array entry )

ldap_add sert à ajouter une entrée dans un dossier LDAP. dn est l'identification complète de l'entrée qui sera ajoutée, et entry est un tableau avec les informations sur la nouvelle entrée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Ces valeurs sont indexées individuellement. En cas de valeurs multiples pour un attribut, elle sont indexées numériquement, à partir de 0.

Exemple de tableau pour ajouter une entrée LDAP

```
<?php
   $entree["attribut1"] = "valeur";
   $entree["attribut2"][0] = "valeur1";
   $entree["attribut2"][1] = "valeur2";
?>
```

Exemple complet avec identification LDAP

```
<?php
$ds=ldap_connect("localhost");  // on suppose que le serveur LDAP est sur le serveur local

if ($ds) {
    // Connexion avec une identitée qui permet les modifications
    $r=ldap_bind($ds,"cn=root, o=Ma Compagnie, c=FR", "secret");

    // prepare les données
    $info["cn"]="Jean Dupont";
    $info["sn"]="Jean";
    $info["mail"]="jeand@serveur.mail.com";
    $info["objectclass"]="personne";

    // Ajoute les données au dossier
    $r=ldap_add($ds, "cn=Jean Dupont, o=Ma Compagnie, c=FR", $info);

    ldap_close($ds);
} else {
    echo "Impossible de se connecter au serveur LDAP";
 }
?>
```

## 6.16.11 ldap_bind() : Autentification au serveur LDAP

bool **ldap_bind** ( resource link_identifier , *string bind_rdn* , *string bind_password* )

ldap_bind s'autentifie auprès du serveur LDAP link_identifier , avec le nom d'utilisateur bind_rdn et le mot de passe bind_password . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_bind effecute une opération de bind avec le serveur. bind_rdn et bind_password sont optionnels. S'ils sont omis, une tentative de connexion anonyme est tentée.

Identification avec LDAP

```
<?php

// Eléments d'identification LDAP
$ldaprdn  = 'nom d\'utilisateur';     // DN ou RDN LDAP
$ldappass = 'mot de passe';           // Mot de passe associé

//Connexion au serveur LDAP
$ldapconn = ldap_connect("ldap.example.com")
    or die("Impossible de se connecter au serveur LDAP.");

if ($ldapconn) {

    //Connexion au serveur LDAP
    $ldapbind = ldap_bind($ldapconn, $ldaprdn, $ldappass);
```

```
    // Identification
    if ($ldapbind) {
        echo "Connexion LDAP réussie";
    } else {
        echo "Connexion LDAP échouée";
    }

}

?>
```

Connexion anonyme à un serveur LDAP

```
 <?php

//Connexion anonyme à un serveur LDAP

//Connexion au serveur LDAP
$ldapconn = ldap_connect("ldap.example.com")
    or die("Impossible de se connecter au serveur LDAP.");

if ($ldapconn) {

    // identification anonyme
    $ldapbind = ldap_bind($ldapconn);

    if ($ldapbind) {
        echo "Connexion LDAP anonmye réussie";
    } else {
        echo "Connexion LDAP anonmye échouée";
    }

 }

?>
```

## 6.16.12 ldap_close() : Ferme une connexion au serveur LDAP

bool **ldap_close** ( resource link_identifier )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_close ferme la connexion au serveur LDAP, représentée par link_identifier .

Cet appel est identique, en interne, à ldap_unbind . L'API LDAP utilise l'appel à ldap_unbind , alors peut être, devriez vous l'utiliser au lieu de ldap_close .

Note

ldap_close est un alias de ldap_unbind .

## 6.16.13 ldap_compare() : Compare une entrée avec des valeurs d'attributs

bool **ldap_compare** ( resource link_identifier , string dn , string attribute , string value )

ldap_compare retourne TRUE si value correspond, et sinon, FALSE . ldap_compare retourne −1 en cas d'erreur.

ldap_compare sert à comparer la valeur value de l'attribut attribute à la valeur du même attribut de l'entrée dn .

L'exemple suivant montre comment vérifier que deux mots de passe correspondent, dont l'un est celui d'une entré du serveur LDAP.

Exemple complet de vérification de mot de passe avec lDAP

```
 <?php
```

```
$ds=ldap_connect("localhost");  // doit être un serveur LDAP valide!

if ($ds) {

    // identification
     if(ldap_bind($ds)) {

        // preparation des données
        $dn = "cn=Matti Meikku, ou=Mon Service, o=Ma Compagnie, c=FR";
        $value = "secretpassword";
        $attr = "password";

         // comparaison des valeurs
        $r=ldap_compare($ds, $dn, $attr, $value);

        if ($r === -1) {
            echo "Erreur : ".ldap_error($ds);
        } elseif ($r === TRUE) {
            echo "Mot de passe correct.";
        } elseif ($r === FALSE) {
            echo "Mal choisi! Mot de passe incorrect!";
        }

    } else {
        echo "Impossible de se connecter au serveur LDAP.";
    }

    ldap_close($ds);

} else {
    echo "Impossible de se connecter au serveur LDAP.";
}
?>
```

Attention

ldap_compare ne peut PAS être utilisé pour comparer des valeurs binaires!

Note

ldap_compare a été ajoutée en 4.0.2.

## 6.16.14 ldap_connect() : Se connecte à un serveur LDAP

resource **ldap_connect** ( *string hostname* , *int port* )

ldap_connect retourne un identifiant positif de serveur LDAP en cas de succès, ou bien FALSE en cas d'erreur.

ldap_connect établit une connexion avec un serveur. Le serveur LDAP situé sur l'hôte hostname et port . Les deux arguments sont optionnels. Sans argument, l'identifiant de la dernière connexion ouverte sera retournée. Si seul hostname est spécifié, le port par défaut est 389.

Si vous utilisez OpenLDAP 2.x.x, vous pouvez spécifier une URL au lieu d'un nom d'hôte. Pour utiliser LDAP avec SSL, compilez OpenLDAP 2.x.x avec le support SSL, configurez PHP avec SSL, et utilisez ldaps://hostname/ comme nom d'hôte. Le paramètre de port port n'est pas utile lorsqu'utilisé avec des URL.

Note

Le support des URL et SSL a été ajouté en PHP 4.0.4.

Exemple de connexion à un serveur LDAP

```
 <?php

// LDAP variables
$ldaphost = "ldap.example.com";  // votre serveur LDAP
$ldapport = 389;                 // votre port de serveur LDAP

// Connecting to LDAP
$ldapconn = ldap_connect( $ldaphost, $ldapport )
           or die( "Impossible de se connecter au serveur LDAP {$ldaphost}" );

?>
```

Exemple de connexion à un serveur LDAP SSL

```
<?php

// Assurez vous que l'hôte est correct
// et que vous avez un certificat valide
$ldaphost = "ldaps://ldap.example.com/";

// Connexion à LDAP
$ldapconn = ldap_connect( $ldaphost )
           or die( "Impossible de se connecter au serveur LDAP {$ldaphost}" );

?>
```

## 6.16.15 ldap_count_entries() : Compte le nombre d'entrées après une recherche

int **ldap_count_entries** ( resource link_identifier , resource result_identifier )

**ldap_count_entries** retourne le nombre d'entrées trouvées dans le résultat result_identifier , sur la connexion link_identifier , ou FALSE en cas d'erreur.

ldap_count_entries retourne le nombre d'entrées stockées dans le résultat de la recherche précédente. result_identifier représente un résultat interne LDAP.

## 6.16.16 ldap_delete() : Efface une entrée dans un dossier

bool **ldap_delete** ( resource link_identifier , string dn )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_delete efface l'entrée identifiée par son dn sur le serveur identifié par link_identifier .

## 6.16.17 ldap_dn2ufn() : Convert it un DN en format UFN (User Friendly Naming)

string **ldap_dn2ufn** ( string dn )

ldap_dn2ufn convertit le DN dn dans un format plus lisible humainement, en supprimant les types des noms.

## 6.16.18 ldap_err2str() : Convertit un numéro d'erreur LDAP en message d'erreur

string **ldap_err2str** ( int errno )

ldap_err2str retourne un message d'erreur

ldap_err2str retourne le message lisible expliquant l'erreur dont le numéro est errno . Bien que les numéros d'erreur LDAP soient standardisés, différentes librairies retournent des messages différents ou même des textes d'erreurs localisés. N'utilisez jamais les messages d'erreur pour identifier une erreur, mais bien les numéros.

Enumerer tous les messages d'erreur LDAP

```
<?php
 for($i=0; $i<100; $i++) {
   printf("Erreur numéro $i : %s<br>\n", ldap_err2str($i));
 }
?>
```

Voir aussi ldap_errno et ldap_error .

## 6.16.19 ldap_errno() : Retourne le numéro d'erreur LDAP de la dernière commande exécutée.

int **ldap_errno** ( resource link_identifier )

ldap_errno retourne le numéro d'erreur LDAP généré par la dernière commande.

ldap_errno retourne le numéro d'erreur standard, généré par la dernière commande LDAP, pour la connexion link_identifier . Ce numéro peut être converti en message textuel avec ldap_err2str .

A moins que vous n'abaissiez suffisamment le niveau d'erreur dans php.ini (ou php3.ini ), ou que vous ne préfixiez vos commandes LDAP avec @ (at) pour supprimer les affichages, les erreurs LDAP s'afficheront aussi dans le code PHP.

Genérer et intercepter une erreur

```
<?php
// Cet exemple contient une erreur, que nous interecepterons.
$ld = ldap_connect("localhost");
$bind = ldap_bind($ld);
// erreur de syntaxe dans l'expression du filtre (errno 87),
// must be "objectclass=*" to work.
$res =  @ldap_search($ld, "o=Myorg, c=DE", "objectclass");
if (!$res) {
    printf("LDAP-Errno: %s<br>\n", ldap_errno($ld));
    printf("LDAP-Error: %s<br>\n", ldap_error($ld));
    die("Argh!<br>\n");
}
$info = ldap_get_entries($ld, $res);
printf("%d entrees trouvées.<br>\n", $info["count"]);
?>
```

Voir aussi ldap_err2str et ldap_error .

## 6.16.20 ldap_error() : Retourne le message LDAP de la dernière commande LDAP

string **ldap_error** ( resource link_identifier )

ldap_error retourne un message d'erreur LDAP.

ldap_error retourne le message d'erreur lié à la connexion link_identifier . Même si les numéros d'erreur LDAP sont standardisés, différentes librairies retournent différents messages, ou parfois, des messages en langue locale. Ne vous fiez pas au message d'erreur, mais bien au numéro d'erreur.

A moins que vous n'abaissiez suffisamment le niveau d'erreur dans php.ini (ou php3.ini ), ou que vous ne préfixiez vos commandes LDAP avec @ pour supprimer les affichages, les erreurs LDAP s'afficheront aussi dans le code PHP.

Voir aussi ldap_err2str et ldap_errno .

## 6.16.21 ldap_explode_dn() : Sépare les différents composants d'un DN

array **ldap_explode_dn** ( string dn , int with_attrib )

ldap_explode_dn sert à extraire les différents composants du DN dn . Chaque composant s'appelle un Nom Distingué Relatif (Relative Distinguished Name ou encore RDN). ldap_explode_dn retourne un tableau de tous ces composants. L'argument with_attrib sert à préciser si les RDN sont retournés seuls, ou bien avec leurs attributs. Pour obtenir les attributs en même temps que les RDN (au format attribut=valeur), donnez à with_attrib la valeur de 0 et sinon, donnez lui la valeur de 1.

## 6.16.22 ldap_first_attribute() : Return first attribute

string **ldap_first_attribute** ( resource link_identifier , resource result_entry_identifier , int ber_identifier )

ldap_first_attribute retourne le premier attribut de l'entrée result_entry_identifier , sur le serveur link_identifier , ou FALSE en cas d'erreur.

Similairement à la lecture des entrées, les attributs sont lis les uns après les auters, pour une entrée. ldap_first_attribute retourne le premier attribut de l'entrée représentée par result_entry_identifier . Les autres attributs sont lus grâce à la fonction ldap_next_attribute , appelée aussi souvent que nécessaire. Le paramètre ber_identifier est l'identifiant de pointeur en mémoire locale. Il est passé par référence. Le même paramètre ber_identifier est passé à la fonction ldap_next_attribute , qui modifiera ce pointeur.

Voir aussi ldap_get_attributes .

## 6.16.23 ldap_first_entry() : Retourne la première entrée

resource **ldap_first_entry** ( resource link_identifier , resource result_identifier )

ldap_first_entry retourne la première entrée du résultat result_identifier , pour la connexion link_identifier , ou FALSE en cas d'erreur.

Les entrées d'un résultat LDAP sont lues séquentiellement avec les fonctions ldap_first_entry et ldap_next_entry . ldap_first_entry retourne la première entrée. Les entrées suivantes seront lues grâce à la fonction lap_next_entry , en appelant cette fonction aussi souvent que nécessaire.

Voir aussi ldap_get_entries .

## 6.16.24 ldap_first_reference() : Return first reference

resource **ldap_first_reference** ( resource link , resource result )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.25 ldap_free_result() : Free result memory

bool **ldap_free_result** ( resource result_identifier )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_free_result libère toute la mémoire allouée en interne pour stocker le résultat result_identifier . Si l'appel de cette fonction est omis, toute la mémoire sera libérée automatiquement à la fin du script.

Typqieuemtn, toute la mémoire allouée pour le résultat LDAP est libérée à la fin du script. Si le script effectue des recherches intensives, qui retournent des résultats de grandes taille, ldap_free_result peut être utilisée pour réduire la consommation de mémoire.

## 6.16.26 ldap_get_attributes() : Lit les attributs d'une entrée

array **ldap_get_attributes** ( resource link_identifier , resource result_entry_identifier )

ldap_get_attributes retourne le détail des informations d'une entrée sous la forme d'un tableau multidimensionnel ou FALSE en cas d'erreur.

ldap_get_attributes sert à simplifier la lecture des attributs et de leur valeur pour une entrée dans un résultat de recherche. La valeur retournée est un tableau multidimensionnel, avec les noms des attributs et leurs valeurs.

Une fois que vous avez repéré une entrée dans un dossier, vous pouvez obtenir plus d'informations sur cette entrée avec cette fonction. Elle pourrait être utilisée dans le cadre d'une application qui cartographie les dossiers et les entrées. Dans de nombreuses applications, vous recherchez des entrées ayant un attribut précis, sans vous soucier des autres attributs.

Structure du tableau retourné par ldap_get_attributes

```
 return_value["count"] = nombre d'attributs dans l'entrée
return_value[0] = premier attribut
return_value[n] = n-ième attribut

return_value["attribute"]["count"] = nombre de valeur de l'attribut
return_value["attribute"][0] = première valeur de l'attribut
return_value["attribute"][i] = (i+1)-ème valeur de l'attribut
```

Affiche la liste des attributs d'une entrée

```
 <?php
// $ds est une ressource de connexion valide

// $sr est une recherche valide, issue d'une opération
// précédente

$entry = ldap_first_entry($ds, $sr);

$attrs = ldap_get_attributes($ds, $entry);

echo $attrs["count"]." attributs dans cette entrée :<p>";

for ($i=0; $i<$attrs["count"]; $i++)
    echo $attrs[$i]."<br>";
?>
```

Voir aussi ldap_first_attribute et ldap_next_attribute .

## 6.16.27 ldap_get_dn() : Lit le DN d'une entrée

string **ldap_get_dn** ( resource link_identifier , resource result_entry_identifier )

ldap_get_dn lit le DN de l'entrée courante dans le résultat de recherche LDAP result_entry_identifier , ou FALSE , en cas d'erreur.

ldap_get_dn sert à lire le DN dans un résultat de recherche.

## 6.16.28 ldap_get_entries() : Lit toutes les entrées du résultat

array **ldap_get_entries** ( resource link_identifier , resource result_identifier )

ldap_get_entries retourne toutes les entrées du résulat result_identifier , sous la forme d'un tableau multi−dimensionnel, ou FALSE en casd d'erreur.

ldap_get_entries sert à simplifier la lecture des entrées dans un résultat, ainsi que celle des attributs correspondants. La structure du tableau retourné la suivante :

L'index d'attribut est converti en minuscules (les attributs sont sensibles à la casse pour les serveurs de dossiers, mais ne le sont pas lorsqu'ils sont utilisés comme index de tableaux.

Structure du tableau retourné par ldap_get_entries

```
 return_value["count"] = nombre d'entrées dans le résultat
return_value[0] : se réfère au détails de la première entrée

return_value[n]["dn"] =  DN de la n−ième entrée du résultat

return_value[n]["count"] = nombre d'attributs de la n−ième entrée
return_value[n][m] = m−ième attribut de la i−ème entrée

return_value[n]["attribute"]["count"] = nombre de valeur des attributs de la i−ième entrée
return_value[n]["attribute"][m] = m−ième valeur de la n−ième entrée
```

Voir aussi ldap_first_entry et ldap_next_entry .

## 6.16.29 ldap_get_option() : Lit/écrit la valeur courante d'une option

bool **ldap_get_option** ( resource link_identifier , int option , mixed retval )

ldap_get_option modifie la valeur de l'option option à la valeur retval pour la connexion link_identifier . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Le paramètre option peut prendre l'une des valeurs suivantes : LDAP_OPT_DEREF, LDAP_OPT_SIZELIMIT, LDAP_OPT_TIMELIMIT, LDAP_OPT_PROTOCOL_VERSION, LDAP_OPT_ERROR_NUMBER, LDAP_OPT_REFERRALS, LDAP_OPT_RESTART, LDAP_OPT_HOST_NAME, LDAP_OPT_ERROR_STRING, LDAP_OPT_MATCHED_DN. Elles sont décrites dans draft−ietf−ldapext−ldap−c−api−xx.txt

Note

ldap_get_option n'est disponible qu'avec OpenLDAP 2.x.x OU Netscape Directory SDK x.x, et a été ajoutée en PHP 4.0.4.

Vérification de la version du protocole avec ldap_get_option

```
 // $ds doit être une ressource de connexion valide
if (ldap_get_option($ds, LDAP_OPT_PROTOCOL_VERSION, $version))
    echo "Nous utilisons le protocole version $version";
else
```

```
    echo "Impossible de déterminer la version du protocole.";
```

Voir aussi ldap_set_option .

## 6.16.30 ldap_get_values_len() : Lit toutes les valeurs binaires d'une entrée

array **ldap_get_values_len** ( resource link_identifier , resource result_entry_identifier , string attribute )

ldap_get_values_len retourne un tableau de valeurs pour l'attribut attribute en cas de succès, et FALSE en cas d'erreur.

ldap_get_values_len sert à lire toutes les valeurs de l'attribut attribute dans le résultat result_entry_identifier . L'entré utilisée et l'entrée courante du résultat de recherche LDAP result_entry_identifier . Le nombre de valeurs retournées est disponibles à l'index 'count' du tableau retourné. Les valeurs sont accessibles individuellement, avec les index numérique du tableau. L'indexation commence à 0.

Cette fonction d'utilise exactement comme ldap_get_values , hormis le fait qu'elle gère les données binaires, et non pas les chaînes de caractères.

| Note |
| --- |
| Cette fonction a été ajoutée en PHP 4.0. |

## 6.16.31 ldap_get_values() : Lit toutes les valeurs d'une entrée LDAP

array **ldap_get_values** ( resource link_identifier , resource result_entry_identifier , string attribute )

ldap_get_values retourne un tableau de valeurs pour l'attribut attribute , ou FALSE en cas d'erreur.

ldap_get_values sert à lire toutes les valeurs de l'attibut attribute , dans l'entrée courante du résultat de recherche result_entry_identifier , pour la connexion link_identifier . Le nombre de valeurs retournées est disponibles à l'index 'count' du tableau retourné. Les valeurs sont accessibles individuellement, avec les index numérique du tableau. L'indexation commence à 0.

L'utilisation de la fonction result_entry_identifier requiert un résultat de recherche, et doit donc être précédé d'une recherche LDAP, et de l'une des fonctions permettant d'accéder à une entrée.

Votre application doit contenir des informations permettant de lire certains attributs (comme "nom" ou "mail"), ou bien vous devrez utiliser la fonction ldap_get_attributes pour savoir quels sont les attributs qui existent pour une entrée donnée.

LDAP permet plus d'une entrée par attribut, ce qui permet de stocker plusieurs addresse emails par personne, tout en n'utilisant qu'une étiquette "mail" :

| Structure du tableau retourné par ldap_get_values |
| --- |
| return_value["count"] = nombre de valeur de l'attribut<br>return_value[0] = première valeur de l'attribut<br>return_value[i] = n-ième valeur de l'attribut |

| Liste toutes les valeurs de l'attribut "mail" d'une entrée |
| --- |

```
 // $ds doit être une ressource de connexion valide

// $sr doit être une ressource de résultat valide, obtenue avec une des fonctions de
//     recherche LDAP.

// $entry est une entrée LDAP valide, obtenue avec une des fonctions
//        LDAP qui retourne une entrée

$values = ldap_get_values($ds, $entry,"mail");

echo $values["count"]." adresses email pour cette entrée.<p>";

for ($i=0; $i < $values["count"]; $i++)
    echo $values[$i]."<br>";
```

## 6.16.32 ldap_list() : Recherche dans un niveau

resource **ldap_list** ( resource link_identifier , string base_dn , string filter , *array attributes* , *int attrsonly* , *int sizelimit* , *int timelimit* , *int deref* )

ldap_list retourne un identfiant de résultat ou bien FALSE en cas d'erreur.

ldap_list effectue une recherche avec le filtre filter dans le dossier base_dn avec l'option LDAP_SCOPE_ONELEVEL .

LDAP_SCOPE_ONELEVEL signifie que la recherche ne peut retourner des entrées que dans le niveau qui est immédiatement sous le niveau base_dn (c'est l'équivalent de la commande "ls", pour obtenir la liste des fichiers et dossiers du dossier courant.

ldap_list prend 5 paramtères optionnels. Voir ldap_search pour connaître le détails de leur utilisation.

| Note |
| --- |
| Ces paramètres optionnels ont été ajoutés en 4.0.2 : attrsonly , sizelimit , timelimit et deref . |

Produit une liste de tous les services d'une société avec ldap_list

```
 <?php
// $ds doit être une ressource de connexion valide

  $basedn = "o=Ma Compagnie, c=FR";
  $justthese = array("ou");

  $sr=ldap_list($ds, $basedn, "ou=*", $justthese);

  $info = ldap_get_entries($ds, $sr);

  for ($i=0; $i<$info["count"]; $i++){
    echo $info[$i]["ou"][0] ;
  }
?>
```

| Note |
| --- |
| Depuis PHP 4.0.5, il est aussi possible de faire des recherches paralleles. Reportez vous à ldap_search pour plus de détails. |

## 6.16.33 ldap_mod_add() : Ajoute un attribut à l'entrée courante

bool **ldap_mod_add** ( resource link_identifier , string dn , array entry )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_mod_add ajoute l'attribut entry à l'entrée dn . Elle effectue la modification au niveau attribut, par opposition au niveau objet. Les additions au niveau objet sont réalisées par ldap_add .

## 6.16.34 ldap_mod_del() : Efface un attribut à l'entrée courante

bool **ldap_mod_del** ( resource link_identifier , string dn , array entry )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_mod_del efface l'attribut entry de l'entrée dn . Elle effectue la modification au niveau attribut, par opposition au niveau objet. Les additions au niveau objet sont réalisées par ldap_delete .

## 6.16.35 ldap_mod_replace() : Remplace un attribut dans l'entrée courante

bool **ldap_mod_replace** ( resource link_identifier , string dn , array entry )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_mod_replace remplace l'attribut entry de l'entrée dn . Elle effectue le remplacement au niveau attribut, par opposition au niveau objet. Les additions au niveau objet sont réalisées par ldap_modify .

## 6.16.36 ldap_modify() : Modifie une entrée LDAP

bool **ldap_modify** ( resource link_identifier , string dn , array entry )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_modify modifie l'entrée identifiée par dn , avec les valeurs fournies dans entry . La structure de entry est la même que détaillées dans ldap_add .

## 6.16.37 ldap_next_attribute() : Lit l'attribut suivant

string **ldap_next_attribute** ( resource link_identifier , resource result_entry_identifier , resource ber_identifier )

ldap_next_attribute retourne l'attribut suivant en cas de succès, et sinon, une erreur.

ldap_next_attribute sert à lire tous les attributs d'une entrée. Le pointeur interne est géré par ber_identifier . Il est passé par référence à la fonction. Le premier appel à ldap_next_attribute est fait avec le result_entry_identifier retourné par ldap_first_attribute .

Voir aussi ldap_get_attributes .

## 6.16.38 ldap_next_entry() : Lit la prochaine entrée

resource **ldap_next_entry** ( resource link_identifier , resource result_entry_identifier )

ldap_next_entry retourne un identifiant d'entrée, pour la prochaine entrée dans le résultat result_entry_identifier . La première entrée a été lue avec la fonction ldap_first_entry . Si il n'y a plus d'entrées dans le résultat, ldap_next_entry retourne FALSE .

ldap_next_entry sert à lire successivement les entrées stockées dans le résultat result_entry_identifier . Des appels répétés à ldap_next_entry retourneront toutes les entrées jusqu'à ce qu'il n'y en ait plus. Le premier appel doit être fait avec la fonction ldap_first_entry . Le paramètre result_entry_identifier est celui qui a été retourné par la fonction ldap_first_entry .

Voir aussi ldap_get_entries .

## 6.16.39 ldap_next_reference() : Lit la référence suivante

resource **ldap_next_reference** ( resource link , resource entry )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.40 ldap_parse_reference() : Extrait les informations d'une référence d'entrée

bool **ldap_parse_reference** ( resource link , resource entry , array referrals )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.41 ldap_parse_result() : Extrait des informations d'un résultat

bool **ldap_parse_result** ( resource link , resource result , int errcode , string matcheddn , string errmsg , array referrals )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.42 ldap_read() : Lit une entrée

resource **ldap_read** ( resource link_identifier , string base_dn , string filter , *array attributes* , *int attrsonly* , *int sizelimit* , *int timelimit* , *int deref* )

ldap_read retourne un identifiant de ressource, ou FALSE en cas d'erreur.

ldap_read effectue une recherche avec le filtre filter dans le dossier base_dn avec la configuration LDAP_SCOPE_BASE . C'est équivalent à lire une entrée dans un dossier.

Un filtre ne peut être vide. Si vous voulez lire toutes les informations d'une entrée, utilisez le filtre " objectClass=* ". Si vous savez quels sont les types qui sont utilisés dans le serveur de dossiers, vous pouvez aussi utiliser un filtre approprié, comme par exemple " objectClass=inetOrgPerson ".

ldap_read prend 5 paramètres optionnels. Reportez vous à ldap_search pour plus de détails.

| Note |
| --- |
| Ces paramètres optionnels ont été ajoutés en PHP 4.0.2 : attrsonly , sizelimit , timelimit et deref . |

Depuis PHP 4.0.5, il est aussi possible d'effectuer des recherches paralelles. Voyez ldap_search poour plus de détails.

## 6.16.43 ldap_rename() : Modifie le nom d'une entrée

bool **ldap_rename** ( resource link_identifier , string dn , string newrdn , string newparent , bool deleteoldrdn )

ldap_rename modifie l'entrée dn , autant pour son nom que pour sa localisation. Le nouveau RDN est spécifié avec newrdn et le nouveau père est spécifié avec newparent . Si le paramètre deleteoldrdn vaut TRUE , l'ancienne valeur RDN est supprimée. Sinon l'ancienne valeur RDN est conservée comme une valeur non distinguée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| ldap_rename ne fonctionne actuellement qu'avec LDAPv3. Vous pouvez être obligé d'utiliser ldap_set_option avant de vous lier pour pouvoir utiliser LDAPv3. Cette fonction est uniquement disponible lorsque vous utilisez OpenLDAP 2.x.x OU Netscape Directory SDK x.x. Elle a été ajoutée en PHP 4.0.5. |

## 6.16.44 ldap_search() : Recherche sur le serveur LDAP

resource **ldap_search** ( resource link_identifier , string base_dn , string filter , *array attributes* , *int attrsonly* , *int sizelimit* , *int timelimit* , *int deref* )

**ldap_search** retourne un identifiant de résultat, ou bien FALSE en cas d'erreur.

ldap_search effectue une recherche avec le filtre filter dans le dossier base_dn avec le paramétrage LDAP_SCOPE_SUBTREE . C'est l'équivalent d'une recherche dans le dossier. base_dn spécifie le DN de base du dossier.

Il y a un quatrième paramètre optionnel, qui peut être fourni pour restreindre les attributs et valeurs retournées par le serveur, afin de les adapter à vos besoins. C'est bien plus efficace que le comportement par défaut (qui retourne tous les attributs et leurs valeurs associées). L'utilisation de ce quatrième est donc vivement recommandé.

Le quatrième paramtère est un tableau de chaînes PHP avec les attributs demandé, comme par exemple : array("mail","sn","cn") . Notez que le "dn" est toujours retourné, indépendamment des attributs demandés.

Notez aussi que certains serveurs de dossiers sont configurés pour retourner un nombre limité d'entrées. Si cette situation survient, le serveur indiquera qu'il a retourné un résultat partiel. C'est aussi le cas si le sixième paramètre sizelimit a été utilisé pour limiter le nombre d'entrées retournées.

Le cinquième paramètre attrsonly doit être mis à 1 si seul les types d'attributs sont désirés. Si ce paramètre est mis à 0, les types d'attributs et leurs valeurs seront retournées : c'est le comportement par défaut.

Avec le sixième paramètre sizelimit , il est possible de limiter le nombre d'entrées lues dans le résultat. En utilisant 0, on indique qu'il n'y a pas de limite. NOTE : ce paramètre NE peut PAS remplacer la configuration du serveur. Vous pouvez au mieux baisser encore cette limite.

Le septième paramètre timelimit permet de spécifier le nombre de secondes de duréee de la recherche. En utilisant la valeur 0, la recherche pourra durée indéfiniment. NOTE : ce paramètre NE peut PAS remplacer la configuration du serveur. Vous pouvez au mieux baisser encore cette limite.

Le huitième paramètre deref spécifie comment les alias doivent être géré durant la rechercheé. Ce paramètre peut prendre l'une des valeurs suivantes :

- LDAP_DEREF_NEVER − (par défaut) les alias ne sont jamais déréférencés.
- LDAP_DEREF_SEARCHING − les alias sont déréférencés durant la recherche, mais pas durant la localisation du dossier de recherche.
- LDAP_DEREF_FINDING − les alias sont déréférencés durant la localisation du dossier de recherche, mais pas durant la recherche.
- LDAP_DEREF_ALWAYS − les alias sont toujours déréférencés.

| Note |
| --- |
| Ces paramètres optionnels ont été ajoutés en 4.0.2 : attrsonly , sizelimit , timelimit et deref . |

Le filtre de recherche peut être simple ou avancé, et utiliser ces opérateurs booléen au format décrit dans la documentation LDAP (reportez vous à Netscape Directory SDK pour plus d'informations sur les filtres).

L'exemple ci−dessous lit le nom du service, le nom, le prénom et le email des employés de la société "Ma Compagnie", dont le nom ou le prénom contient la sous chaîne : $person. Cet exemple illustre l'utilisation de filtres pour indiquer au serveur de faire une recherche sur deux attributs.

Recherche LDAP

```
 <?php
// $ds doit être une ressource de connexion valide

// $person est un nom ou une partie de nom (par exemple, "Jean")

$dn = "o=Ma Compagnie, c=FR";
$filter="(|(sn=$person*)(givenname=$person*))";
$justthese = array( "ou", "sn", "givenname", "mail");

$sr=ldap_search($ds, $dn, $filter, $justthese);

$info = ldap_get_entries($ds, $sr);

print $info["count"]." entrées trouvées.<p>";
?>
```

Depuis PHP 4.0.5, il est possible de faire des recherches paralelles. Pour cela, vous devez utiliser un tableau de ressources de connexion comme premier argument, au lieu d'une ressource simple. Si vous ne voulez pas utiliser le même dossier de base DN et le même filtre pour toutes les recherches, vous pouvez aussi utiliser un tableau de DN et de filtres. Ces tableaux doivent alors etre de la même taille que le table de ressources de connexion : la première entrée du tableau de ressources sera utilisé avec le premier DN et le premier filtre, etc... Lorsque vous faite une recherche paralelle, un tableau de résultats est retourné, sauf en cas d'erreur, auquel cas l'entrée contiendra FALSE . C'est très proche du comportement habituel, hormis le fait qu'un résultat est toujours retourné lorsqu'une recherche est faite. Il y a de rares cas où une recherche retourne FALSE , alors qu'une recherche en paralelle retourne un identifiant.

## 6.16.45 ldap_set_option() : Modifie la valeur d'une option LDAP

bool **ldap_set_option** ( resource link_identifier , int option , mixed newval )

**ldap_set_option** modifie la valeur de l'option option en remplacant la valeur courante par newval . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Le paramètre option peut prendre l'une des valeurs suivantes : LDAP_OPT_DEREF, LDAP_OPT_SIZELIMIT, LDAP_OPT_TIMELIMIT, LDAP_OPT_PROTOCOL_VERSION, LDAP_OPT_ERROR_NUMBER, LDAP_OPT_REFERRALS, LDAP_OPT_RESTART, LDAP_OPT_HOST_NAME, LDAP_OPT_ERROR_STRING, LDAP_OPT_MATCHED_DN, LDAP_OPT_SERVER_CONTROLS, LDAP_OPT_CLIENT_CONTROLS. Voici une brève description des valeurs ci−dessus, dans le fichier draft−ietf−ldapext−ldap−c−api−xx.txt .

Les options LDAP_OPT_DEREF, LDAP_OPT_SIZELIMIT, LDAP_OPT_TIMELIMIT, LDAP_OPT_PROTOCOL_VERSION et LDAP_OPT_ERROR_NUMBER doivent être des entiers, LDAP_OPT_REFERRALS et LDAP_OPT_RESTART doivent être des booléens, et LDAP_OPT_HOST_NAME, LDAP_OPT_ERROR_STRING et LDAP_OPT_MATCHED_DN doivent être des chaînes. Le premier exemple illustre leur utilisation. Les options LDAP_OPT_SERVER_CONTROLS et LDAP_OPT_CLIENT_CONTROLS requièrent une liste de contrôles, ce qui signifie que la valeur doit être un tableau de contrôles. Un contrôle est constitués d'un **oid** identifiant le contrôle, une valeur optionnelle **value** , et un flag optionnel de criticalité ( **criticality** ). En PHP, un contrôle est définit sous la forme d'un tableau, donc les clés sont **oid** avec une chaîne comme valeur, et deux clés optionnelles. Ces clés sont **value** avec une chaîne comem valeur, et **iscritical** avec une valeur booléenne. Par défaut, **iscritical** vaut FALSE . Reportez vous au second exemple pour une illustration.

| Note |
| --- |
| Cette fonction n'est disponible que lorsque vous utilisez OpenLDAP 2.x.x ou Netscape Directory SDK x.x. Elle a été ajoutée en PHP 4.0.4. |

Set protocol version

```php
<?php
// $ds doit être une ressource de connexion valide
if (ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3))
    echo "Version LDAPv3";
else
    echo "Impossible de modifier la version du protocole à 3";
?>
```

Set server controls

```php
// $ds doit être une ressource de connexion valide
// controle sans valeur
$ctrl1 = array("oid" => "1.2.752.58.10.1", "iscritical" => TRUE);
// iscritical vaut par défaut FALSE
$ctrl2 = array("oid" => "1.2.752.58.1.10", "value" => "magic");
// essayez d'utiliser les deux contrôles
if (!ldap_set_option($ds, LDAP_OPT_SERVER_CONTROLS, array($ctrl1, $ctrl2)))
    echo "Impossible de modifier les contrôles du serveur";
```

Voir aussi ldap_get_option .

## 6.16.46 ldap_set_rebind_proc() : Configure une fonction de callback pour refaire des liaisons, lors de recherche de référants.

bool **ldap_set_rebind_proc** ( resource link , string callback )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.47 ldap_sort() : Trie les entrées d'un résultat LDAP

bool **ldap_sort** ( resource link , resource result , string sortfilter )

| Attention |
| --- |

| |
|---|
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.48 ldap_start_tls() : Démarre TLS

bool **ldap_start_tls** ( resource link )

| Attention |
|---|
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.49 ldap_t61_to_8859() : Convertit les caractères t6 en caractères 8859

string **ldap_t61_to_8859** ( string value )

| Attention |
|---|
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.16.50 ldap_unbind() : Déconnecte d'un serveur LDAP

bool **ldap_unbind** ( resource link_identifier )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

ldap_unbind assure la déconnexion du serveur LDAP.

# 6.17 Mail

## 6.17.1 Introduction

La fonction mail permet d'envoyer du mail.

## 6.17.2 Pré−requis

Pour que la fonction mail soit disponible, il faut que PHP ait access au service sendmail sur le serveur, au moment de la compilation. Si vous utilisez un autre programme de mail, comme qmail ou postfix , assurez vous d'utiliser les bonnes API. PHP va commencer à chercher sendmail dans votre PATH , puis, dans les dossiers suivants : /usr/bin:/usr/sbin:/usr/etc:/etc:/usr/ucblib:/usr/lib . Il est hautement recommandé d'avoir sendmail de disponible dans votre PATH . De plus, l'utilisateur qui compile PHP doit avoir le droit d'accéder à l'exécutable sendmail.

## 6.17.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.17.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| SMTP | "localhost" | PHP_INI_ALL |
| smtp_port | "25" | PHP_INI_ALL |
| sendmail_from | NULL | PHP_INI_ALL |
| sendmail_path | DEFAULT_SENDMAIL_PATH | PHP_INI_SYSTEM |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*smtp string*
: Sous Windows seulement : adresse IP ou nom que PHP doit utiliser pour envoyer du mail avec la fonction mail .

*smtp int*
: Sous Windows seulement : numéro de port à utiliser pour se connecter au serveur SMTP lors de l'envoi de mail avec la fonction mail ; par défaut, c'est 25 . Uniquement disponible depuis PHP 4.3.0.

*sendmail_from string*
: Sous Windows seulement : valeur du champ "From:" qui doit être utilisée lors de l'envoi de mail.

*sendmail_path string*
: Localisation du programme de sendmail : habituellement /usr/sbin/sendmail ou /usr/lib/sendmail . configure essaye de repérer la présence de sendmail par lui−même, et affecte ce résultat par défaut. En cas de problème de localisation, vous pouvez établir une nouvelle valeur par défaut ici.

Tout système n'utilisant pas sendmail doit établir cette directive à la valeur chemin du programme de substitution qui remplace le serveur de mail, si celui−ci existe, par exemple, Qmail . Dans ce cas là, vous devez mettre: /var/qmail/bin/sendmail .

qmail−inject ne requiert aucune option pour traiter correctement le mail.

## 6.17.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.17.6 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- ezmlm_hash
- mail


## 6.17.8 mail() : Envoie de mail

bool **mail** ( string to , string subject , string message , *string additional_headers* , *string additional_parameters* )

mail poste automatiquement le message message à destination de to . Les destinataires multiples doivent être séparés par des virgules. Les emails avec pièces jointes ou contenus particuliers (comme les emails en HTML, par exemple), peuvent être réalisés avec cette fonction. Il faut respecter l'encodage MIME. Pour plus de détails, voyez http://www.zend.com/zend/spotlight/sendmimeemailpart1.php et les classes PEAR MIME .

Les RFC suivantes peuvent aussi se révéler utiles : RFC 1896 , RFC 2045 , RFC 2046 , RFC 2047 , RFC 2048 et RFC 2049 .

mail retourne TRUE si le mail est envoyé, et FALSE sinon.

Envoie de courrier électronique (mail)

```
<?php
 mail("maxou@exemple.com", "Mon Sujet", "Ligne 1\nLigne 2\nLigne 3");
?>
```

Le quatrième argument passé sera inséré à la fin de l'en−tête. Typiquement, cela permet d'insérer des en−têtes supplémentaires. Les en−têtes multiples doivent être séparées par des virgules.

Note

Sous Windows 32bits, vous devez utiliser \r\n pour séparer les en−têtes. Notez aussi que les en−têtes cc: et bcc: sont sensibles à la casse et doivent être écrits Cc: et Bcc: sous Win32.

Si le cinquième argument additional_parameters est fourni, PHP l'utilisera dans son appel du programme d'envoie de courrier électronique. Ceci est pratique pour passer une valeur correcte à l'en−tête Return−Path , avec sendmail.

Note

Le cinquième paramètre a été ajouté en PHP 4.0.5.

Envoi de eMail avec des en−têtes supplémentaires

```
 <?php
mail("personne@exemple.com", "Le sujet", $message,
     "From: webmaster@{$_SERVER['SERVER_NAME']}\r\n"
    ."Reply-To: webmaster@{$_SERVER['SERVER_NAME']}\r\n"
    ."X-Mailer: PHP/" . phpversion());
?>
```

Avec le cinquième paramètre, vous pouvez ajouter d'autres paramètres de ligne de commande qui seront utilisés par le programme d'envoi de courrier. Dans l'exemple ci−dessous, l'en−tête Return−Path est correctement paramétré. Normalement, sendmail ajoute automatiquement l'en−tête X−Authentication−Warning (paramètre −f), car l'utilisateur "serveur web" n'est probablement pas un de ses utilisateurs de confiance ("trusted users"). Pour supprimer cette alerte, ajoutez l'utilisateur du serveur web dans la configuration de sendmail.

Envoi de eMail avec des en−têtes supplémentaires et un paramètre de ligne de commande supplémentaire

```
 <?php

mail("personne@exemple.com", "Le sujet", $message,
     "From: webmaster@{$_SERVER['SERVER_NAME']}", "-fwebmaster@{$_SERVER['SERVER_NAME']}");
?>
```

Vous pouvez aussi utiliser des techniques simples de concaténations de chaînes pour construire des messages complexes :

Envoi de mail complexe avec mail

```
 <?php
/* destinataire */
$to  = "Mary <mary@example.com>" . ", " ; // note the comma
$to .= "Kelly <kelly@example.com>";

/* sujet */
$subject = "Anniversaires à souhaiter en Août";

/* message */
$message = "
<html>
<head>
 <title>Rappel des anniversaires d'Août</title>
</head>
<body>
<p>Voici la liste des anniversaires du mois d\'Août!</p>
<table>
 <tr>
  <th>Ami</th><th>Jour</th><th>Mois</th><th>Année</th>
 </tr>
 <tr>
  <td>Guillaume</td><td>3</td><td>Août</td><td>1970</td>
 </tr>
 <tr>
  <td>Sandra</td><td>17</td><td>Août</td><td>1973</td>
 </tr>
</table>
</body>
</html>";

/* Pour envoyer du mail au format HTML, vous pouvez configurer le type Content-type. */
$headers  = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";

/* D'autres en-têtes : errors, From cc's, bcc's, etc */
$headers .= "From: Birthday Reminder <birthday@example.com>\r\n";

$headers .= "Cc: birthdayarchive@example.com\r\n";
$headers .= "Bcc: birthdaycheck@example.com\r\n";

/* et hop, à la poste */
mail($to, $subject, $message, $headers);
?>
```

Note

Assurez−vous qu'il n'y ait aucune nouvelle ligne (ou d'autres espaces ou caractères blancs) dans les paramètres to ou subject , car cela peut avoir des effets secondaires irrationnels.

| Note |
| --- |
| Le paramètre to ne peut pas être une adresse de la forme "Quelqu'un <quidam@exemple.com>". La commande mail ne comprendra pas ce format. |

Voir aussi imap_mail .

# 6.18 Mathématiques

## 6.18.1 Introduction

Ces fonctions ne sont capables de manipuler que des entiers integer , ou nombres à virgule flottante ( float ). Si vous avez besoin de manipuler des nombres plus grands, reportez−vous aux fonctions mathématiques sur des nombres de grande taille .

## 6.18.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.18.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.18.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.18.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.18.6 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

| Constante | Valeur | Description |
|---|---|---|
| M_PI | 3.14159265358979323846 | Pi |
| M_E | 2.7182818284590452354 | e |
| M_LOG2E | 1.4426950408889634074 | log_2 e |
| M_LOG10E | 0.43429448190325182765 | log_10 e |
| M_LN2 | 0.69314718055994530942 | log_e 2 |
| M_LN10 | 2.30258509299404568402 | log_e 10 |
| M_PI_2 | 1.57079632679489661923 | pi/2 |
| M_PI_4 | 0.78539816339744830962 | pi/4 |
| M_1_PI | 0.31830988618379067154 | 1/pi |
| M_2_PI | 0.63661977236758134308 | 2/pi |
| M_SQRTPI | 1.77245385090551602729 | sqrt(pi) [4.0.2] |
| M_2_SQRTPI | 1.12837916709551257390 | 2/sqrt(pi) |
| M_SQRT2 | 1.41421356237309504880 | sqrt(2) |
| M_SQRT3 | 1.73205080756887729352 | sqrt(3) [4.0.2] |
| M_SQRT1_2 | 0.70710678118654752440 | 1/sqrt(2) |
| M_LNPI | 1.14472988584940017414 | log_e(pi) [4.0.2] |
| M_EULER | 0.57721566490153286061 | Euler constant [4.0.2] |

Seule M_PI est disponible dans les verisons de PHP jusqu'à 4.0.0. Toutes les autres constantes on été ajoutés à partir de PHP 4.0.0, et les autres ont été indiquées avec [4.0.2].

- Introduction

- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- abs
- acos
- acosh
- asin
- asinh
- atan
- atan2
- atanh
- base_convert
- bindec
- ceil
- cos
- cosh
- decbin
- dechex
- decoct
- deg2rad
- exp
- expm1
- floor
- fmod
- getrandmax
- hexdec
- hypot
- is_finite
- is_infinite
- is_nan
- lcg_value
- log
- log10
- log1p
- max
- min
- mt_getrandmax
- mt_rand
- mt_srand
- octdec
- pi
- pow
- rad2deg
- rand
- round
- sin
- sinh
- sqrt
- srand
- tan
- tanh

## 6.18.8 acos() : arc cosinus

float **acos** ( float arg )

acos retourne l'arc cosinus de arg ( arg en radians). acos est la fonction inverse de cos , ce qui signifie que a==cos(acos(a)) pour toute valeur qui soit dans l'intervalle de validité de acos '.

Voir aussi acosh , asin et atan .

## 6.18.9 acosh() : Arc cosinus hyperbolique

float **acosh** ( float arg )

acosh retourne l'arc cosinus hyperbolique de arg , c'est à dire la valeur dont le cosinus hyperbolique est arg .

| Note |
| --- |
| Cette fonction n'est pas implémentée sous Windows. |

Voir aussi acos , asinh et atanh .

## 6.18.10 asin() : arc sinus

float **asin** ( float arg )

asin retourne l'arc sinus de arg ( arg en radians). asin est la fonction inverse de sin , ce qui signifie que a==sin(asin(a)) pour toute valeur qui soit dans l'intervalle de validité de asin .

Voir aussi asinh , acos et atan .

## 6.18.11 asinh() : Arc sinus hyperbolique

float **asinh** ( float arg )

asinh retourne l'arc sinus hyperbolique de arg , c'est à dire la valeur dont le sinus hyperbolique est arg .

| Note |
| --- |
| Cette fonction n'est pas implémentée sous Windows. |

Voir aussi asin , acosh et atanh .

## 6.18.12 atan() : arc tangent

float **atan** ( float arg )

atan retourne l'arc tangent de arg ( arg en radians). atan est la fonction inverse de tan , ce qui signifie que a==tan(atan(a)) pour toute valeur qui soit dans l'intervalle de validité de atan .

Voir aussi atanh , asin et acos .

## 6.18.13 atan2() : Arc tangent de deux variables

float **atan2** ( float y , float x )

atan2 retourne l'arc tangent de deux variables x et y . La formule est : " arc tangent (y / x) ", et les signes des arguments sont utilisés pour déterminer le quadrant du résultat.

atan2 retourne un résultat en radians, entre −PI et PI (inclus).

Voir aussi acos et atan .

## 6.18.14 atanh() : Arc tangeant hyperbolique

float **atanh** ( float arg )

atanh retourne l'arc tangente hyperbolique de arg , c'est à dire la valeur dont la tangente hyperbolique est arg .

| Note |
| --- |
| Cette fonction n'est pas implémentée sous Windows. |

Voir aussi atan , asinh et acosh .

## 6.18.15 base_convert() : Convertit un nombre entre des bases arbitraires

string **base_convert** ( string number , int frombase , int tobase )

base_convert retourne une chaîne contenant l'argument number représenté dans la base tobase . La base de représentation de number est donnée par frombase . frombase et tobase doivent être compris entre 2 et 36 inclus. Les chiffres supérieurs à 10 des bases supérieures à 10 seront représentés par les lettres de A à Z, avec A = 10 et Z = 36.

Exemple avec base_convert

```
<?php
  $binary = base_convert($hexadecimal, 16, 2);
?>
```

affichera

```
101000110111001100110100
```

## 6.18.16 bindec() : Convertit de binaire en décimal

int **bindec** ( string binary_string )

bindec retourne la conversion d'un nombre binaire représenté par la chaîne binary_string en décimal.

bindec convertit un nombre binaire en décimal. Le plus grand nombre convertible a 31 bits à 1, soit 2147483647 en décimal.

Voir aussi decbin , octdec , hexdec et base_convert .

## 6.18.17 ceil() : Arrondit au nombre supérieur

float **ceil** ( float value )

ceil retourne l'entier supérieur du nombre value . Utiliser ceil sur un entier ne sert à rien. La valeur retournée est un nombre à virgule flottante ( float ), car ces nombres peuvent être plus grands que les entiers.

Exemple avec ceil

```
<?php
 echo ceil(4.3);    // 5
 echo ceil(9.999);  // 10
?>
```

Voir aussi floor et round .

## 6.18.18 cos() : cosinus

float **cos** ( float arg )

cos retourne le cosinus de arg ( arg en radians).

Voir aussi acos , sin tan et deg2rad .

## 6.18.19 cosh() : Cosinus hyperbolic

float **cosh** ( float arg )

cosh retourne le cosinus hyperbolique de arg , définit comme (exp(arg) + exp(−arg))/2 .

Voir aussi cos , acosh , sin et tan .

## 6.18.20 decbin() : Convertit de décimal en binaire

string **decbin** ( int number )

decbin retourne une chaîne contenant la représentation binaire de l'entier donné en argument. Le plus grand nombre pouvant être converti est 2147483647 en décimal, ce qui donne une série de 31 uns (1).

Voir aussi bindec , decoct , dechex et base_convert .

## 6.18.21 dechex() : Convertit de décimal en hexadécimal

string **dechex** ( int number )

dechex retourne une chaîne contenant la représentation hexadécimale du nombre number . Le nombre le plus grand qui puisse être converti est 2147483647 en décimal, ce qui donnera "7fffffff".

Voir aussi hexdec , decbin , decoct et base_convert .

## 6.18.22 decoct() : Convertit de décimal en octal

string **decoct** ( int number )

decoct retourne une chaîne contenant la représentation octale du nombre donné number . Le nombre le plus grand qui puisse être converti est 2147483647 en décimal, ce qui donnera "17777777777".

Voir aussi octdec , decbin , dechex et base_convert .

## 6.18.23 deg2rad() : Convertit un nombre de degrés en radians

float **deg2rad** ( float number )

deg2rad convertit number de degrés en radians.

Voir aussi rad2deg .

## 6.18.24 exp() : Exponentielle

float **exp** ( float arg )

exp retourne l'exponentielle de arg , c'est−à−dire e élevé à la puissance arg .

Voir aussi pow et log .

## 6.18.25 expm1() : Retourne le résultat de exp(number) − 1, calculé de manière précise, même si val est proche de 0

float **expm1** ( float number )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.18.26 floor() : Arrondit à l'entier inférieur

float **floor** ( float value )

floor retourne l'entier inférieur du nombre value . La valeur retournée est un nombre à virgule flottante, ( float ) car ces nombres peuvent être plus grands que les entiers.

Exemple avec floor

```
<?php
 echo floor(4.3);   // 4
 echo floor(9.999); // 9
?>
```

Voir aussi ceil et round .

## 6.18.27 fmod() : Retourne le reste de la division

float **fmod** ( float x , float y )

**fmod** retourne le reste de la division de ( x ) par ( y ). Ce reste est un nombre à virgule flottante. Le reste est défini par : x = i * y + r, pour un entier i . Si y n'est pas nul, r a le même signe que x est une taille inférieure à y .

Exemple avec fmod

```
 $x = 5.7;
$y = 1.3;
$r = fmod($x, $y);
// $r égale 0.5, car 4 * 1.3 + 0.5 = 5.7
```

## 6.18.28 getrandmax() : Plus grande valeur aléatoire possible

int **getrandmax** ( void )

getrandmax retourne la plus grande valeur aléatoire possible retournée par rand .

Voir aussi rand , srand mt_rand , mt_srand et mt_getrandmax .

## 6.18.29 hexdec() : Convertit de hexadécimal en décimal

int **hexdec** ( string hex_string )

hexdec retourne une chaîne contenant la représentation décimale du nombre hex_string . Le nombre le plus grand qui puisse être converti est 7fffffff en décimal, ce qui donne "2147483647".

hexdec remplace tous les caractères non−hexadécimaux par des 0 . Et si les zéros de gauche sont ignorés, ceux de droite prennent le propre valeur.

Exemple avec hexdec

```
<?php
 var_dump(hexdec("Hop comme ceci"));
 var_dump(hexdec("0000c000e0cec0"));
 var_dump(hexdec("c000e0cec0"));
// les deux affichent "int(14732992)"
 var_dump(hexdec("aussi"));
 var_dump(hexdec("a0000"));
// les deux affichent "int(655360)"
?>
```

Voir aussi dechex , bindec , octdec et base_convert .

## 6.18.30 hypot() : Retourne le résultat de sqrt( num1*num1 + num2*num2)

float **hypot** ( float num1 , float num2 )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.18.31 is_finite() : Indique si un nombre est fini

bool **is_finite** ( float val )

is_finite retourne TRUE si val est une valeur finie, c'est à dire une valeur qui peut être représentée par un nombre à virgule flottante sur cette plate−forme.

Voir aussi is_infinite et is_nan .

## 6.18.32 is_infinite() :

bool **is_infinite** ( float val )

**is_infinite** TRUE si val est infini (positif ou négatif), comme par exemple le résultat de log(0) ou une valeur qui est trop grande pour être représenté par un nombre à virgule flottante sur cette plate−forme.

Voir aussi is_finite et is_nan .

## 6.18.33 is_nan() : Indique si une valeur est un nombre

bool **is_nan** ( float val )

is_nan retourne TRUE si val n'est pas un nombre, comme par exemple, le résultat de acos(1.01) .

Voir aussi is_finite et is_infinite .

## 6.18.34 lcg_value() : Générateur de congruence combinée linéaire

float **lcg_value** ( void )

lcg_value retourne un nombre pseudo−aléatoire, compris entre 0 et 1. lcg_value combine deux générateurs de congruence, de périodes respectives 2^31 − 85 et 2^31 − 249 . La période de cette

fonction est le produit de ces deux nombres premiers (soit (2^31 − 85)*(2^31 − 249) ).

## 6.18.35 log() : Logarithme naturel (népérien)

float **log** ( float arg , *float base* )

log retourne le logarithme naturel (ou népérien) de arg . Si le paramètre optionnel base est spécifié, log retourn alors le logarithme en base base .

| Note |
| --- |
| Le paramètre base a été ajouté en PHP version 4.3.0. |

Vous pouvez toujours calculer le logarithme en base b d'un nombre n , en utilisant la formule ; logb(n) = log(n)/log(b) , où log est le logarithme naturel ou népérien.

Voir aussi exp .

## 6.18.36 log10() : logarithme en base 10

float **log10** ( float arg )

log10 retourne le logarithme en base 10 de arg .

## 6.18.37 log1p() : Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro.

float **log1p** ( float number )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.18.38 max() : La plus grande valeur

mixed **max** ( number arg1 , number arg2 , *number ...* ) mixed **max** ( array numbers , *array ...* )

max retourne la plus grande valeur numérique parmi les valeurs passées en paramètre.

Si le premier paramètre est un tableau, max retourne la plus grande valeur de ce tableau. Si le premier paramètre est un entier, une chaîne ou un nombre à virgule flottante ( float ), max requiert au moins deux paramètres, et retournera alors le plus grand d'entre eux. Le nombre d'arguments est alors illimité.

Si au moins une valeur est un nombre à virgule flottante, elles seront toutes traitées comme des

nombres à virgule flottante, et un nombre à virgule flottante sera retourné. Si aucune valeur n'est un nombre à virgule flottante, elles seront traitées comme des entiers, et un entier sera retourné.

Exemple d'utilisation de max

```
<?php
echo max(1, 3, 5, 6, 7);  // 7
echo max(array(2, 4, 5)); // 5

echo max(0, 'bonjour');     // 0
echo max('bonjour', 0);     // bonjour
echo max(-1, 'bonjour');    // bonjour

// Avec plusieurs tableaux, min() fait les comparaisons de gauche à droite
// dans notre exemple : 2 == 2, mais 4 < 5
$val = max(array(2, 4, 8), array(2, 5, 7)); // array(2, 5, 7)

// Si un mélange de tableau et de scalaires sont fournis,
 // le tableau est toujours retourné, car il est considéré comme le plus grand
$val = max('string', array(2, 5, 7), 42);   // array(2, 5, 7)
?>
```

Voir aussi min et count .

## 6.18.39 min() : La plus petite valeur

mixed **min** ( number arg1 , number arg2 , *number* ... ) mixed **min** ( array numbers , *array* ... )

min retourne la plus petite valeur numérique parmi les valeurs passées en paramètres.

Dans la première syntaxe, vous devez fournir au moins deux paramètres à la fonction min , qui vous retournera la plus petite de deux valeurs. Vous pouvez comparer un nombre illimité de valeurs. Si l'une d'entre elle est indéfinie, min échouera.

Dans la seconde variante, min retourne la plus petite valeur trouvée dans le tableau numbers .

Si au moins une valeur est un nombre à virgule flottante, elles seront toutes traitées comme des nombres à virgule flottante, et un nombre à virgule flottante sera retourné. Si aucune valeur n'est un nombre à virgule flottante, elles seront traitées comme des entiers, et un entier sera retourné.

Exemples avec min

```
<?php
echo min(2, 3, 1, 6, 7);  // 1
echo min(array(2, 4, 5)); // 2

echo min(0, 'bonjour');     // 0
echo min('bonjour', 0);     // bonjour
echo min('bonjour', -1);    // -1

// Vous pouvez implémenter votre propre gestion d'erreur
// en cas d'échec (une variable peut ne pas exister)
if (!$min_value = @min($a, $b, $c)) {
    echo "Impossible de trouver la valeur minimale, essayez à nouveau!.";
} else {
    echo "La valeur minimale est $min_value";
}

// Avec plusieurs tableaux, min() fait les comparaisons de gauche à droite
// dans notre exemple : 2 == 2, mais 4 < 5
$val = min(array(2, 4, 8), array(2, 5, 1)); // array(2, 4, 8)

// Si un mélange de tableau et de scalaires sont fournis,
 // le tableau n'est jamais retourné, car il est considéré comme le plus grand
$val = min('chaine', array(2, 5, 7), 42);   // chaine
?>
```

Voir aussi max et count .

## 6.18.40 mt_getrandmax() : La plus grand valeur aléatoire possible

int **mt_getrandmax** ( void )

mt_getrandmax retourne la plus grand valeur aléatoire possible que peut retourner mt_rand .

Voir aussi mt_rand , mt_srand rand , srand et getrandmax .

## 6.18.41 mt_rand() : Génère une valeur aléatoire (meilleure méthode)

int **mt_rand** ( *int* *min* , int max )

De nombreux générateurs de nombres aléatoires provenant de vieilles bibliothèques libcs ont des comportement douteux et sont très lents. Par défaut, PHP utilise le générateur de nombres aléatoires de libc avec la fonction rand . mt_rand est une fonction de remplacement, pour cette dernière. Elle utilise un générateur de nombres aléatoire de caractéristique connue, le " Mersenne Twister ", qui va produire des nombres utilisables en cryptographie, et qui est 4 fois plus rapide que la fonction standard libc. La "Homepage of the Mersenne Twister " est http://www.math.keio.ac.jp/~matumoto/emt.html .

Appelée sans les arguments optionnels min et max , mt_rand retourne un nombre pseudo−aléatoire, entre 0 et RAND_MAX . Pour obtenir un nombre entre 5 et 15 inclus, il faut utiliser mt_rand(5,15) .

| Note |
| --- |
| Depuis PHP 4.2.0, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec srand ou mt_srand car c'est fait automatiquement. |

| Note |
| --- |
| Dans les versions antérieures à la 3.0.7, la signification du paramètre max était "longueur". Pour avoir le même résultat, il faut utiliser mt_rand (5, 11) pour obtenir un nombre aléatoire entre 5 et 15. |

Voir aussi mt_srand , mt_getrandmax et rand .

## 6.18.42 mt_srand() : Initialise une valeur aléatoire (meilleure méthode)

void **mt_srand** ( *int* *seed* )

mt_srand initialise une meilleure valeur aléatoire avec seed . Depuis PHP 4.2.0, le paramètre seed est devenu optionnel et est généré par PHP si il n'est pas renseigné.

Exemple avec mt_srand

```
 // initialisation avec des microsecondes
function make_seed() {
    list($usec, $sec) = explode(' ', microtime());
    return (float) $sec + ((float) $usec * 100000);
}
mt_srand(make_seed());
$randval = mt_rand();
```

| Note |
| --- |
| Depuis PHP 4.2.0, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec srand ou mt_srand car c'est fait automatiquement. |

Voir aussi mt_rand , mt_getrandmax et srand .

## 6.18.43 octdec() : Convertit d'octal en décimal

int **octdec** ( string octal_string )

octdec retourne une chaîne contenant la représentation décimale du nombre octal_tring . Le nombre le plus grand qui puisse être converti est 17777777777 en décimal, ce qui donnera "2147483647".

Voir aussi decoct , bindec , hexdec et base_convert .

## 6.18.44 pi() : Retourne la valeur de pi

float **pi** ( void )

pi retourne la valeur de pi. La valeur retournée est de type float , et est à la précision indiquée par la directive precision , indiquée dans le fichier php.ini , qui vaut par défaut 14 . De plus, vous pouvez utiliser la constante M_PI , qui retourne un résultat identique à la fonction pi .

Exemple avec pi

```
<?php
 echo pi(); // 3.1415926535898
 echo M_PI; // 3.1415926535898
?>
```

## 6.18.45 pow() : Expression exponentielle

number **pow** ( number base , number exp )

pow retourne base élevé à la puissance exp . Si possible, pow retourne un integer .

Si le calcul ne peut être fait, une alerte sera affichée et pow retournera FALSE .

Quelques exemples avec pow

```
<?php

var_dump( pow(2,8) ); // int(256)
echo pow(-1,20); // 1
echo pow(0, 0); // 1

echo pow(-1, 5.5); // error

?>
```

Attention

En PHP 4.0.6 plus ancien, pow retournait toujours un nombre à virgule flottante ( float ), et n'affichait pas d'alerte. Si le calcul est impossible (racine d'un nombre négatif, par exemple), pow retournait NAN .

Voir aussi exp et sqrt .

## 6.18.46 rad2deg() : Convertit de radians en degrés

float **rad2deg** ( float number )

rad2deg convertit number (supposé en radians) en degrés.

Voir aussi deg2rad .

## 6.18.47 rand() : Génère une valeur aléatoire

int **rand** ( *int* *min* , int max )

Appelée sans les options min et max , rand retourne un nombre pseudo−aléatoire entre 0 et RAND_MAX . Si vous voulez un nombre aléatoire entre 5 et 15 (inclus), par exemple, utilisez rand (5, 15) .

| Note |
| --- |
| Depuis PHP 4.2.0, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec srand ou mt_srand car c'est fait automatiquement. |

| Note |
| --- |
| Dans les versions antérieures à la 3.0.7 la signification du paramètre max était longueur. Pour avoir le même résultat, il faut utiliser mt_rand (5, 11) pour obtenir un nombre aléatoire entre 5 et 15. |

Voir aussi srand , mt_rand et getrandmax .

## 6.18.48 round() : Arrondi

float **round** ( float val , *int* *precision* )

round retourne la valeur arrondie de val à la précision precision (nombre de chiffres après la virgule). Le paramètre precision peut être négatif ou null (sa valeur par défaut).

| Attention |
| --- |
| PHP ne gère par correctement les chaînes telles que "12 300,2" , par défaut. Reportez−vous à la conversion de chaînes . |

Exemple avec round

```
<?php
echo round(3.4);         // 3
echo round(3.5);         // 4
echo round(3.6);         // 4
echo round(3.6, 0);      // 4
echo round(1.95583, 2);  // 1.96
echo round(1241757, −3); // 1242000
?>
```

| Note |
| --- |
| Le paramètre precision est disponible uniquement en PHP 4. |

Voir aussi ceil et floor .

## 6.18.49 sin() : Sinus

float **sin** ( float arg )

sin retourne le sinus de arg ( arg in radians).

Exemple avec sin

```
<?php

// La précision dépend de la directive precision
print sin(deg2rad(60));  //  0.866025403 ...
print sin(60);           // −0.304810621 ...

?>
```

Voir aussi asin , cos , tan et deg2rad .

## 6.18.50 sinh() : Sinyus hyperbolique

float **sinh** ( float arg )

sinh retourne le sinus hyperbolique de arg , défini comme (exp(arg) − exp(−arg))/2 .

Voir aussi sin , asinh , cos et tan .

## 6.18.51 sqrt() : Racine carrée

float **sqrt** ( float arg )

sqrt retourne la racine carrée de arg .

Exemple avec sqrt

```
<?php
 // La précision dépend de votre directive precision
 echo sqrt(9); // 3
 echo sqrt(10); // 3.16227766 ...
?>
```

Voir aussi pow .

## 6.18.52 srand() : Initialise le générateur de nombres aléatoires

void **srand** ( *int seed* )

srand initialise le générateur de nombres aléatoires avec seed . Depuis PHP 4.2.0, le paramètre seed est devenu optionnel et est généré par PHP si il n'est pas renseigné.

Exemple avec srand

```
 // initialise avec les microsecondes
function make_seed() {
    list($usec, $sec) = explode(' ', microtime());
    return (float) $sec + ((float) $usec * 100000);
}
srand(make_seed());
$randval = rand();
```

Note

Depuis PHP 4.2.0, il n'est plus besoin d'initialiser le générateur de nombres aléatoires avec srand

ou mt_srand car c'est fait automatiquement.

Voir aussi rand , getrandmax , mt_rand , mt_srand et mt_getrandmax .

### 6.18.53 tan() : Tangente

float **tan** ( float arg )

tan retourne la tangente de arg ( arg en radians).

Voir aussi atan , sin , cos et deg2rad .

### 6.18.54 tanh() : Tangente hyperbolique

float **tanh** ( float arg )

tanh retourne la tangeante hyperbolique de arg , définie comme sinh(arg)/cosh(arg) .

Voir aussi tan , atanh , sin et cos .

# 6.19 Chaînes de caractères multi−octets

## 6.19.1 Introduction

De nombreuses langues dont les signes ne peuvent pas être exprimés sur un seul octet. Des codes multi−octets sont utilisés pour pallier à cette insuffisance. mbstring est développé pour supporter les caractères japonais. Cependant, de nombreuses fonctions mbstring peuvent supporter d'autres jeux de caractères.

Les jeux de caractères multi−octets représentent les caractères sur plusieurs octets consécutifs (d'où leur nom). Certains systèmes d'encodages ont des caractères d'échappement dédiés, pour démarrer/finir une séquence de caractères multi−octets. De ce fait, certains caractères peuvent être détruit lorsqu'une chaîne est coupée en plusieurs morceaux, ou bien conduire à des résultats erronnés lorsque le nombre de caractère est compté. Il faut utiliser des fonctions qui supportent ces encodages. Les fonctions mbstring supportent les jeux de caractères multi−octets, ainsi que les conversions.

Etant donné que PHP supporte essentiellement le jeu de caractères ISO−8859−1, certains jeux de caractères ne fonctionnent pas bien avec PHP. Par conséquent, il est important de donner une valeur à l'option de configuration mbstring.internal_encoding qui permettent à PHP de travailler correctement.

Pré−requis PHP 4

- Encodage par octet
- Les caractères d'un octet dans l'intervalle 00h−7fh doivent être compatibles avec le code ASCII
- Jeux de caractères multi−octets, qui n'utilisent pas l'intervalle 00h−7fh .

Voici des exemples d'encodage internes :

| Jeu de caractères qui fonctionnent avec PHP |
| --- |
| Jeus de caractères qui fonctionnent avec PHP :<br>ISO−8859−*, EUC−JP, UTF−8<br><br>Jeu de caractères qui NE fonctionnent PAS avec PHP :<br>JIS, SJIS |

Les jeux de caractères qui ne fonctionnent pas comme encodage interne à PHP, peuvent toutefois être utilisé avec les fonctions de conversion de mbstring .

| Note |
| --- |
| SJIS ne doit pas être utilisé comme encodage interne, à moins que vous ne soyez familier de l'analyseur/compilateur, et des problèmes liés aux jeux de caractères. |

| Note |
| --- |
| SI vous utilisez une base de données avec PHP, il est recommandé que vous utilisiez le même jeu de caractère pour la base de données et le jeu de caractère interne de PHP, pour améliorer les performances.<br><br>Si vous utilisez PostgreSQL, il supporte des jeux de caractères qui peuvent être différents de ceux du client. Reportez vous au manuel de PostgreSQL pour plus de détails. |

## 6.19.2 Installation

mbstring est un module PHP. Vous devez activer le module avec le script de configuration configure . Reportez vous à la section installation pour plus de détails.

Les options de configurations suivantes sont liées au module mbstring .

- −−enable−mbstring : Active les fonctions mbstring . Cette option est nécessaire pour utiliser les fonctions mbstring .

| Note |
|---|
| A partir de PHP 4.3.0, l'option −−enable−mbstring sera activée par défaut, et remplacée par −−with−mbstring[=LANG] pour supporter le chinois, coréen et russe. Le jeu de caractères japonais est supporté par défaut. Si −−with−mbstring=cn est utilisé, le chinois simplifié sera supporté. Si −−with−mbstring=tw est utilisé, le chinois traditionnel sera supporté. Si −−with−mbstring=kr est utilisé, le coréen sera supporté. Si −−with−mbstring=ru est utilisé, le russe sera supporté. Si −−with−mbstring=all est utilisé, toutes les langues ci−dessus seront configurée, mais la taille de PHP sera maximisée, à cause des tables de conversions pour l'Unicode. Notez que le chinois, coréen et russe sont expérimentelement supportés en PHP 4.3.0. |

- −−enable−mbstr−enc−trans : Active la conversion automatique des données par HTTP, avec le moteur de conversion de mbstring . Si cette option est activée, les données venants du web via HTTP seront converties dans le jeu de caractères mbstring.internal_encoding , automatiquement. automatically.

| Note |
|---|
| Depuis PHP 4.3.0, l'option −−enable−mbstr−enc−trans sera éliminée, et remplacée par mbstring.encoding_translation . La conversion de jeu de caractères d'entrée HTTP sera activée lorsque cette option sera à On (Par défaut, cette option vaut Off ). |

- −−enable−mbregex : Active les fonctions d'expressions régulières, compatibles avec les caractères multi−octets.

## 6.19.3 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| mbstring.language | NULL | PHP_INI_ALL |
| mbstring.detect_order | NULL | PHP_INI_ALL |
| mbstring.http_input | NULL | PHP_INI_ALL |
| mbstring.http_output | NULL | PHP_INI_ALL |
| mbstring.internal_encoding | NULL | PHP_INI_ALL |
| mbstring.script_encoding | NULL | PHP_INI_ALL |
| mbstring.substitute_character | NULL | PHP_INI_ALL |
| mbstring.func_overload | "0" | PHP_INI_SYSTEM |
| mbstring.encoding_translation | "0" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*mbstring.language string*
: Définit le langage utilisé par mbstring. Notez que cette option définit mbstring.internal_encoding mbstring.internal_encoding doit être placé après mbstring.language dans le fichier php.ini

*mbstring.encoding_translation string*
: Active la détection et la traduction des données d'entrées HTTP vers le codage

interne mbstring.

*mbstring.internal_encoding string*
: Définit l'encodage interne par défaut.

*mbstring.http_input string*
: Définit l'encodage de reception HTTP par défaut.

*mbstring.http_output string*
: Définit l'encodage d'affichage HTTP par défaut.

*mbstring.detect_order string*
: Définit l'ordre de détection des encodages par défaut. Voir aussi mb_detect_order .

*mbstring.substitute_character string*
: Définit l'encodage de substitution par défaut : il est utilisé pour les caractères invalides.

*mbstring.func_overload string*
: Remplace les fonctions de traitement des chaînes par les fonctions mbstring. mail , ereg , etc... sont remplacées par mb_send_mail , mb_ereg , etc... Les valeurs possibles sont 0, 1, 2, 4 ou toute combinaison des quatres. Par exemple, 7 remplacera tout : 0: Aucun remplacement, 1: Remplace mail , 2: Remplace les fonctions str*(), 4: Remplace les fonctions ereg*().

Les navigateurs sont supposés utiliser le même jeu de caractères lorsqu'ils soumettent un formulaire. Mais, tous les navigateurs ne le font pas. Reportez vous à la fonction mb_http_input pour détecter les jeux de caractères utilisé par les navigateurs.

Si enctype est mis à multipart/form−data dans les formulaires HTML, mbstring ne convertira pas les données de POST. L'utilisateur doit les convertir dans le script, si nécessaire.

En général, les navigateurs sont suffisament intelligent pour détecter les jeux de caractères dans le HTML. Si charset est activé dans les en−têtes, cela donnera de meilleurs résultats. Changez la valeur de default_charset avec la valeur appropriée.

Configuration de php.ini pour mbstring

```
 ; Langage par défaut
mbstring.language        = English; Anglais (par défaut)
mbstring.language        = Japanese; Japonais

;; Jeu de caractère interne
;; Note : Assurez vous que ce jeu fonctionne avec PHP
mbstring.internal_encoding    = UTF−8

 ;; Activation de la conversion automatique des entrées HTTP
mbstring.encoding_translation = On

;; Jeu de caractères par défaut pour les données d'entrée HTTP
;; Note : Le script ne peux pas changer cette configuration
mbstring.http_input           = pass    ; Aucune conversion.
 mbstring.http_input           = auto    ; Utilise auto
                                ; "auto" est remplacé par "ASCII,JIS,UTF−8,EUC−JP,SJIS"
mbstring.http_input           = SJIS    ; Utilise SJIS
mbstring.http_input           = UTF−8,SJIS,EUC−JP ; Specifie l'ordre

;;Jeu de caractères par défaut pour les données de sortie HTTP
mbstring.http_output          = pass    ; Aucune conversion
mbstring.http_output          = UTF−8   ; Utilise UTF−8

;; Ordre de détection des jeux de caractères
mbstring.detect_order         = auto    ; Utilise la détection automatique
mbstring.detect_order         = ASCII,JIS,UTF−8,SJIS,EUC−JP ; Spécifie l'ordre

;; Set default substitute character
mbstring.substitute_character = 12307   ; Spécifie une valeur Unicode
mbstring.substitute_character = none    ; Ne pas afficher de caractères
mbstring.substitute_character = long    ; Exemple complet : U+3000,JIS+7E7E
```

Configuration de php.ini pour les utilisateurs de EUC−JP

```
 ;; Désactive la bufferisation de sortie
output_buffering      = Off

;; Choisi le jeu de caractères
default_charset       = EUC−JP
```

```
;; Le langage par défaut est le japonais
mbstring.language = Japanese

;; Activation de la traduction automatique des données d'entrée HTTP
mbstring.encoding_translation = On

;; Acitvation de la converstion automatique
mbstring.http_input   = auto

 ;; Convertit les sorties en EUC−JP
mbstring.http_output  = EUC−JP

;; Utilise le jeu de caractères interne EUC−JP
mbstring.internal_encoding = EUC−JP

;; Ne pas afficher les caractères invalides
mbstring.substitute_character = none
```

```
Configuration de php.ini pour les utilisateurs de SJIS
 ;; Active la bufferisation de sortie
output_buffering     = On

;; Utilise le gestionnaire mb_output_handler pour la conversion de sortie
output_handler       = mb_output_handler

;; Choisi le jeu de caractères
default_charset      = Shift_JIS

;; Le langage par défaut est le japonais
mbstring.language = Japanese

;; Activation de la traduction automatique des données d'entrée HTTP
mbstring.http_input  = auto

 ;; Convertit en SJIS
mbstring.http_output = SJIS

;;Utilise le jeu de caractères interne EUC−JP
mbstring.internal_encoding = EUC−JP

;; Ne pas afficher les caractères invalides
mbstring.substitute_character = none
```

## 6.19.4 Types de ressources

Cette extension ne définit aucune ressource.

## 6.19.5 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*MB_OVERLOAD_MAIL (entier)*
*MB_OVERLOAD_STRING (entier)*
*MB_OVERLOAD_REGEX (entier)*

## 6.19.6 Entrées/Sorties HTTP

La conversion automatiques des entrées/sorties HTTP peuvent aussi convertir des données binaires. Les utilisateurs doivent contrôler les conversions, si des données binaires doivent être utilisées via HTTP.

Si l'option enctype d'un formulaire HTML vaut multipart/form−data , mbstring ne convertira pas les données du POST. Dans ce cas, les chaînes de caractères doivent être convertis manuellement.

- Entrée HTTP

Il n'y a pas de moyen de contrôler la conversion des caractères HTTP en entrée, depuis un script PHP. Pour désactiver cette conversion, il faut le faire dès le fichier php.ini .

Désactive la conversion HTTP dans le php.ini

```
 ;; Désactive la conversion HTTP
mbstring.http_input = pass
;; Désactive la conversion HTTP (PHP 4.3.0 ou plus récent)
mbstring.encoding_translation = Off
```

Lorsque vous utilisez PHP comme module Apache, il est possible d'annuler la configuration du php.ini pour chaque Virtual Host dans le fichier httpd.conf ou par dossier avec le fichier .htaccess . Reportez vous à la section de configuration ainsi qu'au manuel Apache.

- Sorties HTTP

Il y a plusieurs moyens d'activer la conversion en sortie de script PHP. L'un d'entre eux utilise php.ini , un autre utilise ob_start avec la fonction mb_output_handler comme fonction de call−back.

Note

Pour les utilisateurs PHP3−i18n, le système de conversion de mbstring diffère de celui de PHP3−i18n. Le jeu de caractère est converti avec un buffer de sortie.

Exemple de configuration de mbstring dans php.ini

```
 ;; Active la conversion de sortie pour toute les pages PHP

;; Active la bufferisation de sortie
output_buffering    = On

;; Choisi mb_output_handler pour effectuer la conversion de sortie
output_handler      = mb_output_handler
```

Exemple de script avec mbstring

```
 <?php

// Active la conversion de caractère uniquement pour cette page

// Choisi le jeu de caractères SJIS
mb_http_output('SJIS');

// Commence la bufferisation et spécifie "mb_output_handler"
// comme fonction de callback
ob_start('mb_output_handler');
?>
```

## 6.19.7 Jeux de caractères supportés

Actuellement, les jeux de caractères suivants sont supportés par mbstring . L'encodage de caractère peut être spécifié par les paramètres encoding dans les fonctions mbstring .

Les jeux de caractères suivants sont supportés par mbstring :

UCS−4 , UCS−4BE , UCS−4LE , UCS−2 , UCS−2BE , UCS−2LE , UTF−32 , UTF−32BE , UTF−32LE , UCS−2LE , UTF−16 , UTF−16BE , UTF−16LE , UTF−8 , UTF−7 , ASCII , EUC−JP , SJIS , eucJP−win , SJIS−win , ISO−2022−JP , JIS , ISO−8859−1 , ISO−8859−2 , ISO−8859−3 , ISO−8859−4 , ISO−8859−5 , ISO−8859−6 , ISO−8859−7 , ISO−8859−8 , ISO−8859−9 , ISO−8859−10 , ISO−8859−13 , ISO−8859−14 , ISO−8859−15 , byte2be , byte2le , byte4be , byte4le , BASE64 , 7bit , 8bit et UTF7−IMAP .

Depuis PHP 4.3.0, les jeux de caractèrs suivants sont ajoutés, mais restent expérimentaux : EUC−CN , CP936 , HZ , EUC−TW , CP950 , BIG−5 , EUC−KR , UHC ( CP949 ), ISO−2022−KR , Windows−1251 ( CP1251 ), Windows−1252 ( CP1252 ), CP866 et KOI8−R .

Les entrées du fichiers php.ini , qui acceptent des noms de jeux de caractères, acceptent aussi les valeurs " auto " et " pass ". Les fonctions mbstring ,qui acceptent des noms de jeux de caractères, acceptent aussi la valeur " auto "/

Si " pass " est utilisée, aucune conversion n'est effectuée.

Si " auto " est utilisée, elle est remplacée par " ASCII,JIS,UTF−8,EUC−JP,SJIS ".

Voir aussi mb_detect_order .

| Note |
| --- |
| Un jeu de caractère supporté n'est pas forcément un bon choix comme jeu de caractères interne. |

## 6.19.8 Exploitation des chaînes multi−octets en PHP

Comme presque tout PHP est écrit pour des langues qui sont simple octets, il y a des difficultés à gérer le japonais. Toutes les fonctions PHP de gestion des chaînes de caractères, comme par exemple substr ne supportent par les chaînes multi octets.

L'extension Multibyte (multi−octets, alias mbstring) dispose de fonctions de chaînes, qui supportent le multi−octet, comme par exemple mb_substr , qui est un remplaçant de substr .

mbstring supporte aussi le remplacement de fonction, pour permettre le support des chaînes multi−octets sans modifier les scripts PHP. En utilisant ce système de remplacement de fonctions, certains fonctions PHP seront remplacées par leur équivalent de mbstring. Par exemple mb_substr remplacera substr . Ce système de remplacement transparent, permet un portage simple et efficace des applications.

mbstring.func_overload , dans le php.ini , doit être configuré avec une valeur positive : 1 remplace la fonction d'envoi de mail; 2 active les fonction de chaînes; 4 active les fonctions d'expression régulières. Par exemple, avec la valeur 7, toutes les fonctions possibles sont remplacées. Voici la liste complète des fonctions remplacées, avec leur fonction de remplacement.

| Valeur de mbstring.func_overload | Fonction originale | Fonction de remplacement |
| --- | --- | --- |
| 1 | mail | mb_send_mail |
| 2 | strlen | mb_strlen |
| 2 | strpos | mb_strpos |
| 2 | strrpos | mb_strrpos |
| 2 | substr | mb_substr |
| 4 | ereg | mb_ereg |
| 4 | eregi | mb_eregi |
| 4 | ereg_replace | mb_ereg_replace |
| 4 | eregi_replace | mb_eregi_replace |
| 4 | split | mb_split |

## 6.19.9 Cas des caractères japonais

La plupart des caractères japonais demandent plus d'un octet pour être représentés. De plus, plusieurs jeux de caractères japonais existent : il y a notamment EUC−JP, Shift_JIS et ISO−2022−JP. Unicode devient de plus en plus populaire, et UTF−8 aussi. Pour développer des applications Web en environnement japonais, il faut savoir que les encodages ci−dessus dépendent de l'application qu'on en fait : entrée/sortie HTTP, bases de données ou courrier électronique.

- La taille nécessaire à un caractère peut aller jusqu'à 4 octets.

- Un caractère multi−octets occupe généralement deux octets, à comparer avec les caractères simple−octet traditionnellement utilisé. Les caractères les plus gros sont appelés "zen−kaku" (i.e. grande largeur) et les plus petits sont appelés "han−kaku" (i.e. demi−largeur). Les caractères "zen−kaku" sont généralement de taille constante.
- Certains encodage de caractères définissent des séquences de début/fin pour les sections multi−octets.
- Les bases de données allouent des tailles de stockages différentes de celles utilisées par PHP, même si le même encodage de caractère est utilisé (par exemple, PostGreSQL).
- Le courrier électronique utilise généralement ISO−2022−JP.
- Les sites web en "i−mode" utilisent Shift_JIS.

## 6.19.10 Références

Les jeux de caractères multi−octets et leurs techniques sont très complexes. Il n'est pas possible de couvrir tous les aspects en détails ici. Reportez−vous aux URL suivantes, pour d'autres ressouces complémentaires :

- Unicode/UTF/UCS/etc

  http://www.unicode.org/
- Japonais/coréen/Chinois

  ftp://ftp.ora.com/pub/examples/nutshell/ujip/doc/cjk.inf

- Introduction
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Entrées/Sorties HTTP
- Jeux de caractères supportés
- Exploitation des chaînes multi−octets en PHP
- Cas des caractères japonais
- Références
- mb_convert_case
- mb_convert_encoding
- mb_convert_kana
- mb_convert_variables
- mb_decode_mimeheader
- mb_decode_numericentity
- mb_detect_encoding
- mb_detect_order
- mb_encode_mimeheader
- mb_encode_numericentity
- mb_ereg_match
- mb_ereg_replace
- mb_ereg_search_getpos
- mb_ereg_search_getregs
- mb_ereg_search_init
- mb_ereg_search_pos
- mb_ereg_search_regs
- mb_ereg_search_setpos
- mb_ereg_search

- mb_ereg
- mb_eregi_replace
- mb_eregi
- mb_get_info
- mb_http_input
- mb_http_output
- mb_internal_encoding
- mb_language
- mb_output_handler
- mb_parse_str
- mb_preferred_mime_name
- mb_regex_encoding
- mb_regex_set_options
- mb_send_mail
- mb_split
- mb_strcut
- mb_strimwidth
- mb_strlen
- mb_strpos
- mb_strrpos
- mb_strtolower
- mb_strtoupper
- mb_strwidth
- mb_substitute_character
- mb_substr_count
- mb_substr

## 6.19.12 mb_convert_encoding() : Conversion d'encodage

string **mb_convert_encoding** ( string str , string to−encoding , *mixed from−encoding* )

mb_convert_encoding convertit la chaîne str depuis l'encodage from−encoding vers l'encodage to−encoding .

str à convertir.

from−encoding est l'encodage de la chaîne str à l'origine. Il sera détecté parmi plusieurs encodage fournis sous forme d'un tableau, ou d'une liste d'encodages séparés par des virgules.

Exemple avec mb_convert_encoding

```
<?php
/* Convertit l'encodage interne vers SJIS */
$str = mb_convert_encoding($str, "SJIS");
/* Convertit EUC−JP en UTF−7 */
$str = mb_convert_encoding($str, "UTF−7", "EUC−JP");
/* Détecte automatiquement un encodage entre JIS, eucjp−win ou sjis−win,
   Puis convertit en UCS−2LE */
$str = mb_convert_encoding($str, "UCS−2LE", "JIS, eucjp−win, sjis−win");
/* "auto" signifie "ASCII,JIS,UTF−8,EUC−JP,SJIS" */
$str = mb_convert_encoding($str, "EUC−JP", "auto");
?>
```

Voir aussi mb_detect_order .

## 6.19.13 mb_convert_kana() : Convertit entre les différents "kana"

string **mb_convert_kana** ( string str , string option , *mixed encoding* )

mb_convert_kana effectue une conversion "han−kaku" − "zen−kaku" sur la chaîne str . Elle retourne la chaîne convertie. Cette fonction n'est utile que pour le japonais.

option est l'option de conversion. La valeur par défaut est "KV" .

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Spécifiez les conversions en combinant les valeurs suivantes. La valeur par défaut est KV .

| Option | Valeur |
|---|---|
| r | Convertit l'alphabet "zen−kaku" en "han−kaku". |
| R | Convertit l'alphabet "han−kaku" en "zen−kaku". |
| n | Convertit les nombres "zen−kaku" en "han−kaku". |
| N | Convertit les nombres "han−kaku" en "zen−kaku". |
| a | Convertit les nombres et alphabets "zen−kaku" en "han−kaku". |
| A | Convertit les nombres et alphabets "zen−kaku" en "han−kaku". (Les caractères inclus dans les options "a", "A" sont U+0021 − U+007E en excluant U+0022, U+0027, U+005C, U+007E) |
| s | Convertit "zen−kaku" en "han−kaku" (U+3000 −> U+0020) |
| S | Convertit "han−kaku" en "zen−kaku" (U+0020 −> U+3000) |
| k | Convertit "zen−kaku kata−kana" en "han−kaku kata−kana" |
| K | Convertit "han−kaku kata−kana" en "zen−kaku kata−kana". |
| h | Convertit "zen−kaku hira−gana" en "han−kaku kata−kana". |
| H | Convertit "han−kaku kata−kana" en "zen−kaku hira−gana". |
| c | Convertit "zen−kaku kata−kana" en "zen−kaku hira−gana". |
| C | Convertit "zen−kaku hira−gana" en "zen−kaku kata−kana". |
| V | Supprime les notations vocales, et les convertit en caractères. A utiliser avec "K","H". |

Exemple avec mb_convert_kana

```
<?php
/* Convertit tous les "kana" en "zen−kaku" "kata−kana" */
$str = mb_convert_kana($str, "KVC");
/* Convertit "han−kaku" "kata−kana" en "zen−kaku" "kata−kana"
   et "zen−kaku" alpha−numeric en "han−kaku" */
$str = mb_convert_kana($str, "KVa");
?>
```

## 6.19.14 mb_convert_variables() : Convertit l'encodage de variables

string **mb_convert_variables** ( string to−encoding , mixed from−encoding , mixed vars )

mb_convert_variables convertit l'encodage des variables vars depuis l'encodage from−encoding vers l'encodage to−encoding , puis retourne le nom de l'encodage détecté, en cas de succès, ou FALSE en cas d'échec.

from−encoding est une liste d'encodages possibles pour les variables vars , fourni sous forme d'un tableau ou d'une liste d'encodage, séparés par des virgules. Si from−coding est omis, les encodages fournit dans mb_detect_order sont utilisés.

vars est une référence sur une variables à convertir. Les chaînes, tableaux et objets sont aussi supportés.

Exemple avec mb_convert_variables

```
<?php
/* Convertit les variables $post1, $post2 en encodage interne */
$interenc = mb_internal_encoding();
$inputenc = mb_convert_variables($interenc, "ASCII,UTF−8,SJIS−win", $post1, $post2);
?>
```

## 6.19.15 mb_decode_mimeheader() : Décode une en−tête MIME

string **mb_decode_mimeheader** ( string str )

mb_decode_mimeheader décode l'en−tête MIME str , obtenue dans un courrier électronique.

mb_decode_mimeheader retourne la chaîne décodée, encodée au format interne.

Voir aussi mb_encode_mimeheader .

## 6.19.16 mb_decode_numericentity() : Décode les entités HTML en caractères

string **mb_decode_numericentity** ( string str , array convmap , *string encoding* )

mb_decode_numericentity la chaîne d'entités HTML str en chaîne, et retourne cette chaîne.

convmap est un tableau qui spécifie les codes à convertir.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Exemple avec le paramètre convmap

```
 $convmap = array (
   int start_code1, int end_code1, int offset1, int mask1,
   int start_code2, int end_code2, int offset2, int mask2,
   ........
   int start_codeN, int end_codeN, int offsetN, int maskN );
// Spécifie les valeurs Unicode de début (start_codeN) et fin (end_codeN)
// Ajoutez offsetN à la valeur, et faites un ET bit−à−bit avec maskN, puis
// il convertit la valeur obtenu en entité numérique
```

Voir aussi mb_encode_numericentity .

## 6.19.17 mb_detect_encoding() : Détecte un encodage

string **mb_detect_encoding** ( string str , *mixed encoding−list* )

mb_detect_encoding détecte l'encodage utilisé par la chaîne str . mb_detect_encoding retourne le nom de l'encodage détecté.

encoding−list est une liste d'encodage, sous forme de tableau, ou bien de chaîne, les valeurs étant séparés par des virgules.

Si encoding_list est omis, l'ordre spécifié par mb_detect_order est utilisé.

Exemple avec mb_detect_encoding

```
 <?php
/* Détecte l'encodage avec les valeurs par défaut */
echo mb_detect_encoding($str);
/* "auto" signifie "ASCII,JIS,UTF−8,EUC−JP,SJIS" */
echo mb_detect_encoding($str, "auto");
/* Spécifie une liste d'encodages possibles avec une liste à virgules */
echo mb_detect_encoding($str, "JIS, eucjp−win, sjis−win");
/* Spécifie une liste d'encodages possibles avec un tableau  */
$ary[] = "ASCII";
$ary[] = "JIS";
$ary[] = "EUC−JP";
echo mb_detect_encoding($str, $ary);
?>
```

Voir aussi mb_detect_order .

## 6.19.18 mb_detect_order() : Lit/modifie l'ordre de détection des encodages

array **mb_detect_order** ( *mixed encoding−list* )

mb_detect_order remplace l'ordre de détection des encodages courant par encoding−list . mb_detect_order retourne TRUE en cas de succès, et FALSE en cas d'erreur failure.

encoding−list est un tableau, ou une liste d'encodages séparés par une virgule. La valeur "auto" est automatiquement remplacé par "ASCII, JIS, UTF−8, EUC−JP, SJIS".

Si encoding−list est omis, mb_detect_order retourne l'ordre de détection courant des encodages.

Ce paramétrage affected les fonctions mb_detect_encoding et mb_send_mail .

Note

Actuellement, mbstring supporte les filtres de détections ci−dessous. Si une séquence d'octets est invalide pour l'un des filtres suivants, la détection échouera.

UTF−8 , UTF−7 , ASCII , EUC−JP , SJIS , eucJP−win , SJIS−win , JIS et ISO−2022−JP

Pour ISO−8859−* , mbstring détecte toujours ISO−8859−* .

Pour UTF−16 , UTF−32 , UCS2 et UCS4 la détection échouera toujours.

Exemple d'ordre de détection inutile

```
 ; Always detect as ISO-8859-1
detect_order = ISO-8859-1, UTF-8

; Always detect as UTF-8, since ASCII/UTF-7 values are
 ; valid for UTF-8
detect_order = UTF-8, ASCII, UTF-7
```

Exemple avec mb_detect_order

```
 <?php
/* Remplace l'ordre de détection par une liste énumérée */
mb_detect_order("eucjp-win,sjis-win,UTF-8");
/* Remplace l'ordre de détection par un tableau */
$ary[] = "ASCII";
$ary[] = "JIS";
$ary[] = "EUC-JP";
mb_detect_order($ary);
/* Affiche l'ordre de détection courant */
echo implode(", ", mb_detect_order());
?>
```

Voir aussi mb_internal_encoding , mb_http_input , mb_http_output et mb_send_mail

## 6.19.19 mb_encode_mimeheader() : Encode une chaîne pour une en−tête MIME

string **mb_encode_mimeheader** ( string str , *string charset* , *string transfer−encoding* , *string linefeed* )

mb_encode_mimeheader convertit la chaîne str en en−tête MIME, et retourne la chaîne encodée.

charset est le nom de l'encodage. Par défaut, c'est ISO−2022−JP .

transfer−encoding est l'encodage de transfert. Il peut être "B" (Base64) ou "Q" (Quoted−Printable). Par défaut, c'est "B" .

linefeed est le marqueur de fin de ligne. Par défaut, c'est "\r\n" (CRLF).

Exemple avec mb_convert_kana

```
<?php
$name = ""; // kanji
$mbox = "kru";
$doma = "gtinn.mon";
$addr = mb_encode_mimeheader($name, "UTF−7", "Q") . "<" . $mbox . "@" . $doma . ">";
echo $addr;
?>
```

Voir aussi mb_decode_mimeheader .

## 6.19.20 mb_encode_numericentity() : Encode des entités HTML

string **mb_encode_numericentity** ( string str , array convmap , *string encoding* )

mb_encode_numericentity convertit la chaîne str depuis encodage interne en les codes numériques HTML, puis retourne cette chaîne.

convmap est un tableau qui spécifie les codes à convertir.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Exemple de paramètre convmap

```
<?php
$convmap = array (
 int start_code1, int end_code1, int offset1, int mask1,
 int start_code2, int end_code2, int offset2, int mask2,
 ........
 int start_codeN, int end_codeN, int offsetN, int maskN );
// Spécifie les valeurs Unicode de début (start_codeN) et fin (end_codeN)
// Ajoutez offsetN à la valeur, et faites un ET bit−à−bit avec maskN, puis
// il convertit la valeur obtenu en entite numérique
?>
```

Exemple avec mb_encode_numericentity

```
<?php
/* Convertit du ISO−8859−1 en entités HTML */
$convmap = array(0x80, 0xff, 0, 0xff);
$str = mb_encode_numericentity($str, $convmap, "ISO−8859−1");
/* Convertit du code SJIS−win (uniquement le bloc 95−104) en entités numérique */
$convmap = array(
       0xe000, 0xe03e, 0x1040, 0xffff,
       0xe03f, 0xe0bb, 0x1041, 0xffff,
       0xe0bc, 0xe0fa, 0x1084, 0xffff,
       0xe0fb, 0xe177, 0x1085, 0xffff,
       0xe178, 0xe1b6, 0x10c8, 0xffff,
       0xe1b7, 0xe233, 0x10c9, 0xffff,
       0xe234, 0xe272, 0x110c, 0xffff,
       0xe273, 0xe2ef, 0x110d, 0xffff,
       0xe2f0, 0xe32e, 0x1150, 0xffff,
       0xe32f, 0xe3ab, 0x1151, 0xffff );
$str = mb_encode_numericentity($str, $convmap, "sjis−win");
?>
```

Voir aussi mb_decode_numericentity .

## 6.19.21 mb_ereg_match() : Expression régulière POSIX pour les chaînes multi−octets

bool **mb_ereg_match** ( string pattern , string string , *string option* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_match retourne TRUE si string vérifie l'expression régulière pattern , FALSE sinon.

Le jeu de caractère interne ou bien le jeu spécifié par la fonction mb_regex_encoding sera utilisé comme jeu de caractères.

| Note |
| --- |
| Cette fonction est supportée par PHP 4.2.0 et plus récent. |

Voir aussi mb_regex_encoding et mb_ereg .

## 6.19.22 mb_ereg_replace() : Remplace des segments de chaînes, avec le support des expressions régulières mutli−octet

string **mb_ereg_replace** ( string pattern , string replacement , string string , *array option* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_replace analyse la chaîne string pour trouver des segments de chaînes correspondant au masque pattern , puis, elle les remplace avec le texte de remplacement replacement et retourne le résultat ou bien FALSE en cas d'erreur. Des caractères multi−octets peuvent être utilisé dans le masque pattern .

Des options de recherches peuvent être configurés avec le paramètre option . Si i est spécifié, la casse sera ignorée. Si x est spécifié, les espaces blancs seront ignorés. Si m est spécifié, la recherche se fera en mode multi−ligne, et les nouvelles lignes seront inclues dans le joker . . Si p est spécifié, la recherche se fera en mode POSIX, les nouvelles lignes seront considérées comme des caractères normaux. Si e est spécifiée, replacement sera évaluée comme une expression PHP.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_eregi_replace .

## 6.19.23 mb_ereg_search_getpos() : Retourne l'offset du début du prochain segment repéré par une expression régulière

array **mb_ereg_search_getpos** ( void )

| Attention |
| --- |
| |

| |
|---|
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_search_getpos retourne l'offset du prochain segment de code qui correspond au masque utilisé dans mb_ereg_search , mb_ereg_search_pos et mb_ereg_search_regs . L'offset est le nombre de caractères depuis le début de la chaîne.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
|---|
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg_search_setpos .

## 6.19.24 mb_ereg_search_getregs() : Lit le dernier segment de chaîne multi−octets qui correspond au masque

array **mb_ereg_search_getregs** ( void )

| Attention |
|---|
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_search_getregs retourne un tableau incluant toutes les sous chaînes qui ont été trouvées par mb_ereg_search , mb_ereg_search_pos et mb_ereg_search_regs . Si il y a une solution qui a été trouvée, le premier élément sera la sous−chaîne trouvée, le second représentera la première parenthèse capturante, le troisième représentera la deuxième parenthèse capturante, etc... Cette fonction retourne FALSE en cas d'erreur.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
|---|
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg_search_init .

## 6.19.25 mb_ereg_search_init() : Configure les chaînes et les expressions régulières pour le support des caractères multi−octets

array **mb_ereg_search_init** ( string string , *string pattern* , *string option* )

| Attention |
|---|
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_search_init configure string et pattern pour suporter les expressions régulières multi−octets. Ces valeurs sont utilisées par mb_ereg_search , mb_ereg_search_pos et mb_ereg_search_regs . Cette fonction retourne TRUE en cas de succès, et FALSE en cas d'erreur.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg_search_regs .

## 6.19.26 mb_ereg_search_pos() : Retourne la position et la longueur du segement de chaîne qui vérifie le masque de l'expression régulière

array **mb_ereg_search_pos** ( *string pattern* , *string option* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_search_pos retourne un tableau contenant la position et la taille du segement repéré grâce à l'expression régulière. Le premier élément du tableau retourné doit être le début du segment, et le second est la taille du segment, exprimée en octets. Cette fonction retourne FALSE en cas d'erreur.

La chaîne à utiliser est spécifiée par mb_ereg_search_init . Si elle ne l'est pas, la précédente sera utilisée.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg_search_init .

## 6.19.27 mb_ereg_search_regs() : Retourne le segment de chaîne trouvé par une expression régulière multi−octets

array **mb_ereg_search_regs** ( *string pattern* , *string option* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_search_regs exécute l'expression régulière pattern , et si un segment de chaîne correspond, elle le retourne dans un tableau, dont le premier élément est le segment de chaîne trouvé, le deuxième élément est le contenu de la première parenthèse capturante, le troisème élément est le contenu de la troisième paranthèse capturante, etc... La fonction retourne FALSE en cas d'erreur.

Le jeu de caractères internet ou bien le jeu de caractères spécifié par mb_regex_encoding sera utilisé.

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0 et plus récent. |

Voir aussi mb_regex_encoding et mb_ereg_search_init .

## 6.19.28 mb_ereg_search_setpos() : Choisit le point de départ de la recherche par expression régulière

array **mb_ereg_search_setpos** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_search_setpos choisit le point de départ de la recherche que va effectuer la fonction mb_ereg_search .

Le jeu de caractères internet ou bien le jeu de caractères spécifié par mb_regex_encoding sera utilisé.

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0 et plus récent. |

Voir aussi mb_regex_encoding et mb_ereg_search_init .

## 6.19.29 mb_ereg_search() : Recherche par expression régulière multi−octets

bool **mb_ereg_search** ( *string pattern* , *string option* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_search retourne TRUE si la chaîne multi−octets correspond au masque d'expression régulière, ou bien FALSE sinon. La chaîne à étudier a été configurée avec la fonction mb_ereg_search_init . Si le masque pattern n'est pas spécifié, le précédent sera utilisé.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg_search_init .

## 6.19.30 mb_ereg() : Recherche par expression régulière avec support des caractères multi−octets

int **mb_ereg** ( string pattern , string string , *array regs* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg utilise l'expression régulière pattern avec le support des caractères multi−octets sur la chaîne string . mb_ereg retourne 1 si des segments de chaîne qui vérifient le masque ont été trouvé. Le troisième paramètre est optionnel. S'il est fournit, mb_ereg retournera la taille du segment de chaîne identifié, regs contient les sous−chaînes. Si rien n'est trouvé, la fonction retourne FALSE .

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_eregi

## 6.19.31 mb_eregi_replace() : Expression régulière avec support des caractères multi−octets, sans tenir compte de la casse

string **mb_eregi_replace** ( string pattern , string replace , string string )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_ereg_replace analyse la chaîne string avec le masque d'expression régulière pattern , puis remplace le texte trouvé par replacement . Enfin, mb_ereg_replace retourne la nouvelle chaîne ainsi bâtie, ou FALSE en cas d'erreur. Les caractères multi−octets peuvent être utilisés dans la chaîne pattern . La casse sera ignorée.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg_replace .

## 6.19.32 mb_eregi() : Regular expression match ignoring case with multibyte support

int **mb_eregi** ( string pattern , string string , *array regs* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_eregi utilise l'expression régulière pattern avec le support des caractères multi−octets sur la chaîne string . mb_eregi ignore la casse dans ses recherches. mb_ereg retourne 1 si des segments de chaîne qui vérifient le masque ont été trouvé. Le troisième paramètre est optionnel. S'il est fournit, mb_ereg retournera la taille du segment de chaîne identifié, regs contient les sous−chaînes. Si rien n'est trouvé, la fonction retourne FALSE .

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg .

## 6.19.33 mb_get_info() : Lit la configuration interne de l'extension mbstring

string **mb_get_info** ( *string type* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_get_info lit la configuration interne de l'extension mbstring internal setting.

Si type n'est pas spécifié, ou si il vaut "all", un tableau contenant les éléments "internal_encoding", "http_output", "http_input" et "func_overload" sera retourné.

Si type est spécifié, et si il vaut "http_output", "http_input", "internal_encoding" ou "func_overload", la paramètre spécifié sera retourné.

Voir aussi mb_internal_encoding et mb_http_output .

## 6.19.34 mb_http_input() : Détecte le type d'encodage d'un caractère HTTP

string **mb_http_input** ( *string type* )

mb_http_input retourne le type d'encodage utilisé par une requête HTTP.

Le paramètre type spécifie le type d'entrée HTTP. Il peut prendre l'une des valeurs suivantes : "G" pour GET, "P" pour POST, "C" pour COOKIE. Si type est omis, il prend la valeur du dernier type utilisé.

Valeur retournée : nom de l'encodage utilisé. Si mb_http_input ne peut traiter ce type d'encodage, elle retourne FALSE .

Voir aussi mb_internal_encoding , mb_http_output et mb_detect_order

## 6.19.35 mb_http_output() : Lit/modifie l'encodage d'affichage

string **mb_http_output** ( *string encoding* )

Si encoding est fourni, mb_http_output utilisera dorénavant l'encodage encoding pour les affichages HTTP : les caractères qui seront envoyés aux clients web seront convertis dans le jeu de caractères encoding . mb_http_output retourne TRUE en cas de succès, et FALSE en cas d'échec.

Si encoding est omis, mb_http_output retourne l'encodage d'affichage courant.

Voir aussi mb_internal_encoding , mb_http_input et mb_detect_order

## 6.19.36 mb_internal_encoding() : Lit/modifie l'encodage interne

string **mb_internal_encoding** ( *string encoding* )

mb_internal_encoding modifie l'encodage interne courant en le remplacant par encoding . Si ce paramètre est omis, l'encodage interne courant est retourné.

encoding sert lors des conversions des chaînes en provenance et en direction du web, ainsi que lors de la création de chaînes avec le module mbstring.

encoding : Nom d'encodage.

Valeur retournée : si encoding est fourni, mb_internal_encoding retourne TRUE en cas de succès, et FALSE sinon. Si encoding est omis, mb_internal_encoding retourne le nom de l'encodage courant.

Exemple avec mb_internal_encoding

```
<?php
/* Utilise l'encodage interne UTF-8 */
mb_internal_encoding("UTF-8");
/* Affiche l'encodage interne courant */
echo mb_internal_encoding();
?>
```

Voir aussi mb_http_input , mb_http_output et mb_detect_order

## 6.19.37 mb_language() : Lit/modifie le langage courant

string **mb_language** ( *string language* )

mb_language remplace le langage courant par language . Si language est omis, mb_language retourne le langage courant.

Le paramètre language sert à encoder les messages électroniques. Les langages valides sont : "Japanese" (japonais), "ja" (japonais), "English" (anglais), "en" (anglais) and "uni" (UTF−8). mb_send_mail utilise cette option pour encoder les emails.

Le langage et sa configuration valle ISO−2022−JP/Base64 pour le japonais, UTF−8/Base64 pour l'UTF−8 et ISO−8859−1/quoted printable pour l'anglais.

Si language est fourni et est valide, mb_language retourne TRUE . Sinon, elle retourne FALSE . Lorsque le paramètre language est omis, mb_language retourne le nom du langage courant, sous

forme de chaîne. Si aucun langage n'avait été configuré, mb_language retourne FALSE .

Voir aussi mb_send_mail .

## 6.19.38 mb_output_handler() : Fonction de traitement des affichages web

string **mb_output_handler** ( string contents , int status )

mb_output_handler est la fonction à fournir à ob_start . mb_output_handler convertit les caractères envoyés au client Web, dans l'encodage paramétré avec mb_http_output .

contents : Le contenu à traiter

status : L'état du contenu

mb_output_handler retourne la chaîne convertie.

Exemple avec mb_output_handler

```
<?php
 mb_http_output("UTF-8");
 ob_start("mb_output_handler");
?>
```

Note

Si vous souhaitez envoyer des données binaires telles que des images issues d'un script PHP, vous devez spécifier l'encodage spécial "pass" , avec la fonction mb_http_output .

Voir aussi ob_start .

## 6.19.39 mb_parse_str() : Analyse les données HTTP GET/POST/COOKIE et assigne les variables globales

bool **mb_parse_str** ( string encoded_string , *array result* )

mb_parse_str analyse les données d'entrées HTTP GET/POST/COOKIE et assigne les variables globales. Etant donné que PHP ne fournit pas de valeurs brutes de POST/COOKIE, cette fonction n'est utilisable que sur les données en méthode GET. mb_parse_str prend les données de l'URL appelante, détecte le jeu de caractères, converti les données en jeu de caractères interne, et affecte les valeurs au tableau de variables globales.

encoded_string : Les données encodées de l'URL.

result : Un tableau contenant les valeurs décodées, et les noms des jeux de caractères.

mb_parse_str retourne TRUE en cas de succès, et FALSE sinon.

Voir aussi mb_detect_order et mb_internal_encoding .

## 6.19.40 mb_preferred_mime_name() : Détecte l'encodage MIME

string **mb_preferred_mime_name** ( string encoding )

mb_preferred_mime_name retourne le type d'encodage MIME utilisé dans le mail encoding . Le nom de l'encodage est retourné sous forme de chaîne.

Exemple avec mb_preferred_mime_string

```
<?php
 $outputenc = "sjis-win";
 mb_http_output($outputenc);
 ob_start("mb_output_handler");
 header("Content-Type: text/html; charset=" . mb_preferred_mime_name($outputenc));
?>
```

## 6.19.41 mb_regex_encoding() : Retourne le jeu de caractères courant pour les expressions régulières

string **mb_regex_encoding** ( *string encoding* )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

mb_regex_encoding retourne le jeu de caractères courant pour les expressions régulières, sous forme de chaîne.

Si le troisième argument optionnel encoding est fourni, il prendra la valeur du jeu de caractères par défaut pour les expressions régulières. Par défaut, cette option est identique au jeu de caractères interne.

Note

Cette fonction est supportée depuis PHP 4.2.0.

Voir aussi mb_internal_encoding et mb_ereg

## 6.19.42 mb_regex_set_options() : Lit et modifie les options des fonctions d'expression régulières à support de caractères multi-octets

string **mb_regex_set_options** ( *string options* )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

mb_regex_set_options configure les options par défaut avec les nouvelles valeurs contenues dans options , pour les fonctions d'expressions régulières à support de caractères multi-octets.

mb_regex_set_options retourne la configuration précédente. Si options est omise, la fonction retourne la chaîne qui décrit les options courantes.

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.3.0. |

Voir aussi mb_split , mb_ereg et mb_eregi

## 6.19.43 mb_send_mail() : Envoie un mail encodé ISO−2022−JP (mail japonais)

bool **mb_send_mail** ( string to , string subject , string message , *string additional_headers* , *string additional_parameter* )

mb_send_mail envoie un courrier électronique. Les en−têtes et le corps du mesage sont convertis et encodés en ISO−2022−JP. mb_send_mail est une version adaptée de mail .

to est l'adresse de destination du mail. Les adresses multiples peuvent être spécifiées en les séparant par des virgules.

subject est le sujet du mail.

message est le message du mail.

La chaîne additional_headers est insérée à la fin de l'en−tête mail. Elle sert à ajouter d'autres en−têtes email. N'oubliez pas de les séparer par des nouvelles lignes (\n).

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi mail , mb_encode_mimeheader et mb_language .

## 6.19.44 mb_split() : Scinde une chaîne en tableau avec une expression régulière multi−octets

array **mb_split** ( string pattern , string string , *int limit* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

mb_split scinde la chaîne mutli−octets string en utilisant l'expression régulière pattern puis en retournant le résultat sous forme de tableau.

Si le paramètre optionnel limit est spécifié, la chaîne sera scindée en limit éléments au plus.

Le jeu de caractères interne, ou le jeu de caractères est spécifié par mb_regex_encoding .

| Note |
| --- |
| Cette fonction est supportée depuis PHP 4.2.0. |

Voir aussi mb_regex_encoding et mb_ereg .

## 6.19.45 mb_strcut() : Coupe une partie de chaîne

string **mb_strcut** ( string str , int start , *int length* , *string encoding* )

mb_strcut retourne la portion de la chaîne str qui commence au caractère start et a la longueur de length caractères.

mb_strcut effectue une recherche de type strpos , en tenant compte des caractères multi−octets. La position de needle est comptée à partir du début de la chaîne haystack : les positions commencent à 0.

mb_strcut soustrait la partie de la chaîne str qui compte length caractères.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi mb_substr et mb_internal_encoding .

## 6.19.46 mb_strimwidth() : Tronque une chaîne

string **mb_strimwidth** ( string str , int start , int width , string trimmarker , *string encoding* )

mb_strimwidth tronque la chaîne str à la longueur width . Elle retourne la chaîne tronquée.

SI trimmarker est fourni, trimmarker est ajoutée à la fin de la chaîne retournée.

start est l'offet de départ, en nombre de caractères depuis le début de la chaîne (cela commence à 0).

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Exemple avec mb_strimwidth

```
<?php
 $str = mb_strimwidth($str, 0, 40, "..>");
?>
```

Voir aussi mb_strwidth et mb_internal_encoding .

## 6.19.47 mb_strlen() : Retourne la taille d'une chaîne

string **mb_strlen** ( string str , *string encoding* )

mb_strlen retourne le nombre de caractères dans la chaîne str , avec l'encodage encoding . Un caractère multi−octets est alors compté pour 1.

Voir aussi mb_internal_encoding , strlen .

## 6.19.48 mb_strpos() : Repère la première occurence d'un caractère dans une chaîne

int **mb_strpos** ( string haystack , string needle , *int offset* , *string encoding* )

mb_strpos retourne la position numérique de la première occurence du caractère needle dans la chaîne haystack . Si needle est introuvable, mb_strpos retourne FALSE .

mb_strpos effectue une recherche de type strpos , en tenant compte des caractères multi−octets. La position de needle est comptée à partir du début de la chaîne haystack : les positions commencent à 0.

Si encoding est omis, l'encodage interne par défaut est utilisé. mb_strrpos accepte des chaînes comme argument needle , alors que strrpos n'accepte que des caractères.

offset est l'offset de début de recherche. S'il est omis, il sera utilisé à 0 (début de la chaîne).

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi mb_strpos , mb_internal_encoding et strpos

## 6.19.49 mb_strrpos() : Repère la dernière occurence d'un caractère dans une chaîne

int **mb_strrpos** ( string haystack , string needle , *string encoding* )

mb_strrpos retourne la position numérique de la première occurence du caractère needle dans la chaîne haystack . Si needle est introuvable, mb_strrpos retourne FALSE .

mb_strrpos effectue une recherche de type strpos , en tenant compte des caractères multi−octets. La position de needle est comptée à partir du début de la chaîne haystack : les positions commencent à 0.

Si encoding est omis, l'encodage interne par défaut est utilisé. mb_strrpos accepte des chaînes comme argument needle , alors que strrpos n'accepte que des caractères.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi mb_strpos , mb_internal_encoding et strrpos .

## 6.19.50 mb_strtolower() : Met tous les caractères en minuscules

string **mb_strtolower** ( string str , *string encoding* )

mb_strtolower retourne la chaîne str après avoir converti tous les caractères alphabétiques en minuscules.

encoding spécifie le jeu de caractères de la chaîne str ; si il est omis, le jeu de caractères interne sera utilisé.

Pour plus d'informations sur les propriétés de l'Unicode, voyez http://www.unicode.org/unicode/reports/tr21/ .

Contrairement à strtolower , le concept de caractère 'alphabétique' est déterminé par les propriétés Unicode. De ce fait, le comportement de cette fonction n'est pas modifié par les configurations locales, et elle peut convertir tout les caractères qui sont considérés comme alphabétiques comme le c cédille (Ç).

| Exemple avec mb_strtolower |
| --- |
| `$str = "Marie A Un Petit Agneau Et Elle L'Aime BEAUCOUP.";`<br>`$str = mb_strtolower($str);`<br>`echo $str; // marie a un petit agneau et elle l'aime beaucoup` |

Voir aussi strtolower , mb_strtoupper et mb_convert_case .

## 6.19.51 mb_strtoupper() : Met tous les caractères en majuscules

string **mb_strtoupper** ( string str , *string encoding* )

mb_strtoupper retourne la chaîne str après avoir converti tous les caractères alphabétiques en majuscules.

encoding spécifie le jeu de caractères de la chaîne str ; si il est omis, le jeu de caractères interne sera utilisé.

Pour plus d'informations sur les propriétés de l'Unicode, voyez http://www.unicode.org/unicode/reports/tr21/ .

Contrairement à strtoupper , le concept de caractère 'alphabétique' est déterminé par les propriétés Unicode. De ce fait, le comportement de cette fonction n'est pas modifié par les configurations locales, et elle peut convertir tout les caractères qui sont considérés comme alphabétiques comme le c cédille (ç).

| Exemple avec mb_strtoupper |
| --- |
| `$str = "Marie A Un Petit Agneau Et Elle L'Aime BEAUCOUP.";`<br>`$str = mb_strtoupper($str);`<br>`echo $str; // MARIE A UN PETIT AGNEAU ET ELLE L'AIME BEAUCOUP.` |

Voir aussi strtoupper , mb_strtolower et mb_convert_case .

## 6.19.52 mb_strwidth() : Retourne la largeur d'une chaîne

int **mb_strwidth** ( string str , *string encoding* )

mb_strwidth retourne la largeur de la chaîne str .

Les chaînes à encodage multi−octet sont généralement deux fois plus grandes que les chaînes à simple−octet.

| Caractères | Taille |
| --- | --- |
| U+0000 − U+0019 | 0 |
| U+0020 − U+1FFF | 1 |
| U+2000 − U+FF60 | 2 |
| U+FF61 − U+FF9F | 1 |
| U+FFA0 − | 2 |

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi mb_strimwidth et mb_internal_encoding .

## 6.19.53 mb_substitute_character() : Lit/modifie les caractères de substitution

mixed **mb_substitute_character** ( *mixed substrchar* )

mb_substitute_character spécifie le caractère de substitution des caractères invalides, ou des encodages invalides. Les caractères invalides peuvent être remplacés par NULL (pas d'affichage, ils sont supprimés), une chaîne ou un code hexadécimal.

Ce paramétrage affecte mb_detect_encoding et mb_send_mail .

substchar spécifie une valeur Unicode sous la forme d'un entier, ou bien une chaîne sous ces formes :

- "none" : pas d'affichage
- "long" : affiche la valeur hexadécimale (Par exemple : U+3000,JIS+7E7E)

SI substchar est fourni, mb_substitute_character retourne TRUE en cas de succès, et FALSE en cas d'erreur. Si substchar est omis, mb_substitute_character retourne une valeur Unicode, ou bien " none "/" long ".

Exemple avec mb_substitute_character

```
<?php
/* Configure le caractère de substitution avec U+3013 (GETA MARK) */
mb_substitute_character(0x3013);
/* Configure le caractère de substitution avec un format hexadécimal */
mb_substitute_character("long");
/* Affiche la configuration courante */
echo mb_substitute_character();
?>
```

## 6.19.54 mb_substr_count() : Compte le nombre d'occurrences d'une sous−chaîne

int **mb_substr_count** ( string haystack , string needle , *string encoding* )

mb_substr_count retourne le nombre d'occurrences de la chaîne needle dans la chaîne haystack string.

encoding spécifie le jeu de caractères utilisé pour needle et haystack . Si omis, le jeu de caractères interne est utilisé.

Exemple avec mb_substr_count

```
<?php
print mb_substr_count("Ceci est un test", "es");
// affiche 2
?>
```

Voir aussi substr_count , mb_strpos et mb_substr .

## 6.19.55 mb_substr() : Lit une sous−chaîne

string **mb_substr** ( string str , int start , *int length* , *string encoding* )

mb_substr retourne la portion de la chaîne str qui commence au caractère start et a la longueur de length caractères.

mb_substr effectue une recherche de type strpos , en tenant compte des caractères multi−octets. La position de needle est comptée à partir du début de la chaîne haystack : les positions commencent à 0.

encoding est un nom d'encodage de caractères. S'il n'est pas spécifié, l'encodage interne est utilisé.

Voir aussi mb_strcut et mb_internal_encoding .

# 6.20 Chiffrage mcrypt

## 6.20.1 Introduction

Ces fonctions permettent d'accéder à la librairie mcrypt, qui dispose d'une grande variété d'algorithmes de chiffrage, tels que DES, TripleDES, Blowfish (par défaut), 3−WAY, SAFER−SK64, SAFER−SK128, TWOFISH, TEA, RC2 et GOST en modes CBC, OFB, CFB et ECB. De plus, elle accepte aussi RC6 et IDEA qui sont considérés comme "non−libres".

## 6.20.2 Pré−requis

Ces fonctions utilisent mcrypt . Pour utiliser cette librairie, téléchargez le fichier libmcrypt−x.x.tar.gz sur ce site et suivez les instructions d'installation fournies. Les utilisateurs Windows trouverons toutes librairies nécessaires ici .

Si vous compilez PHP avec la librairie libmcrypt 2.4.x , les algorithmes suivants sont supportés : CAST, LOKI97, RIJNDAEL, SAFERPLUS, SERPENT ainsi que les chiffrements suivants : ENIGMA (chiffrage), PANAMA, RC4 et WAKE. Avec libmcrypt 2.4.x un autre mode de chiffrement est disponible : nOFB.

## 6.20.3 Installation

Vous devez compiler PHP avec l'option −−with−mcrypt pour activer cette extension.

## 6.20.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Changeable |
|---|---|---|
| mcrypt.algorithms_dir | NULL | PHP_INI_ALL |
| mcrypt.modes_dir | NULL | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

## 6.20.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.20.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Mcrypt peut opérer en 4 modes de chiffrage (CBC, OFB, CFB, et ECB). Si vous utilisez libmcrypt−2.4.x ou plus récent, les fonctions peuvent aussi opérer en mode nOFB et en mode STREAM. Nous allons présenter la technique d'utilisation de ces modes. Pour plus de références et de détails, reportez−vous au livre suivant : Applied Cryptography by Schneier (ISBN

0−471−11709−9).

- ECB (electronic codebook) ECB (electronic codebook) est prévu pour des données aléatoires, telles que des clés. Etant donné que les données sont peu nombreuses et aléatoires, les inconvénients de l'ECB ont ici un effet négatif favorable.
- CBC (cipher block chaining) est spécialement pratique avec les fichiers dont la sécurité ECB n'est pas suffisante.
- CFB (cipher feedback) est la meilleure méthode pour chiffrer des flots d'octets, quand les octets doivent être encryptés un par un.
- OFB (output feedback) est comparable à CFB, mais peut être utilisé lorsque des erreurs ne doivent pas être propagées.
- nOFB (output feedback, in nbit) est comparable à OFB, mais plus sûr, car il opère avec la taille de blocs de l'algorithme.
- STREAM est un mode supplémentaire, pour permettre l'utilisation d' algorithmes tels que WAKE ou RC4.

Voici quelques autres modes et méthodes de compression :

*MCRYPT_ENCRYPT (entier)*
*MCRYPT_DECRYPT (entier)*
*MCRYPT_DEV_RANDOM (entier)*
*MCRYPT_DEV_URANDOM (entier)*
*MCRYPT_RAND (entier)*

## 6.20.7 Mcrypt ciphers

Voici une liste non exhaustive des modes de chiffrement de l'extension mcrypt. Pour disposer d'une liste complète des chiffrements supportés, voyez les définitions dans le fichier mcrypt.h . La règle générale est avec l'API mcrypt−2.2.x API est que vous pouvez accéder au mode de chiffrement depuis PHP avec la constante MCRYPT_ciphername. Avec la librairie libmcrypt−2.4.x et libmcrypt−2.5.x, ces constantes fonctionnent toujours, mais il est possible de spécifier le nom du chiffrement dans une chaîne, lors de l'appel à mcrypt_module_open .

- MCRYPT_3DES
- MCRYPT_ARCFOUR_IV (libmcrypt > 2.4.x seulement)
- MCRYPT_ARCFOUR (libmcrypt > 2.4.x seulement)
- MCRYPT_BLOWFISH
- MCRYPT_CAST_128
- MCRYPT_CAST_256
- MCRYPT_CRYPT
- MCRYPT_DES
- MCRYPT_DES_COMPAT (libmcrypt 2.2.x seulement)
- MCRYPT_ENIGMA (libmcrypt > 2.4.x seulement, alias de MCRYPT_CRYPT )
- MCRYPT_GOST
- MCRYPT_IDEA (non−free)
- MCRYPT_LOKI97 (libmcrypt > 2.4.x seulement)
- MCRYPT_MARS (libmcrypt > 2.4.x seulement, non−libre)
- MCRYPT_PANAMA (libmcrypt > 2.4.x seulement)
- MCRYPT_RIJNDAEL_128 (libmcrypt > 2.4.x seulement)
- MCRYPT_RIJNDAEL_192 (libmcrypt > 2.4.x seulement)
- MCRYPT_RIJNDAEL_256 (libmcrypt > 2.4.x seulement)
- MCRYPT_RC2
- MCRYPT_RC4 (libmcrypt 2.2.x seulement)

- MCRYPT_RC6 (libmcrypt > 2.4.x seulement)
- MCRYPT_RC6_128 (libmcrypt 2.2.x seulement)
- MCRYPT_RC6_192 (libmcrypt 2.2.x seulement)
- MCRYPT_RC6_256 (libmcrypt 2.2.x seulement)
- MCRYPT_SAFER64
- MCRYPT_SAFER128
- MCRYPT_SAFERPLUS (libmcrypt > 2.4.x seulement)
- MCRYPT_SERPENT (libmcrypt > 2.4.x seulement)
- MCRYPT_SERPENT_128 (libmcrypt 2.2.x seulement)
- MCRYPT_SERPENT_192 (libmcrypt 2.2.x seulement)
- MCRYPT_SERPENT_256 (libmcrypt 2.2.x seulement)
- MCRYPT_SKIPJACK (libmcrypt > 2.4.x seulement)
- MCRYPT_TEAN (libmcrypt 2.2.x seulement)
- MCRYPT_THREEWAY
- MCRYPT_TRIPLEDES (libmcrypt > 2.4.x seulement)
- MCRYPT_TWOFISH (pour les vieilles versions mcrypt 2.x, ou mcrypt > 2.4.x )
- MCRYPT_TWOFISH128 (les TWOFISH xxx sont disponibles dans les nouvelles versions 2.x, mais pas dans les versions 2.4.x)
- MCRYPT_TWOFISH192
- MCRYPT_TWOFISH256
- MCRYPT_WAKE (libmcrypt > 2.4.x seulement)
- MCRYPT_XTEA (libmcrypt > 2.4.x seulement)

Vous devez (mode OFB et OFB ) ou pouvez (mode CBC ) fournir un vecteur d'initialisation (IV) pour ces modes de chiffrement. IV doit être unique, et avoir la même valeur au chiffrement et au déchiffrement. Pour des données qui seront enregistrées après chiffrement, vous pouvez prendre le résultat d'une fonction telle que MD5, appliquée sur le nom du fichier. Sinon, vous pouvez envoyer IV avec les données chiffrées, (reportez−vous au chapitre 9.3 de Applied Cryptography by Schneier (ISBN 0−471−11709−9) by Schneier (ISBN 0−471−11709−9) pour plus de détails sur le sujet).

## 6.20.8 Exemples

Mcrypt permet de chiffrer et de déchiffrer, en utilisant les méthodes mentionnées ci−dessus. Les 4 commandes importantes mcrypt_cfb , mcrypt_cbc , mcrypt_ecb et mcrypt_ofb ) peuvent toutes opérer en mode MCRYPT_ENCRYPT et MCRYPT_DECRYPT .

Chiffre une valeur avec un TripleDES, en mode ECB.

```
<?php
 $key = "Cette cle est ultra−secrete";
 $input = "Rencontrons−nous dans notre place secrete a 9 h 00.";
 $encrypted_data = mcrypt_ecb(MCRYPT_TripleDES, $key, $input, MCRYPT_ENCRYPT);
?>
```

Cet exemple va retourner les données cryptées dans la variable $encrypted_data.

Si vous avez compilé PHP avec libmcrypt 2.4.x, ces fonctions sont toujours disponibles, mais il est vivement conseillé d'utiliser les nouvelles fonctions avancées.

Encryption d'une valeur avec TripleDES sous 2.4.x en mode ECB

```
<?php
 $key = "Ceci est une vraie cle secrete";
 $input = "Rendez−vous à 9 heures, dans notre planque.";
 $td = mcrypt_module_open (MCRYPT_TripleDES, "", MCRYPT_MODE_ECB, "");
 $iv = mcrypt_create_iv (mcrypt_enc_get_iv_size ($td), MCRYPT_RAND);
 mcrypt_generic_init ($td, $key, $iv);
 $encrypted_data = mcrypt_generic ($td, $input);
 mcrypt_generic_end ($td);
?>
```

Cet exemple va retourner les données cryptées dans la variable $encrypted_data .


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Mcrypt ciphers
- Exemples
- mcrypt_cbc
- mcrypt_cfb
- mcrypt_create_iv
- mcrypt_decrypt
- mcrypt_ecb
- mcrypt_enc_get_algorithms_name
- mcrypt_enc_get_block_size
- mcrypt_enc_get_iv_size
- mcrypt_enc_get_key_size
- mcrypt_enc_get_modes_name
- mcrypt_enc_get_supported_key_sizes
- mcrypt_enc_is_block_algorithm_mode
- mcrypt_enc_is_block_algorithm
- mcrypt_enc_is_block_mode
- mcrypt_enc_self_test
- mcrypt_encrypt
- mcrypt_generic_deinit
- mcrypt_generic_end
- mcrypt_generic_init
- mcrypt_generic
- mcrypt_get_block_size
- mcrypt_get_cipher_name
- mcrypt_get_iv_size
- mcrypt_get_key_size
- mcrypt_list_algorithms
- mcrypt_list_modes
- mcrypt_module_close
- mcrypt_module_get_algo_block_size
- mcrypt_module_get_algo_key_size
- mcrypt_module_get_supported_key_sizes
- mcrypt_module_is_block_algorithm_mode
- mcrypt_module_is_block_algorithm
- mcrypt_module_is_block_mode
- mcrypt_module_open
- mcrypt_module_self_test
- mcrypt_ofb
- mdecrypt_generic


## 6.20.10 mcrypt_cfb() : Chiffre/déchiffre des données en mode CFB

string **mcrypt_cfb** ( int cipher , string key , string data , int mode , string iv ) string **mcrypt_cfb** ( string cipher , string key , string data , int mode , *string iv* )

La première syntaxe utilise libmcrypt 2.2.x, et la seconde utilise libmcrypt 2.4.x.

mcrypt_cfb ne doit plus être utilisée. Vous pouvez la remplacer par mcrypt_generic et mdecrypt_generic .

## 6.20.11 mcrypt_create_iv() : Crée un vecteur d'initialisation à partir d'une source aléatoire.

string **mcrypt_create_iv** ( int size , int source )

mcrypt_create_iv sert à créer un IV (vecteur d'initialisation).

mcrypt_create_iv prend deux arguments, size détermine la taille de IV, source spécifie la source de IV.

La source peut être MCRYPT_RAND (générateur de nombres aléatoires système), MCRYPT_DEV_RANDOM (lecture des données depuis le fichier /dev/random) et MCRYPT_DEV_URANDOM (lecture des données depuis le fichier /dev/urandom). Si vous utilisez MCRYPT_RAND , assurez−vous de bien appeler srand pour initialiser le générateur de nombres aléatoires.

Exemple avec mcrypt_create_iv

```
<?php
   $size = mcrypt_get_iv_size (MCRYPT_CAST_256, MCRYPT_MODE_CFB);
   $iv = mcrypt_create_iv ($size, MCRYPT_DEV_RANDOM);
?>
```

Le VI est uniquement une solution alternative pour initialiser les routines de chiffrement. Le VI n'a pas besoin d'être spécifiquement secret, même si c'est mieux. Vous pouvez même l'envoyer avec vos données chiffrées sans perdre de sécurité.

Plus d'informations sont disponibles sur les sites de http://www.ciphersbyritter.com/GLOSSARY.HTM#IV , http://fn2.freenet.edmonton.ab.ca/~jsavard/crypto/co0409.htm et dans le chapitre 9.3 du livre "Applied Cryptography" de Schneier (ISBN 0−471−11709−9).

## 6.20.12 mcrypt_decrypt() : Déchiffre un texte

string **mcrypt_decrypt** ( string cipher , string key , string data , string mode , *string iv* )

cipher est une constante MCRYPT_ciphername qui indique le nom de l'algorithme sous forme de chaîne.

key est la clé utilisée pour chiffrer les données. Si elle est plus petite que nécessaire, elle sera complétée avec des ' \0 '.

data sont les données qui doivent être encryptées. Si la taille des données n'est pas de la forme n * taille_de_bloc, elles seront complétées avec des ' \0 '. La valeur retournée peut être plus grande que la valeur d'origine.

mode est une constante MCRYPT_MODE_modename qui peut valoir : "ecb", "cbc", "cfb", "ofb", "nofb" ou "stream".

iv (Vecteur d'Initialisation) est utilisé pour les modes CBC, CFB, OFB, et dans certains algorithmes de mode STREAM. Si vous ne fournissez pas le VI, alors qu'il est nécessaire, la fonction affichera une alerte, et utilise un VI composé de caractères ' \0 '.

## 6.20.13 mcrypt_ecb() : Chiffre/déchiffre des données en mode ECB

string **mcrypt_ecb** ( int cipher , string key , string data , int mode ) string **mcrypt_ecb** ( string cipher , string key , string data , int mode , *string iv* )

La première syntaxe utilise libmcrypt 2.2.x, et la seconde utilise libmcrypt 2.4.x.

mcrypt_ecb ne doit plus être utilisée. Vous pouvez la remplacer par mcrypt_generic et mdecrypt_generic .

## 6.20.14 mcrypt_enc_get_algorithms_name() : Retourne le nom de l'algorithme

string **mcrypt_enc_get_algorithms_name** ( resource td )

mcrypt_enc_get_algorithms_name retourne le nom de l'algorithme désigné par td .

Exemple avec mcrypt_enc_get_algorithms_name

```
<?php
   $td = mcrypt_module_open (MCRYPT_CAST_256, '', MCRYPT_MODE_CFB, '');
   echo mcrypt_enc_get_algorithms_name($td). "\n";

   $td = mcrypt_module_open ('cast-256', '', MCRYPT_MODE_CFB, '');
   echo mcrypt_enc_get_algorithms_name($td). "\n";
?>

Affiche :
CAST-256
CAST-256
```

## 6.20.15 mcrypt_enc_get_block_size() : Retourne la taille de blocs d'un algorithme

int **mcrypt_enc_get_block_size** ( resource td )

mcrypt_enc_get_block_size retourne la taille de blocs d'un algorithme, en octets.

## 6.20.16 mcrypt_enc_get_iv_size() : Retourne la taille du VI d'un algorithme

int **mcrypt_enc_get_iv_size** ( resource td )

mcrypt_enc_get_iv_size retourne la taille du VI de l'algorithme désigné par td , en octets. Si la valeur retournée est 0, c'est que l'algorithme ne demande pas de VI. Un VI est demandé en mode cbc, cfb et ofb, et parfois en mode stream.

## 6.20.17 mcrypt_enc_get_key_size() : Retourne la taille maximale de la clé pour un mode

int **mcrypt_enc_get_key_size** ( resource td )

mcrypt_enc_get_key_size retourne la taille maximale de clé acceptée par le mode désigné par td , en octets.

## 6.20.18 mcrypt_enc_get_modes_name() : Retourne le nom du mode

string **mcrypt_enc_get_modes_name** ( resource td )

mcrypt_enc_get_modes_name retourne le nom du mode désigné par td .

Exemple avec mcrypt_enc_get_modes_name

```
<?php
   $td = mcrypt_module_open (MCRYPT_CAST_256, '', MCRYPT_MODE_CFB, '');
   echo mcrypt_enc_get_modes_name($td). "\n";

   $td = mcrypt_module_open ('cast-256', '', 'ecb', '');
   echo mcrypt_enc_get_modes_name($td). "\n";
?>

Affiche :
CFB
ECB
```

## 6.20.19 mcrypt_enc_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés acceptées par un algorithme

array **mcrypt_enc_get_supported_key_sizes** ( resource td )

mcrypt_enc_get_supported_key_sizes retourne un tableau contenant les tailles des clés supportées par l'algorithme désigné par td . S'il retourne un tableau vide, c'est que toutes les clés entre 1 et mcrypt_enc_get_key_size sont acceptées par l'algorithme.

Exemple avec mcrypt_enc_get_supported_key_sizes

```
<?php
   $td = mcrypt_module_open ('rijndael-256', '', 'ecb', '');
   var_dump (mcrypt_enc_get_supported_key_sizes($td));
?>

Affiche :

array(3) {
  [0]=>
  int(16)
  [1]=>
  int(24)
  [2]=>
  int(32)
}
?>
```

## 6.20.20 mcrypt_enc_is_block_algorithm_mode() : Teste le chiffrage par blocs d'un mode

bool **mcrypt_enc_is_block_algorithm_mode** ( resource td )

mcrypt_enc_is_block_algorithm_mode retourne 1 si ce mode utilise des algorithmes par blocs, et 0 sinon. (i.e. 0 pour stream, et 1 pour cbc, cfb, ofb).

## 6.20.21 mcrypt_enc_is_block_algorithm() : Teste le chiffrage par blocs d'un algorithme

bool **mcrypt_enc_is_block_algorithm** ( resource td )

mcrypt_enc_is_block_algorithm retourne 1 si l'algorithme utilisé est un algorithme par blocs, et 0 si c'est un algorithme par flux.

## 6.20.22 mcrypt_enc_is_block_mode() : Teste si le mode retourne les données par blocs

bool **mcrypt_enc_is_block_mode** ( resource td )

mcrypt_enc_is_block_mode retourne 1 si le mode retourne des blocs d'octets, ou bien 0 s'il retourne des octets (par flux). (i.e. 1 pour cbc et ecb, et 0 pour cfb et stream).

## 6.20.23 mcrypt_enc_self_test() : Teste un module ouvert

bool **mcrypt_enc_self_test** ( resource td )

mcrypt_enc_self_test effectue un test du module ouvert et désigné par td . Si le test est concluant, elle retourne 0, sinon, 1.

## 6.20.24 mcrypt_encrypt() : Chiffre un texte

string **mcrypt_encrypt** ( string cipher , string key , string data , string mode , *string iv* )

mcrypt_encrypt chiffre les données, et retourne les données cryptées.

cipher est une constante MCRYPT_ciphername qui indique le nom de l'algorithme sous forme de chaîne.

key est la clé utilisée pour chiffrer les données. Si elle est plus petite que nécessaire, elle sera complétée avec des ' \0 '.

data sont les données qui doivent être encryptées. Si la taille des données n'est pas de la forme n * taille_de_bloc, elles seront complétées avec des ' \0 '. La valeur retournée peut être plus grande que la valeur d'origine.

mode est une constante MCRYPT_MODE_modename qui peut valoir : "ecb", "cbc", "cfb", "ofb", "nofb" ou "stream".

iv (Vecteur d'Initialisation) est utilisé pour les modes CBC, CFB, OFB, et dans certains algorithmes de mode STREAM. Si vous ne fournissez pas le VI, alors qu'il est nécessaire, la fonction affichera une alerte, et utilisera un vecteur d'initialisation composé de caractères ' \0 '.

Exemple avec mcrypt_encrypt

```
<?php
 $iv = mcrypt_create_iv(mcrypt_get_iv_size (MCRYPT_RIJNDAEL_256, MCRYPT_MODE_ECB), MCRYPT_RAND);
 $key = "Ceci est une clé secrète";
 $text = "Rencontrons nous à 11 heures, derrière le monument";
 echo strlen ($text)."\n";
```

```
  $crypttext = mcrypt_encrypt (MCRYPT_RIJNDAEL_256, $key, $text, MCRYPT_MODE_ECB, $iv);
  echo strlen ($crypttext)."\n";
?>
```

L'exemple ci−dessus affichera :
42
64

Voir aussi mcrypt_module_open .

## 6.20.25 mcrypt_generic_deinit() : Prépare le module pour le déchargement

bool **mcrypt_generic_deinit** ( resource td )

mcrypt_generic_deinit prépare le module de chiffrement td pour le déchargement. Tous les buffers sont vidées, mais le module n'est pas déchargé. Vous devez appeler mcrypt_module_close vous même (mais PHP le fera pour vous a la fin du script). mcrypt_generic_deinit retourne FALSE en cas d'erreur, et TRUE sinon.

Pour un exemple, voir mcrypt_module_open et l'entrée sur mcrypt_generic_init .

## 6.20.26 mcrypt_generic_end() : Termine un chiffrage

bool **mcrypt_generic_end** ( resource td )

mcrypt_generic_end termine le chiffrage désigné par le pointeur td . En fait, elle supprime tous les buffers, et ferme les modules utilisés. Elle retourne FALSE en cas d'erreur, et TRUE sinon.

## 6.20.27 mcrypt_generic_init() : Initialise tous les buffers nécessaires

int **mcrypt_generic_init** ( resource td , string key , string iv )

La taille maximale de la clé doit être cette retournée par mcrypt_enc_get_key_size et toutes les valeurs inférieures seront aussi valides. Le vecteur d'initialisation (VI) doit avoir la taille d'un bloc, mais vous devez lire sa taille en appelant mcrypt_enc_get_iv_size . IV est ignoré en mode ECB. IV DOIT exister en modes CFB, CBC, STREAM, nOFB et OFB. Il doit être aléatoire et unique (mais pas secret). Le même VI doit être utilisé pour le chiffrage et le déchiffrage. Si vous ne voulez pas l'utiliser, remplissez le de zéros, mais ce n'est pas recommandé. La fonction retourne (−1) en cas d'erreur.

Vous devez appeler mcrypt_generic_init avant chaque appel à mcrypt_generic ou mdecrypt_generic .

## 6.20.28 mcrypt_generic() : Chiffre

string **mcrypt_generic** ( resource td , string data )

mcrypt_generic chiffre les données data . Les données sont complétées par des " \0 " pour obtenir une taille multiple de la taille d'un bloc. Elle retourne les données chiffrées. Notez que la longueur

de la chaîne retournée peut être plus longue que celle passée en argument, à cause du complément.

Le gestionnaire de chiffrement td doit être initialisée avec la fonction mcrypt_generic_init , avec une clé et un VI, avant d'appeler cette fonction. Lorsque le chiffrement est réalisé, vous devez libérer les buffers en appelant la fonction mcrypt_generic_deinit . Voyez mcrypt_module_open pour un exemple.

Voir aussi mdecrypt_generic , mcrypt_generic_init et mcrypt_generic_deinit .

## 6.20.29 mcrypt_get_block_size() : Retourne la taille de blocs d'un chiffrement

int **mcrypt_get_block_size** ( int cipher ) int **mcrypt_get_block_size** ( string cipher , string module )

mcrypt_get_block_size sert à lire la taille de blocs du chiffrement cipher (en combinaison avec un mode de chiffrement.

Le premier prototype sert lorsque PHP est compilé avec la librairie libmcrypt 2.2.x, le second lorsque PHP est compilé avec libmcrypt 2.4.x ou 2.5.x.

Cet exemple montre comment utiliser cette fonction lorsque PHP est compilé avec libmcrypt 2.4.x et 2.5.x.

Exemple avec mcrypt_get_block_size

```
<?php
   echo mcrypt_get_block_size ('tripledes', 'ecb');
?>

Affiche :
8
```

Voir aussi mcrypt_get_key_size et mcrypt_encrypt .

## 6.20.30 mcrypt_get_cipher_name() : Lit le nom du chiffrement utilisé

string **mcrypt_get_cipher_name** ( int cipher ) string **mcrypt_get_cipher_name** ( string cipher )

mcrypt_get_cipher_name retourne le nom du chiffrement utilisé.

mcrypt_get_cipher_name prend le numéro de chiffrement (avec libmcrypt 2.2.x) ou prend le nom du chiffrement (avec libmcrypt 2.4.x) comme paramètre, et retourne le nom du chiffrement, ou FALSE , si ce chiffrement n'existe pas.

Exemple avec mcrypt_get_cipher_name

```
<?php
$cipher = MCRYPT_TripleDES;
print mcrypt_get_cipher_name($cipher);
?>
```

L'exemple ci−dessus va afficher :
3DES

## 6.20.31 mcrypt_get_iv_size() : Retourne la taille du VI utilisé par un couple chiffrement/mode

int **mcrypt_get_iv_size** ( resource td ) int **mcrypt_get_iv_size** ( string cipher , string mode )

La première syntaxe utilise libmcrypt 2.2.x, et la seconde utilise libmcrypt 2.4.x.

mcrypt_get_iv_size retourne la taille du Vecteur d'initialisation (VI). En cas d'erreur, la fonction retourne FALSE . Si le VI est ignoré dans le couple chiffrement/mode demandé, zéro est retourné.

cipher est une constante MCRYPT_ciphername qui indique le nom de l'algorithme sous forme de chaîne.

mode est une constante MCRYPT_MODE_modename qui peut valoir : "ecb", "cbc", "cfb", "ofb", "nofb" ou "stream".

Cet exemple illuste l'utilisation de cette fonction avec les deux librairies libmcrypt 2.4.x et 2.5.x.

```
Exemple avec mcrypt_get_block_size
<?php
   $size = mcrypt_get_iv_size (MCRYPT_CAST_256, MCRYPT_MODE_CFB);

   $size = mcrypt_get_iv_size ('des', 'ecb');
?>
```

Voir aussi mcrypt_get_block_size et mcrypt_create_iv .

## 6.20.32 mcrypt_get_key_size() : Retourne la taille de la clé d'un chiffrement

int **mcrypt_get_key_size** ( int cipher ) int **mcrypt_get_key_size** ( string cipher , string module )

mcrypt_get_key_size sert à lire la taille de clé du chiffrement cipher .

mcrypt_get_block_size prend comme argument le chiffrement cipher et retourne une taille en octets.

Cet exemple illustre comment utiliser cette fonction avec les deux librairies libmcrypt 2.4.x et 2.5.x.

```
Exemple avec mcrypt_get_block_size
<?php
   echo mcrypt_get_key_size ('tripledes', 'ecb');
?>

Affiche :
24
```

Voir aussi mcrypt_get_block_size et mcrypt_encrypt .

## 6.20.33 mcrypt_list_algorithms() : Liste tous les algorithmes de chiffrement supportés

array **mcrypt_list_algorithms** ( *string lib_dir* )

mcrypt_list_algorithms sert à lister tous les algorithmes de chiffrement de lib_dir . mcrypt_list_algorithms prend un argument optionnel, qui spécifie le dossier qui contient tous les algorithmes. S'il est omis, la valeur de mcrypt.algorithms_dir dans php.ini est utilisée.

Exemple avec mcrypt_list_algorithms

```
<?php
$algorithms = mcrypt_list_algorithms ("/usr/local/lib/libmcrypt");
foreach ($algorithms as $cipher) {
    echo $cipher."/n";
}
?>
```

L'exemple ci−dessus va afficher tous les algorithmes supportés dans le dossier " /usr/local/lib/libmcrypt ".

## 6.20.34 mcrypt_list_modes() : Liste tous les modes de chiffrement supportés

array **mcrypt_list_modes** ( *string* *lib_dir* )

mcrypt_list_algorithms sert à lister tous les modes de chiffrement de lib_dir . mcrypt_list_algorithms prend un argument optionnel, qui spécifie le dossier qui contient tous les algorithmes. S'il est omis, la valeur de mcrypt.algorithms_dir dans php.ini est utilisée.

Exemple avec mcrypt_list_modes

```
<?php
$modes = mcrypt_list_modes ();
foreach ($modes as $mode) {
    echo $mode."<BR>";";
}
?>
```

L'exemple ci−dessus va afficher tous les modes supportés dans le dossier " /usr/local/lib/libmcrypt ".

## 6.20.35 mcrypt_module_close() : Décharge le module de chiffrement

bool **mcrypt_module_close** ( resource td )

mcrypt_module_close décharge le module td .

Voir mcrypt_module_open .

## 6.20.36 mcrypt_module_get_algo_block_size() : Retourne la taille de blocs d'un algorithme

int **mcrypt_module_get_algo_block_size** ( string algorithm , *string* *lib_dir* )

mcrypt_module_get_algo_block_size retourne la taille de blocs d'un algorithme, en octets. Le paramètre optionnel lib_dir contient le chemin jusqu'au module de l'algorithme sur le système.

## 6.20.37 mcrypt_module_get_algo_key_size() : Retourne la taille maximale de clé

int **mcrypt_module_get_algo_key_size** ( string algorithm , *string* *lib_dir* )

mcrypt_module_get_algo_key_size retourne la taille maximale de la clé supportée par l'algorithme algorithm . Le paramètre optionnel lib_dir contient le chemin jusqu'au module de l'algorithme sur le système.

## 6.20.38 mcrypt_module_get_supported_key_sizes() : Retourne un tableau contenant les tailles de clés supportée par un algorithme de chiffrement

array **mcrypt_module_get_supported_key_sizes** ( string algorithm , *string lib_dir* )

mcrypt_module_get_supported_key_sizes retourne un tableau contenant les tailles de clés supportée par l'algorithme de chiffrement algorithm . Si il retourne un tableau vide, alors toutes les clés entre 1 et mcrypt_module_get_algo_key_size sont supportées par l'algorithme. Le paramètre optionnel lib_dir peut contenir le chemin du dossier d'installation du module, sur le système.

Voir aussi mcrypt_enc_get_supported_key_sizes qui est utiliés pour ouvrir les modules de chiffrement.

## 6.20.39 mcrypt_module_is_block_algorithm_mode() : Indique si un mode fonctionne par blocs

bool **mcrypt_module_is_block_algorithm_mode** ( string mode , *string lib_dir* )

mcrypt_module_is_block_algorithm_mode retourne TRUE si le mode doit être utilisé avec un algorithme par bloc, sinon retourne 0 (i.e. 0 pour stream, et 1 pour cbc, cfb, ofb). Le paramètre optionnel lib_dir contient le chemin jusqu'au module de l'algorithme sur le système.

## 6.20.40 mcrypt_module_is_block_algorithm() : Indique si un algorithme fonctionne par blocs

bool **mcrypt_module_is_block_algorithm** ( string algorithm , *string lib_dir* )

mcrypt_module_is_block_algorithm retourne TRUE si algorithm est un algorithme par bloc, sinon retourne 0. Le paramètre optionnel lib_dir contient le chemin jusqu'au module de l'algorithme sur le système.

## 6.20.41 mcrypt_module_is_block_mode() : Indique si un mode travaille par blocs

bool **mcrypt_module_is_block_mode** ( string mode , *string lib_dir* )

mcrypt_module_is_block_mode retourne TRUE si ce mode fournit des blocs d'octets, ou bien un flux d'octets. (i.e. 1 pour cbc et ecb, et 0 pour cfb et stream). Le paramètre optionnel lib_dir contient le chemin jusqu'au module de l'algorithme sur le système.

## 6.20.42 mcrypt_module_open() : Ouvre le module de l'algorithme et le mode à utiliser

resource **mcrypt_module_open** ( string algorithm , string algorithm_directory , string mode , string mode_directory )

mcrypt_module_open ouvre le module de l'algorithme et du mode à utiliser. Le nom de l'algorithme est spécifié par le paramètre algorithm (par exemple : "twofish"), ou bien une des constantes MCRYPT_ciphername. La librairie est refermée en appelant mcrypt_module_close, mais il n'est pas nécessaire d'appeler cette fonction si mcrypt_generic_end est utilisé. Normalement, mcrypt_module_open retourne un pointeur d'encryption, ou bien FALSE en cas d'erreur.

algorithm_directory et mode_directory servent à repérer les modules d'encryption. Si vous fournissez un nom de dossier, il sera utilisé. Si vous passez une chaîne vide ("""), la valeur utilisé par mcrypt.algorithms_dir ou mcrypt.modes_dir sera celle indiquée dans les directives de configuration. Lorsque ces paramètres ne sont pas fournis les valeurs par défaut, compilées avec la librairie sont utilisées. (généralement /usr/local/lib/libmcrypt).

Exemple avec mcrypt_module_open

```
<?php
 $td = mcrypt_module_open (MCRYPT_DES, "", MCRYPT_MODE_ECB, "/usr/lib/mcrypt-modes");
?>
```

L'exemple ci−dessus va essayer d'ouvrir le module de chiffrement par DES, dans le dossier par défaut, et le mode EBC dans le dossier /usr/lib/mcrypt−modes.

Using mcrypt_module_open in encryption

```
<?php
   /* Charge un chiffrement */
   $td = mcrypt_module_open ('rijndael-256', '', 'ofb', '');

   /* Crée le VI et détermine la taille de la clé */
   $iv = mcrypt_create_iv (mcrypt_enc_get_iv_size($td), MCRYPT_DEV_RANDOM);
   $ks = mcrypt_enc_get_key_size ($td);

   /* Crée la clé */
   $key = substr (md5 ('very secret key'), 0, $ks);

   /* Intialise le chiffrement */
   mcrypt_generic_init ($td, $key, $iv);

   /* Chiffre les données */
   $encrypted = mcrypt_generic ($td, 'This is very important data');

   /* Libère le gestionnaire de chiffrement */
   mcrypt_generic_deinit ($td);

   /* Initialise le module de chiffrement pour le déchiffrement */
   mcrypt_generic_init ($td, $key, $iv);

   /* Déchiffre les données */
   $decrypted = mdecrypt_generic ($td, $encrypted);

   /* Libère le gestionnaire de déchiffrement, et ferme le module */
   mcrypt_generic_deinit ($td);
   mcrypt_module_close ($td);

   /* Affiche la chaîne */
   echo trim ($decrypted)."\n";
?>
```

La première ligne de l'exemple ci−dessus va essayer d'ouvrir le chiffrement DES, dans le dossier par défaut, et le mode EBC dans le dossier /usr/lib/mcrypt−modes . Le second exemple utilise les chaînes comme nom pour le chiffrement et le mode. Cela ne fonctionne que si l'extension est compilée avec libmcrypt 2.4.x or 2.5.x.

Voir aussi mcrypt_module_close , mcrypt_generic , mdecrypt_generic , mcrypt_generic_init et mcrypt_generic_deinit .

## 6.20.43 mcrypt_module_self_test() : Teste un mode

bool **mcrypt_module_self_test** ( string algorithm , *string lib_dir* )

mcrypt_module_self_test effectue un test sur l'algorithme spécifié. Le paramètre optionnel lib_dir contient le chemin jusqu'au module de l'algorithme sur le système.

mcrypt_module_self_test retourne TRUE si le test fonctionne, et FALSE sinon.

## 6.20.44 mcrypt_ofb() : Chiffre/déchiffre des données en mode OFB

string **mcrypt_ofb** ( int cipher , string key , string data , int mode , string iv ) string **mcrypt_ofb** ( string cipher , string key , string data , int mode , *string iv* )

La première syntaxe utilise libmcrypt 2.2.x, et la seconde utilise libmcrypt 2.4.x.

mcrypt_ofb ne doit plus être utilisée. Vous pouvez la remplacer par mcrypt_generic et mdecrypt_generic .

## 6.20.45 mdecrypt_generic() : Déchiffre

string **mdecrypt_generic** ( resource td , string data )

mdecrypt_generic déchiffre les données data . Notez que la longueur de la chaîne déchiffrée peut être plus longue que la chaîne originale, car elle peut avoir été complétée par des caractères " \0 ".

Exemple avec mdecrypt_generic

```php
<?php
   /* Données */
   $key = 'Ceci est une très longue clé de chiffrement, voire même trop longue';
   $plain_text = 'Ceci sont des données importantes';

   /* Ouvre le module et crée un VI */
    $td = mcrypt_module_open ('des', '', 'ecb', '');
   $key = substr ($key, 0, mcrypt_enc_get_key_size ($td));
   $iv_size = mcrypt_enc_get_iv_size ($td);
   $iv = mcrypt_create_iv ($iv_size, MCRYPT_RAND);

   /* Initialise le module de chiffrement */
   if (mcrypt_generic_init ($td, $key, $iv) != -1) {

       /* Chiffre les données */
       $c_t = mcrypt_generic ($td, $plain_text);
       mcrypt_generic_deinit ($td);

       /* Réinitialise les buffers pour le déchiffrement */
       mcrypt_generic_init ($td, $key, $iv);
       $p_t = mdecrypt_generic ($td, $c_t);

       /* Nettoye */
       mcrypt_generic_deinit ($td);
       mcrypt_module_close ($td);
   }

   if (strncmp ($p_t, $plain_text, strlen($plain_text)) == 0) {
       echo "ok\n";
   } else {
       echo "erreur\n";
   }
?>
```

L'exemple ci−dessus montre comment vérifier que les données avant chiffrage sont bien les mêmes que celles après chiffrage/déchiffrage. Il est très important de réinitialiser le buffer de chiffrement avec mcrypt_generic_init avant que nous ne déchiffriez les données.

Le gestionnaire de déchiffrement doit toujours être initialisé par la fonction mcrypt_generic_init avec une clée et un VI avant d'appeler cette fonction. Lorsque le chiffrement est fait, il faut que vous libérirez les données chiffrées en appelant mcrypt_generic_deinit . Voir mcrypt_module_open pour un exemple.

Voir aussi mcrypt_generic , mcrypt_generic_init et mcrypt_generic_deinit .

# 6.21 Ming pour Flash

| Attention |
| --- |
| Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez ce module à vos risques et périls. |

## 6.21.1 Introduction

Ming n'est pas un acronyme.

Ming est une librairie open−source (LGPL) qui vous permet de créer des animations au format Flash. Ming supporte toutes les fonctionnalités de Flash 4 : les formes (shapes), les gradients, les images bitmaps (JPEG et PNG), les morphing (transformations d'une forme en une autre), les textes, actions, sprites (mini animations), le streaming MP3 et les transformations de couleurs. Le seul ajout futur est celui des événements sons.

Notez que toutes les distances spécifiées (longueurs, distances, tailles...) sont en "twips", c'est−à−dire 20 unités par pixels. C'est plus ou moins arbitraire, car le lecteur Flash fait une mise à l'échelle avec les valeurs qui lui sont fournis dans la balise embed, ou la frame courante si la balise embed n'est pas utilisée.

Ming propose de nombreux avantages par rapport à l'extension swf. Vous pouvez utiliser Ming sur tous les OS où vous pouvez compiler le code, tandis que swf est limité à Windows. Ming vous évite la déconcertante complexité du format SWF, en transformant les éléments des animations en objets PHP. Enfin, Ming est toujours en cours de développement et surveillé par son auteur : si vous souhaitez une nouvelle fonctionnalité, dites le lui : ming@opaque.net .

Ming et tous les objets cités ont été ajouté en PHP 4.0.5.

## 6.21.2 Pré−requis

Pour utiliser Ming avec PHP, vous devez d'abord installer la librairy Ming. Le code source et les instructions d'installation sont disponible sur la page d'accueil de Ming : http://ming.sourceforge.net/ , avec des exemples un tutorial et l'actualité Ming.

Téléchargez l'archive Ming. Décompressez la et allez dans le dossier Ming. Faites "make", puis "make install".

Cela va compiler le fichier libming.so et l'installer dans /usr/lib/ , et copier ming.h into /usr/include/ . Editez la ligne PREFIX= dans le fichier Makefile pour indiquer votre dossier d'installation.

## 6.21.3 Installation

Compilation CGI avec PHP (Unix)

```
mkdir <phpdir>/ext/ming
       cp php_ext/* <phpdir>/ext/ming
       cd <phpdir>
       ./buildconf
       ./configure −−with−ming <other config options>
```

Compilez et installez PHP comme d'habitude. Redémarrez votre serveur web si nécessaire.

Maintenant, soit vous ajoutez simplement extension=php_ming.so dans votre fichier php.ini , soit vous ajoutez la ligne dl('php_ming.so'); au début de tous vos scripts qui utilisent Ming.

## 6.21.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.21.5 Types de ressources

## 6.21.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*SWFBUTTON_HIT (entier)*
*SWFBUTTON_DOWN (entier)*
*SWFBUTTON_OVER (entier)*
*SWFBUTTON_UP (entier)*
*SWFBUTTON_MOUSEUPOUTSIDE (entier)*
*SWFBUTTON_DRAGOVER (entier)*
*SWFBUTTON_DRAGOUT (entier)*
*SWFBUTTON_MOUSEUP (entier)*
*SWFBUTTON_MOUSEDOWN (entier)*
*SWFBUTTON_MOUSEOUT (entier)*
*SWFBUTTON_MOUSEOVER (entier)*
*SWFFILL_RADIAL_GRADIENT (entier)*
*SWFFILL_LINEAR_GRADIENT (entier)*
*SWFFILL_TILED_BITMAP (entier)*
*SWFFILL_CLIPPED_BITMAP (entier)*
*SWFTEXTFIELD_HASLENGTH (entier)*
*SWFTEXTFIELD_NOEDIT (entier)*
*SWFTEXTFIELD_PASSWORD (entier)*
*SWFTEXTFIELD_MULTILINE (entier)*
*SWFTEXTFIELD_WORDWRAP (entier)*
*SWFTEXTFIELD_DRAWBOX (entier)*
*SWFTEXTFIELD_NOSELECT (entier)*
*SWFTEXTFIELD_HTML (entier)*
*SWFTEXTFIELD_ALIGN_LEFT (entier)*
*SWFTEXTFIELD_ALIGN_RIGHT (entier)*
*SWFTEXTFIELD_ALIGN_CENTER (entier)*
*SWFTEXTFIELD_ALIGN_JUSTIFY (entier)*
*SWFACTION_ONLOAD (entier)*
*SWFACTION_ENTERFRAME (entier)*
*SWFACTION_UNLOAD (entier)*
*SWFACTION_MOUSEMOVE (entier)*
*SWFACTION_MOUSEDOWN (entier)*
*SWFACTION_MOUSEUP (entier)*
*SWFACTION_KEYDOWN (entier)*
*SWFACTION_KEYUP (entier)*

*SWFACTION_DATA (entier)*

## 6.21.7 Classes prédefinies

Ces classes sont définies par cette extension, et elles ne seront disponibles que si cette extension a été compilée avec PHP, ou bien chargée dynamiquement.

Ming introduit 13 objets en PHP. Pour les utilisez, vous devez être familier avec les objets .

*swfshape*
*swffill*
*swfgradient*
*swfbitmap*
*swftext*
*swftextfield*
*swffont*
*swfdisplayitem*
*swfmovie*
*swfbutton*
*swfaction*
*swfmorph*
*swfsprite*

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Classes prédefinies
- ming_setcubicthreshold
- ming_setscale
- ming_useswfversion
- swfaction
- swfbitmap−>getheight
- swfbitmap−>getwidth
- swfbitmap
- swfbutton_keypress
- swfbutton−>addaction
- swfbutton−>addshape
- swfbutton−>setaction
- swfbutton−>setdown
- swfbutton−>sethit
- swfbutton−>setover
- swfbutton−>setup
- swfbutton
- swfdisplayitem−>addcolor
- swfdisplayitem−>move
- swfdisplayitem−>moveto
- swfdisplayitem−>multcolor
- swfdisplayitem−>remove
- swfdisplayitem−>rotate
- swfdisplayitem−>rotateto

- swfdisplayitem−>scale
- swfdisplayitem−>scaleto
- swfdisplayitem−>setdepth
- swfdisplayitem−>setname
- swfdisplayitem−>setratio
- swfdisplayitem−>skewx
- swfdisplayitem−>skewxto
- swfdisplayitem−>skewy
- swfdisplayitem−>skewyto
- swfdisplayitem
- swffill−>moveto
- swffill−>rotateto
- swffill−>scaleto
- swffill−>skewxto
- swffill−>skewyto
- swffill
- swffont−>getwidth
- swffont
- swfgradient−>addentry
- swfgradient
- swfmorph−>getshape1
- swfmorph−>getshape2
- swfmorph
- swfmovie−>add
- swfmovie−>nextframe
- swfmovie−>output
- swfmovie−>remove
- swfmovie−>save
- swfmovie−>setbackground
- swfmovie−>setdimension
- swfmovie−>setframes
- swfmovie−>setrate
- swfmovie−>streammp3
- swfmovie
- swfshape−>addfill
- swfshape−>drawcurve
- swfshape−>drawcurveto
- swfshape−>drawline
- swfshape−>drawlineto
- swfshape−>movepen
- swfshape−>movepento
- swfshape−>setleftfill
- swfshape−>setline
- swfshape−>setrightfill
- swfshape
- swfsprite−>add
- swfsprite−>nextframe
- swfsprite−>remove
- swfsprite−>setframes
- swfsprite
- swftext−>addstring
- swftext−>getwidth
- swftext−>moveto
- swftext−>setcolor

- swftext−>setfont
- swftext−>setheight
- swftext−>setspacing
- swftext
- swftextfield−>addstring
- swftextfield−>align
- swftextfield−>setbounds
- swftextfield−>setcolor
- swftextfield−>setfont
- swftextfield−>setheight
- swftextfield−>setindentation
- swftextfield−>setleftmargin
- swftextfield−>setlinespacing
- swftextfield−>setmargins
- swftextfield−>setname
- swftextfield−>setrightmargin
- swftextfield

### 6.21.9 ming_setscale() : Configure l'échelle (??)

void **ming_setscale** ( int scale )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

### 6.21.10 ming_useswfversion() : Utilise la version de SWF (??)

void **ming_useswfversion** ( int version )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

### 6.21.11 swfaction() : Crée une nouvelle action

new **swfaction** ( string script )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfaction crée une nouvelle action, et compile le script script .

La syntaxe du script est basée sur le langage C, mais il utilise aussi beaucoup de notions propres à SWF : le bytecode SWF est trop simpliste pour faire l'essentiel de ce que l'on veut. Par exemple, il n'est pas possible de faire des fonctions sans descendre profondément dans les entrailles de la machine, car le bytecode de saut est écrit en dur. Pas moyen de pousser une adresse dans la pile, ou de dépiler − Chaque fonction doit savoir exactement où elle retourne.

Alors, que reste−t−il? Le compilateur reconnaît les mots suivants :

- break
- for
- continue
- if
- else
- do
- while

Il n'y a pas de type de données : toutes les valeurs de SWF sont stockées comme des chaînes de caractères. Les fonctions suivantes peuvent être utilisées dans les expressions :

*time()*
: Retourne le nombre de milli−secondes depuis le début de l'animation.

*random(seed)*
: Retourne un nombre pseudo−aléatoire, entre 0 et seed.

*length(expr)*
: Retourne la taille de l'expression donnée.

*int(number)*
: Retourne le nombre number, arrondi à l'entier inférieur le plus proche.

*concat(expr, expr)*
: Retourne la concaténation des deux expressions.

*ord(expr)*
: Retourne le code ASCII du caractère expr.

*chr(num)*
: Retourne le caractère pour le code ASCII num.

*substr(string, location, length)*
: Retourne la sous−chaîne, extraite de string, de longueur length et commençant au caractère location.

De plus, les commandes suivantes sont accessibles :

*duplicateClip(clip, name, depth)*
: Duplique le sprite nommé clip. La nouvelle animation a le nouveau nom name et la profondeur depth.

*removeClip(expr)*
: Supprime l'animation nommée expr.

*trace(expr)*
: Ecrit l'expression expr dans le fichier de d'historique. Il est peut probable que le navigateur ou le lecteur ne fasse quoi que ce soit avec.

*startDrag(target, lock, [left, top, right, bottom])*
: Comment à déplacer l'animation target. L'argument lock indique si le déplacement verrouille la souris (utilisez 0, FALSE ) ou 1 ( TRUE )). Les paramètres optionnels délimitent la zone de déplacement.

*stopDrag()*
: Cesse le déplacement de l'animation.

*callFrame(expr)*
: Appelle l'image expr comme une fonction.

*getURL(url, target, [method])*
: Charge l'URL url dans l'objet target. target peut être l'image courante (je pense) ou une des valeurs magiques "_level0" (remplace l'animation courante) ou "_level1" (charge une nouvelle animation à la place de la courante). L'argument optionnel method peut être post ou get, si vous voulez envoyer des variables au serveur.

*loadMovie(url, target)*
: La même chose que ci−dessus, plus ou moins. En fait, je ne sais pas trop quelle est

la différence.

*nextFrame()*
: Va à l'image suivante.

*prevFrame()*
: Va à l'image précédente.

*play()*
: Joue l'animation.

*stop()*
: Cesse de jouer l'animation.

*toggleQuality()*
: Passe de haute en basse qualité (et vice−versa).

*stopSounds()*
: Cesse de jouer les sons.

*gotoFrame(num)*
: Va à l'image numéro num. Les images sont numérotées à partir de 0.

*gotoFrame(name)*
: Va à l'image nommée name. Ce qui est carrément cool, car les labels ne sont pas encore supportés pour les images.

*setTarget(expr)*
: Modifie le contexte de l'action. C'est ce qu'ils disent, mais je n'ai pas trop d'idées là−dessus.

Et il y a un truc bizarre : l'expression frameLoaded(num) peut être utilise dans les conditions if et dans les boucles while pour vérifier si une image a été chargée. En tous cas, c'est ce qu'il est supposé faire, mais je ne l'ai jamais testé, et je doute sérieusement que cela fonctionne. Vous pouvez utiliser plutôt /:framesLoaded à la place.

Les sprites ont des propriétés. Vous pouvez les lire toutes (vraiment?), en modifier quelques unes. Les voici :

- x
- y
- xScale
- yScale
- currentFrame − (lecture seule)
- totalFrames − (lecture seule)
- alpha − Niveau de transparence
- visible − 1=on, 0=off (?)
- width − (lecture seule)
- height − (lecture seule)
- rotation
- target − (lecture seule) (???)
- framesLoaded − (lecture seule)
- name
- dropTarget − (lecture seule) (???)
- url − (lecture seule) (???)
- highQuality − 1=high, 0=low (?)
- focusRect − (???)
- soundBufTime − (???)

Modifier la position d'un sprite est aussi simple que /box.x = 100; . Pourquoi le slash initial? C'est comme cela que Flash garde la trace des sprites dans les animations, un peu comme des fichiers sous Unix. Si le sprite qui s'appelle box a lui−même un autre sprite appelé biff, vous pouvez y accéder avec la commande /box/biff.x = 100; . En tous cas, ça marche pour moi. Corrigez moi si

c'est faux!.

Cet exemple simple va déplacer le gros carré rouge dans la fenêtre.

Exemple avec swfaction

```
<?php
$s = new SWFShape();
 $f = $s->addFill(0xff, 0, 0);
 $s->setRightFill($f);
 $s->movePenTo(-500,-500);
 $s->drawLineTo(500,-500);
 $s->drawLineTo(500,500);
 $s->drawLineTo(-500,500);
 $s->drawLineTo(-500,-500);
 $p = new SWFSprite();
 $i = $p->add($s);
 $i->setDepth(1);
 $p->nextFrame();
 for($n=0; $n<5; ++$n)
 {
   $i->rotate(-15);
   $p->nextFrame();
 }
 $m = new SWFMovie();
 $m->setBackground(0xff, 0xff, 0xff);
 $m->setDimension(6000,4000);
 $i = $m->add($p);
 $i->setDepth(1);
 $i->moveTo(-500,2000);
 $i->setName("box");
 $m->add(new SWFAction("/box.x += 3;"));
 $m->nextFrame();
 $m->add(new SWFAction("gotoFrame(0); play();"));
 $m->nextFrame();
 header('Content-type: application/x-shockwave-flash');
 $m->output();
```

Cet exemple suit votre souris sur l'écran.

Exemple avec swfaction

```
<?php

 $m = new SWFMovie();
 $m->setRate(36.0);
 $m->setDimension(1200, 800);
 $m->setBackground(0, 0, 0);
 /* sprite de suivi de souris : vide, mais il suit la souris
 de manière à ce que nous connaissions ses coordonnées*/
 $i = $m->add(new SWFSprite());
 $i->setName('mouse');
 $m->add(new SWFAction("
   startDrag('/mouse', 1); /* '1' signifie verrouiller la souris */
 "));
 /* On peut tout simplement virer l'anti-aliasing, car il n'y a
 que des gros carrés, finalement */
 $m->add(new SWFAction("
   this.quality = 0;
 "));
 /* boîte de morphing */
 $r = new SWFMorph();
 $s = $r->getShape1();
 /* Notez que ce n'est pas pratique pour les formes habituelles.
 Aucune idée de pourquoi */
 $s->setLeftFill($s->addFill(0xff, 0xff, 0xff));
 $s->movePenTo(-40, -40);
 $s->drawLine(80, 0);
 $s->drawLine(0, 80);
 $s->drawLine(-80, 0);
 $s->drawLine(0, -80);
 $s = $r->getShape2();
 $s->setLeftFill($s->addFill(0x00, 0x00, 0x00));
 $s->movePenTo(-1, -1);
 $s->drawLine(2, 0);
 $s->drawLine(0, 2);
 $s->drawLine(-2, 0);
 $s->drawLine(0, -2);
 /* sprite contenant la boîte de morphing -
    c'est juste un scénario avec une boîte de morphing */
 $box = new SWFSprite();
 $box->add(new SWFAction("
   stop();
 "));
 $i = $box->add($r);
 for($n=0; $n<=20; ++$n)
 {
   $i->setRatio($n/20);
   $box->nextFrame();
```

```
  }
  /* ce conteneur nous permet d'utiliser la même action plusieurs fois */
  $cell = new SWFSprite();
  $i = $cell->add($box);
  $i->setName('box');
  $cell->add(new SWFAction("
    setTarget('box');
    /* ...x signifie abscisse du parent, c'est-à-dire (..).x */
    dx = (/mouse.x + random(6)-3 - ...x)/5;
    dy = (/mouse.y + random(6)-3 - ...y)/5;
    gotoFrame(int(dx*dx + dy*dy));
  "));
  $cell->nextFrame();
  $cell->add(new SWFAction("
    gotoFrame(0);
    play();
  "));
  $cell->nextFrame();
  /* finalement, ajoutons quelques cellules à l'animation */
  for($x=0; $x<12; ++$x)
  {
    for($y=0; $y8; ++$y)
    {
      $i = $m->add($cell);
      $i->moveTo(100*$x+50, 100*$y+50);
    }
  }
  $m->nextFrame();
  $m->add(new SWFAction("
    gotoFrame(1);
    play();
  "));
  header('Content-type: application/x-shockwave-flash');
  $m->output();
```

La même chose que ci-dessus, mais en couleurs.

Exemple avec swfaction

```
 <?php

  $m = new SWFMovie();
  $m->setDimension(11000, 8000);
  $m->setBackground(0x00, 0x00, 0x00);
  $m->add(new SWFAction("
this.quality = 0;
/frames.visible = 0;
startDrag('/mouse', 1);
  "));
  // sprite de suivi de souris
  $t = new SWFSprite();
  $i = $m->add($t);
  $i->setName('mouse');
  $g = new SWFGradient();
  $g->addEntry(0, 0xff, 0xff, 0xff, 0xff);
  $g->addEntry(0.1, 0xff, 0xff, 0xff, 0xff);
  $g->addEntry(0.5, 0xff, 0xff, 0xff, 0x5f);
  $g->addEntry(1.0, 0xff, 0xff, 0xff, 0);
  // gradient
  $s = new SWFShape();
  $f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
  $f->scaleTo(0.03);
  $s->setRightFill($f);
  $s->movePenTo(-600, -600);
  $s->drawLine(1200, 0);
  $s->drawLine(0, 1200);
  $s->drawLine(-1200, 0);
  $s->drawLine(0, -1200);
  // on en fait un sprite pour utiliser la fonction multColor()
  $p = new SWFSprite();
  $p->add($s);
  $p->nextFrame();
  // Ajoute la forme ici, chaque forme dans une couleur différente
  $q = new SWFSprite();
  $q->add(new SWFAction("gotoFrame(random(7)+1); stop();"));
  $i = $q->add($p);
  $i->multColor(1.0, 1.0, 1.0);
  $q->nextFrame();
  $i->multColor(1.0, 0.5, 0.5);
  $q->nextFrame();
  $i->multColor(1.0, 0.75, 0.5);
  $q->nextFrame();
  $i->multColor(1.0, 1.0, 0.5);
  $q->nextFrame();
  $i->multColor(0.5, 1.0, 0.5);
  $q->nextFrame();
  $i->multColor(0.5, 0.5, 1.0);
  $q->nextFrame();
  $i->multColor(1.0, 0.5, 1.0);
  $q->nextFrame();
  // Enfin, le code de l'action
```

```
  $p = new SWFSprite();
  $i = $p->add($q);
  $i->setName('frames');
  $p->add(new SWFAction("
dx = (/:mousex-/:lastx)/3 + random(10)-5;
dy = (/:mousey-/:lasty)/3;
x = /:mousex;
y = /:mousey;
alpha = 100;
  "));
  $p->nextFrame();
  $p->add(new SWFAction("
this.x = x;
this.y = y;
this.alpha = alpha;
x += dx;
y += dy;
dy += 3;
alpha -= 8;
  "));
  $p->nextFrame();
  $p->add(new SWFAction("prevFrame(); play();"));
  $p->nextFrame();
  $i = $m->add($p);
  $i->setName('frames');
  $m->nextFrame();
  $m->add(new SWFAction("
lastx = mousex;
lasty = mousey;
mousex = /mouse.x;
mousey = /mouse.y;
++num;
if(num == 11)
  num = 1;
removeClip('char' & num);
duplicateClip(/frames, 'char' & num, num);
  "));
  $m->nextFrame();
  $m->add(new SWFAction("prevFrame(); play();"));
  header('Content-type: application/x-shockwave-flash');
  $m->output();
```

Cet exemple simple gère le clavier (vous devrez cependant cliquer dans la fenêtre pour lui donner le focus, et vous devrez aussi laisser votre souris dans la fenêtre. Si vous savez comment faire cela automatiquement, dites-le moi!).

Exemple avec swfaction

```
 <?php
  /* Le sprite n'a qu'une lettre par image */
  $p = new SWFSprite();
  $p->add(new SWFAction("stop();"));
  $chars = "abcdefghijklmnopqrstuvwxyz".
       "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
           "1234567890!@#$%^&*()_+-=/[]{}|;:,.<>?`~";
  $f = new SWFFont("_sans");
  for($n=0; $nremove($i);
    $t = new SWFTextField();
    $t->setFont($f);
    $t->setHeight(240);
    $t->setBounds(600,240);
    $t->align(SWFTEXTFIELD_ALIGN_CENTER);
    $t->addString($c);
    $i = $p->add($t);
    $p->labelFrame($c);
    $p->nextFrame();
  }
  /* région de clic pour le bouton : toute la fenêtre */
  $s = new SWFShape();
  $s->setFillStyle0($s->addSolidFill(0, 0, 0, 0));
  $s->drawLine(600, 0);
  $s->drawLine(0, 400);
  $s->drawLine(-600, 0);
  $s->drawLine(0, -400);
  /* le bouton s'assure des touches pressées, et envoie le bon
     sprite à l'image de droite */
  $b = new SWFButton();
  $b->addShape($s, SWFBUTTON_HIT);
  for($n=0; $naddAction(new SWFAction("
setTarget('/char');
gotoFrame('$c');
    "), SWFBUTTON_KEYPRESS($c));
  }
  $m = new SWFMovie();
  $m->setDimension(600,400);
  $i = $m->add($p);
  $i->setName('char');
  $i->moveTo(0,80);
  $m->add($b);
```

```
header('Content-type: application/x-shockwave-flash');
$m->output();
```

## 6.21.12 swfbitmap->getheight() : Retourne la hauteur d'une bitmap

int **swfbitmap->getheight** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfbitmap->getheight retourne la hauteur d'une bitmap, en pixels.

Voir aussi swfbitmap->getwidth .

## 6.21.13 swfbitmap->getwidth() : Retourne la largeur d'une bitmap

int **swfbitmap->getwidth** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfbitmap->getwidth retourne la largeur d'une bitmap, en pixels.

Voir aussi swfbitmap->getheight .

## 6.21.14 swfbitmap() : Crée un objet bitmap

new **swfbitmap** ( string filename , *int alphafilename* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfbitmap crée un objet bitmap à partir d'un fichier JPEG ou DBL, nommé filename . alphafilename indique un fichier de masque à utiliser comme canal alpha sur une image JPEG.

| Note |
| --- |
| Seule les JPEG baseline (frame 0) sont supportés. Les baseline optimisée ou les JPEG progressives ne sont pas supportées. |

SWFBitmap a les méthodes suivantes : swfbitmap->getwidth et swfbitmap->getheight .

Il n'est pas possible d'importer directement des images PNG, il faut utiliser l'utilitaire de conversion png2dbl pour en faire un fichier .dbl ("define bits lossless"). La raison est que l'auteur ne souhaite pas de dépendance avec la librairie PNG. Le fichier d'autoconfiguration devrait régler ce problème, mais il n'est pas encore fait.

Importation de fichiers PNG sous Ming

```
<?php
  $s = new SWFShape();
  $f = $s->addFill(new SWFBitmap("png.dbl"));
  $s->setRightFill($f);
  $s->drawLine(32, 0);
  $s->drawLine(0, 32);
  $s->drawLine(-32, 0);
  $s->drawLine(0, -32);
  $m = new SWFMovie();
  $m->setDimension(32, 32);
  $m->add($s);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

Et vous pouvez ajouter un masque alpha sur une image JPEG.

Exemple avec swfbitmap

```
<?php
  $s = new SWFShape();
  //les fichiers .msk sont générés par l'utilitaire "gif2mask"
  $f = $s->addFill(new SWFBitmap("alphafill.jpg", "alphafill.msk"));
  $s->setRightFill($f);
  $s->drawLine(640, 0);
  $s->drawLine(0, 480);
  $s->drawLine(-640, 0);
  $s->drawLine(0, -480);
  $c = new SWFShape();
  $c->setRightFill($c->addFill(0x99, 0x99, 0x99));
  $c->drawLine(40, 0);
  $c->drawLine(0, 40);
  $c->drawLine(-40, 0);
  $c->drawLine(0, -40);
  $m = new SWFMovie();
  $m->setDimension(640, 480);
  $m->setBackground(0xcc, 0xcc, 0xcc);
  // décide un fond à damier
  for($y=0; $y<480; $y+=40)
  {
    for($x=0; $x<640; $x+=80)
    {
      $i = $m->add($c);
      $i->moveTo($x, $y);
    }
    $y+=40;
    for($x=40; $x<640; $x+=80)
    {
      $i = $m->add($c);
      $i->moveTo($x, $y);
    }
  }
  $m->add($s);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

## 6.21.15 swfbutton_keypress() : Retourne l'option d'actino pour keyPress (char)

int **swfbutton_keypress** ( string str )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.21.16 swfbutton->addaction() : Ajoute une action au bouton

void **swfbutton->addaction** ( resource action , int flags )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swfbutton−>setaction ajoute l'action action (créée par swfaction ) au bouton courant, dans les conditions précisées par flags . Les valeurs valides de flags sont : SWFBUTTON_MOUSEOVER, SWFBUTTON_MOUSEOUT, SWFBUTTON_MOUSEUP, SWFBUTTON_MOUSEUPOUTSIDE, SWFBUTTON_MOUSEDOWN, SWFBUTTON_DRAGOUT et SWFBUTTON_DRAGOVER

Voir aussi swfbutton−>addshape et swfaction .

## 6.21.17 swfbutton−>addshape() : Ajoute une forme à un bouton

void **swfbutton−>addshape** ( resource shape , int flags )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfbutton−>addshape ajoute la forme shape au bouton. Les valeurs possibles de flags sont : SWFBUTTON_UP, SWFBUTTON_OVER, SWFBUTTON_DOWN ou SWFBUTTON_HIT. SWFBUTTON_HIT n'est même pas affiché, elle désigne la région du clic d'un bouton. C'est−à−dire que tous points où le bouton est déssiné est considéré comme accessible.

## 6.21.18 swfbutton−>setaction() : Assigne l'action du bouton

void **swfbutton−>setaction** ( resource action )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfbutton−>setaction assigne l'action qui sera exécutée lorsque le bouton sera cliqué. C'est un alias de addAction(shape, SWFBUTTON_MOUSEUP) . action est une swfaction .

Voir aussi swfbutton−>addshape et swfaction .

## 6.21.19 swfbutton−>setdown() : Alias de addShape(shape, SWFBUTTON_DOWN)

void **swfbutton−>setdown** ( resource shape )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfbutton−>setdown est un alias pour addShape(shape, SWFBUTTON_DOWN) .

Voir aussi swfbutton−>addshape et swfaction .

## 6.21.20 swfbutton−>sethit() : Alias de addShape(shape, SWFBUTTON_HIT)

void **swfbutton−>sethit** ( resource shape )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfbutton−>sethit est un alias pour addShape(shape, SWFBUTTON_HIT) .

Voir aussi swfbutton−>addshape et swfaction .

## 6.21.21 swfbutton−>setover() : Alias de addShape(shape, SWFBUTTON_OVER)

void **swfbutton−>setover** ( resource shape )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfbutton−>setover est un alias pour addShape(shape, SWFBUTTON_OVER) .

Voir aussi swfbutton−>addshape et swfaction .

## 6.21.22 swfbutton−>setup() : Alias de SWFbutton−>addShape(shape, SWFBUTTON_UP)

void **swfbutton−>setup** ( resource shape )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfbutton−>setup est un alias pour SWFbutton−>addShape(shape, SWFBUTTON_UP) .

Voir aussi swfbutton−>addshape et swfaction .

## 6.21.23 swfbutton() : Crée un nouveau bouton

new **swfbutton** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfbutton crée un nouveau bouton. Cliquez−le, passez la souris dessus, et appelez des actions. Facile!

SWFButton a les méthodes suivantes : swfbutton−>addshape , swfbutton−>setup , swfbutton−>setover swfbutton−>setdown , swfbutton−>sethit swfbutton−>setaction et swfbutton−>addaction .

Cet exemple simplissime vous montre comme faire un roll−over, un roll−on, un clic, un relaché de souris, et rien du tout (pas d'action).

Exemple avec swfbutton

```
<?php
 $f = new SWFFont("_serif");
 $p = new SWFSprite();
 function label($string)
 {
   global $f;
   $t = new SWFTextField();
   $t->setFont($f);
   $t->addString($string);
   $t->setHeight(200);
   $t->setBounds(3200,200);
   return $t;
 }
 function addLabel($string)
 {
   global $p;
   $i = $p->add(label($string));
   $p->nextFrame();
   $p->remove($i);
 }
 $p->add(new SWFAction("stop();"));
 addLabel("NO ACTION");
 addLabel("SWFBUTTON_MOUSEUP");
 addLabel("SWFBUTTON_MOUSEDOWN");
 addLabel("SWFBUTTON_MOUSEOVER");
 addLabel("SWFBUTTON_MOUSEOUT");
 addLabel("SWFBUTTON_MOUSEUPOUTSIDE");
 addLabel("SWFBUTTON_DRAGOVER");
 addLabel("SWFBUTTON_DRAGOUT");
 function rect($r, $g, $b)
 {
   $s = new SWFShape();
   $s->setRightFill($s->addFill($r, $g, $b));
   $s->drawLine(600,0);
   $s->drawLine(0,600);
   $s->drawLine(-600,0);
   $s->drawLine(0,-600);
   return $s;
 }
 $b = new SWFButton();
 $b->addShape(rect(0xff, 0, 0), SWFBUTTON_UP | SWFBUTTON_HIT);
 $b->addShape(rect(0, 0xff, 0), SWFBUTTON_OVER);
 $b->addShape(rect(0, 0, 0xff), SWFBUTTON_DOWN);
 $b->addAction(new SWFAction("setTarget('/label'); gotoFrame(1);"),
           SWFBUTTON_MOUSEUP);
 $b->addAction(new SWFAction("setTarget('/label'); gotoFrame(2);"),
       SWFBUTTON_MOUSEDOWN);
 $b->addAction(new SWFAction("setTarget('/label'); gotoFrame(3);"),
       SWFBUTTON_MOUSEOVER);
 $b->addAction(new SWFAction("setTarget('/label'); gotoFrame(4);"),
       SWFBUTTON_MOUSEOUT);
 $b->addAction(new SWFAction("setTarget('/label'); gotoFrame(5);"),
       SWFBUTTON_MOUSEUPOUTSIDE);
 $b->addAction(new SWFAction("setTarget('/label'); gotoFrame(6);"),
       SWFBUTTON_DRAGOVER);
 $b->addAction(new SWFAction("setTarget('/label'); gotoFrame(7);"),
       SWFBUTTON_DRAGOUT);
 $m = new SWFMovie();
 $m->setDimension(4000,3000);
 $i = $m->add($p);
 $i->setName("label");
 $i->moveTo(400,1900);
 $i = $m->add($b);
 $i->moveTo(400,900);
 header('Content-type: application/x-shockwave-flash');
 $m->output();
?>
```

Cet exemple simple illustre le déplacement d'un gros bouton rouge dans la fenêtre. Ce n'est pas du tirer−déposer, mais juste du tirer.

Exemple avec swfbutton->addaction

```
<?php
  $s = new SWFShape();
  $s->setRightFill($s->addFill(0xff, 0, 0));
  $s->drawLine(1000,0);
  $s->drawLine(0,1000);
  $s->drawLine(-1000,0);
  $s->drawLine(0,-1000);
  $b = new SWFButton();
  $b->addShape($s, SWFBUTTON_HIT | SWFBUTTON_UP | SWFBUTTON_DOWN | SWFBUTTON_OVER);
  $b->addAction(new SWFAction("startDrag('/test', 0);"),
        SWFBUTTON_MOUSEDOWN);
// '0' signifie : ne pas verrouiller la souris
  $b->addAction(new SWFAction("stopDrag();"),
        SWFBUTTON_MOUSEUP | SWFBUTTON_MOUSEUPOUTSIDE);
  $p = new SWFSprite();
  $p->add($b);
  $p->nextFrame();
  $m = new SWFMovie();
  $i = $m->add($p);
  $i->setName('test');
  $i->moveTo(1000,1000);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

## 6.21.24 swfdisplayitem->addcolor() : Ajoute une couleur à une transoformation

void **swfdisplayitem->addcolor** ( *int red* , *int green* , *int blue* , *int a* )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swfdisplayitem->addcolor ajoute une couleur à la transformations courante. La couleur est donnée sous la forme RGB.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

## 6.21.25 swfdisplayitem->move() : Déplace un objet en coordonnées relatives

void **swfdisplayitem->move** ( int dx , int dy )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swfdisplayitem->move déplace la forme courante de dx et dy unités, depuis sa position courante.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi swfdisplayitem->moveto .

## 6.21.26 swfdisplayitem−>moveto() : Déplace un objet en coordonnées globales

void **swfdisplayitem−>moveto** ( int x , int y )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>moveto déplace la forme courante jusqu'au point de coordonnées globales ( x , y ).

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfdisplayitem−>move .

## 6.21.27 swfdisplayitem−>multcolor() : Multiplie la couleur de transformation

void **swfdisplayitem−>multcolor** ( *int red* , *int green* , *int blue* , *int a* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>multcolor multiplie la couleur de transformation par les valeurs données.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Cet exemple simple modifie l'atmosphère de votre image, et en fait une scène d'Halloween (utilisez un paysage ou une image claire pour un meilleur effet)

Exemple avec swfdisplayitem−>multcolor

```
<?php
 $b = new SWFBitmap("backyard.jpg");
 // Utilisez une de vos images
 $s = new SWFShape();
 $s->setRightFill($s->addFill($b));
 $s->drawLine($b->getWidth(), 0);
 $s->drawLine(0, $b->getHeight());
 $s->drawLine(-$b->getWidth(), 0);
 $s->drawLine(0, -$b->getHeight());
 $m = new SWFMovie();
 $m->setDimension($b->getWidth(), $b->getHeight());
 $i = $m->add($s);
 for($n=0; $n<=20; ++$n)
 {
   $i->multColor(1.0-$n/10, 1.0, 1.0);
   $i->addColor(0xff*$n/20, 0, 0);
   $m->nextFrame();
 }
 header('Content-type: application/x-shockwave-flash');
 $m->output();
?>
```

## 6.21.28 swfdisplayitem−>remove() : Supprime un objet d'une animation

void **swfdisplayitem−>remove** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>remove supprime cet objet de la liste d'affichage.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfmovie−>add .

## 6.21.29 swfdisplayitem−>rotate() : Fait tourner une forme relativement

void **swfdisplayitem−>rotate** ( float ddegrees )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>rotate fait tourner la forme de ddegrees degrés, en plus de sa rotation courante.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfdisplayitem−>rotateto .

## 6.21.30 swfdisplayitem−>rotateto() : Tourne un objet en angle absolu

void **swfdisplayitem−>rotateto** ( float degrees )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>rotateto tourne l'objet jusqu'à l'angle absolu degrees , en degrés.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Cet exemple amène trois chaînes tournoyantes depuis le fond de l'écran. Plutôt sympa.

Exemple avec swfdisplayitem−>rotateto

```
<?php
 $thetext =  "ming!";
 $f = new SWFFont("Bauhaus 93.fdb");
 $m = new SWFMovie();
 $m−>setRate(24.0);
```

```
  $m->setDimension(2400, 1600);
  $m->setBackground(0xff, 0xff, 0xff);
  // Les fonctions avec un nombre d'arguments sont vraiment une bonne idées.
  // Sincèrement!
  function text($r, $g, $b, $a, $rot, $x, $y, $scale, $string)
  {
    global $f, $m;
    $t = new SWFText();
    $t->setFont($f);
    $t->setColor($r, $g, $b, $a);
    $t->setHeight(960);
    $t->moveTo(-($f->getWidth($string))/2, $f->getAscent()/2);
    $t->addString($string);
    // On peut ajouter des propriétés comme pour une variable PHP standard
    // tant que les noms ne sont pas déjà pris.
    // e.g., vous ne pouvez pas utilisez $i->scale, car c'est une fonction.
    $i = $m->add($t);
    $i->x = $x;
    $i->y = $y;
    $i->rot = $rot;
    $i->s = $scale;
    $i->rotateTo($rot);
    $i->scale($scale, $scale);
    // mais les modification sont locales à une fonction, donc il faut
    // retourner l'objet modifié. Pas pratique...
    return $i;
 }
  function step($i)
  {
    $oldrot = $i->rot;
    $i->rot = 19*$i->rot/20;
    $i->x = (19*$i->x + 1200)/20;
    $i->y = (19*$i->y + 800)/20;
    $i->s = (19*$i->s + 1.0)/20;
    $i->rotateTo($i->rot);
    $i->scaleTo($i->s, $i->s);
    $i->moveTo($i->x, $i->y);
    return $i;
  }
  // Alors? &Ccedil;a valait la peine, non?
  $i1 = text(0xff, 0x33, 0x33, 0xff, 900, 1200, 800, 0.03, $thetext);
  $i2 = text(0x00, 0x33, 0xff, 0x7f, -560, 1200, 800, 0.04, $thetext);
  $i3 = text(0xff, 0xff, 0xff, 0x9f, 180, 1200, 800, 0.001, $thetext);
  for($i=1; $i<=100; ++$i)
  {
    $i1 = step($i1);
    $i2 = step($i2);
    $i3 = step($i3);
    $m->nextFrame();
  }
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

Voir aussi swfdisplayitem->rotate .

## 6.21.31 swfdisplayitem->scale() : Etire un objet relativement

void **swfdisplayitem->scale** ( int dx , int dy )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem->scale étire un objet de ( dx , dy ), à partir de sa taille courante.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie->add .

Voir aussi swfdisplayitem->scaleto .

## 6.21.32 swfdisplayitem−>scaleto() : Etire un objet en coordonnées globales

void **swfdisplayitem−>scaleto** ( int x , int y )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>scaleto étire un objet jusqu'au dimensions ( x , y ).

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfdisplayitem−>scale .

## 6.21.33 swfdisplayitem−>setdepth() : Modifie la place en profondeur (z−order)

void **swfdisplayitem−>setdepth** ( float depth )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>rotate place l'objet à la profondeur depth . Par défaut, l'objet est placé au niveau où il a été ajouté dans l'animation. Les objets les plus anciens sont placés tout en bas, et les nouveaux sont superposés.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

## 6.21.34 swfdisplayitem−>setname() : Nomme un objet

void **swfdisplayitem−>setname** ( string name )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>setname donne à l'objet courant le nom de name . Cela servira à repérer les acteurs d'un script d'action. Cela ne sert qu'avec les sprites.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

## 6.21.35 swfdisplayitem−>setratio() : Modifie le ratio de l'objet

void **swfdisplayitem−>setratio** ( float ratio )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfdisplayitem−>setratio modifie le ratio de l'objet, et le fixe à ratio . Uniquement utile pour les morphings.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Cet exemple simple effectue un morphing délicat de trois cercles concentriques.

Exemple swfdisplayitem−>setname

```
<?php
 $p = new SWFMorph();
 $g = new SWFGradient();
 $g->addEntry(0.0, 0, 0, 0);
 $g->addEntry(0.16, 0xff, 0xff, 0xff);
 $g->addEntry(0.32, 0, 0, 0);
 $g->addEntry(0.48, 0xff, 0xff, 0xff);
 $g->addEntry(0.64, 0, 0, 0);
 $g->addEntry(0.80, 0xff, 0xff, 0xff);
 $g->addEntry(1.00, 0, 0, 0);
 $s = $p->getShape1();
 $f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
 $f->scaleTo(0.05);
 $s->setLeftFill($f);
 $s->movePenTo(−160, −120);
 $s->drawLine(320, 0);
 $s->drawLine(0, 240);
 $s->drawLine(−320, 0);
 $s->drawLine(0, −240);
 $g = new SWFGradient();
 $g->addEntry(0.0, 0, 0, 0);
 $g->addEntry(0.16, 0xff, 0, 0);
 $g->addEntry(0.32, 0, 0, 0);
 $g->addEntry(0.48, 0, 0xff, 0);
 $g->addEntry(0.64, 0, 0, 0);
 $g->addEntry(0.80, 0, 0, 0xff);
 $g->addEntry(1.00, 0, 0, 0);
 $s = $p->getShape2();
 $f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
 $f->scaleTo(0.05);
 $f->skewXTo(1.0);
 $s->setLeftFill($f);
 $s->movePenTo(−160, −120);
 $s->drawLine(320, 0);
 $s->drawLine(0, 240);
 $s->drawLine(−320, 0);
 $s->drawLine(0, −240);
 $m = new SWFMovie();
 $m->setDimension(320, 240);
 $i = $m->add($p);
 $i->moveTo(160, 120);
 for($n=0; $n<=1.001; $n+=0.01)
 {
   $i->setRatio($n);
   $m->nextFrame();
 }
 header('Content−type: application/x−shockwave−flash');
 $m->output();
?>
```

## 6.21.36 swfdisplayitem−>skewx() : Incline suivant les X relativement

void **swfdisplayitem−>skewx** ( float ddegrees )

| Attention |
| --- |

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfdisplayitem−>skewx ajoute ddegrees à l'inclinaison courante (x−skew).

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfdisplayitem−>skewx , swfdisplayitem−>skewy and swfdisplayitem−>skewyto .

## 6.21.37 swfdisplayitem−>skewxto() : Incline suivant les X

void **swfdisplayitem−>skewxto** ( float degrees )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfdisplayitem−>skewxto modifie l'inclinaison (x−skew) à degrees . Si degrees vaut 1.0, l'angle sera de 45¡, en avant. S'il vaut plus, ce sera plus penché, et s'il vaut moins, ce sera plus droit.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfdisplayitem−>skewx , swfdisplayitem−>skewy and swfdisplayitem−>skewyto .

## 6.21.38 swfdisplayitem−>skewy() : Incline suivant les Y relativement

void **swfdisplayitem−>skewy** ( float ddegrees )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfdisplayitem−>skewy ajoute ddegrees à l'inclinaison courante (y−skew).

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfdisplayitem−>skewyto , swfdisplayitem−>skewx and swfdisplayitem−>skewxto .

## 6.21.39 swfdisplayitem−>skewyto() : Incline suivant les Y

void **swfdisplayitem−>skewyto** ( float degrees )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS

PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfdisplayitem−>skewyto modifie l'inclinaison (y−skew) à degrees . Si degrees vaut 1.0, l'angle sera de 45¡, en haut. S'il vaut plus, ce sera plus penché, et s'il vaut moins, ce sera plus droit.

L'objet peut être swfshape , swfbutton , swftext ou swfsprite . Il doit avoir été ajouté à une animation avec la fonction swfmovie−>add .

Voir aussi swfdisplayitem−>skewy , swfdisplayitem−>skewx and swfdisplayitem−>skewxto .

## 6.21.40 swfdisplayitem() : Crée un nouvel objet d'affichage displayitem

new **swfdisplayitem** ( void )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfdisplayitem crée un nouvel objet d'affichage displayitem.

C'est là que toute l'animation prend vie. Une fois que vous avez défini une forme, un texte, un sprite ou un bouton, vous l'ajoutez à une animation, puis vous utilisez la ressource retournée pour déplacer, étirer, contracter, faire tourner ou incliner la forme.

SWFDisplayItem a les méthodes suivantes : swfdisplayitem−>move , swfdisplayitem−>moveto , swfdisplayitem−>scaleto , swfdisplayitem−>scale , swfdisplayitem−>rotate , swfdisplayitem−>rotateto , swfdisplayitem−>skewxto , swfdisplayitem−>skewx , swfdisplayitem−>skewyto swfdisplayitem−>skewyto , swfdisplayitem−>setdepth swfdisplayitem−>remove , swfdisplayitem−>setname swfdisplayitem−>setratio , swfdisplayitem−>addcolor et swfdisplayitem−>multcolor .

## 6.21.41 swffill−>moveto() : Déplace l'origine de l'objet SWFFill

void **swffill−>moveto** ( int x , int y )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swffill−>moveto déplace l'origine de la forme jusqu'au point de coordonnées globales ( x , y ).

## 6.21.42 swffill−>rotateto() : Tourne la forme

void **swffill−>rotateto** ( float degrees )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swffill−>rotateto tourne la forme depuis sont orientation initiale jusqu'à un angle de degrees degrés.

## 6.21.43 swffill−>scaleto() : Modifie l'échelle de la forme

void **swffill−>scaleto** ( int x , int y )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swffill−>scaleto modifie l'échelle de la forme de x dans le sens des abscisses et y dans le sens des ordonnées.

## 6.21.44 swffill−>skewxto() : Incline (abscisses)

void **swffill−>skewxto** ( float x )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swffill−>skewxto incline la forme de x suivant l'axe des abscisses. Si x vaut 1.0, l'inclinaison sera de 45¡ degrés, en avant. Si x vaut plus, l'inclinaison sera plus forte, et sinon, la forme sera plus droite.

## 6.21.45 swffill−>skewyto() : Incline (ordonnées)

void **swffill−>skewyto** ( float y )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swffill−>skewyto incline la forme de y suivant l'axe des abscisses. Si y vaut 1.0, l'inclinaison sera de 45¡ degrés, en avant. Si x vaut plus, l'inclinaison sera plus forte, et sinon, la forme sera plus droite.

## 6.21.46 swffill() : Crée un objet de remplissage

new **swffill** ( )

swffill vous permet de transformer une image bitmap ou un gradient. Les objets swffill sont créé par swfshape−>addfill .

SWFFill a les méthodes suivantes : swffill−>moveto , swffill−>scaleto , swffill−>rotateto , swffill−>skewxto et swffill−>skewyto .

## 6.21.47 swffont−>getwidth() : Retourne la taille de la chaîne

int **swffont−>getwidth** ( string string )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swffont−>getwidth retourne la taille de la chaîne string , avec la police courante. Vous utiliserez plutôt la même méthode de l'objet swftext , qui utilise les paramètres de l'objet.

## 6.21.48 swffont() : Charge une police

new **swffont** ( string filename )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

si filename est le nom d'un fichier FDB (i.e., si le nom de fichier se termine par ".fdb"), charge la police.

FDB ("font definition block") est un petit utilitaire pour Flash DefineFont2 qui contient une description complète de la police. Vous pouvez créer des fichiers FDB à partir du "SWT Generator", qui est inclus avec les utilitaires makefdb − regardez dans le dossier utilitaire de Ming.

Les polices utilisateurs ne contiennent aucune information autre que le nom de la police. On suppose que la police sera elle−même accessible au lecteur. Les polices "_serif", "_sans", et "_typewriter" doivent être universellement disponibles. Par exemple :

```
<?php
  $f = newSWFFont("_sans");
?>
```

vous donne la police standard "sans−serif", probablement identique à celle que vous obtenez avec le code <font name="sans−serif"> .

swffont retourne une ressource de police, à utiliser avec les méthodes swftext−>setfont et swftextfield−>setfont .

SWFFont a les méthodes suivantes : swffont−>getwidth .

## 6.21.49 swfgradient−>addentry() : Ajoute une couleur à la liste du gradient

void **swfgradient−>addentry** ( float ratio , int red , int green , int blue , *int a* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfgradient−>addentry ajoute une couleur à la liste des couleurs du gradient. ratio est un nombre de 0 à 1, qui indique l'ordre d'apparition des couleurs. Vous devez ajouter les couleurs dans l'ordre croissant de ratio.

red , green , blue représente une couleur, au format RGB. Le dernier paramètre a est optionnel.

## 6.21.50 swfgradient() : Crée un objet gradient

new **swfgradient** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfgradient crée un nouvel objet gradient.

Une fois que vous avez ajouté les couleurs à votre gradient, vous pouvez l'utiliser dans des formes, avec la fonction swfshape−>addfill .

SWFGradient a la méthode suivante : swfgradient−>addentry .

Cet exemple simple affiche un gradient noir−blanc comme fond, et un gradient concentrique au centre.

Exemple avec swfgradient

```
<?php
 $m = new SWFMovie();
 $m->setDimension(320, 240);
 $s = new SWFShape();
 // gradient noir-blanc
 $g = new SWFGradient();
 $g->addEntry(0.0, 0, 0, 0);
 $g->addEntry(1.0, 0xff, 0xff, 0xff);
 $f = $s->addFill($g, SWFFILL_LINEAR_GRADIENT);
 $f->scaleTo(0.01);
 $f->moveTo(160, 120);
 $s->setRightFill($f);
 $s->drawLine(320, 0);
 $s->drawLine(0, 240);
 $s->drawLine(-320, 0);
 $s->drawLine(0, -240);
 $m->add($s);
 $s = new SWFShape();
 // gradient radial : rouge vers transparent
 $g = new SWFGradient();
 $g->addEntry(0.0, 0xff, 0, 0, 0xff);
 $g->addEntry(1.0, 0xff, 0, 0, 0);
 $f = $s->addFill($g, SWFFILL_RADIAL_GRADIENT);
 $f->scaleTo(0.005);
 $f->moveTo(160, 120);
 $s->setRightFill($f);
 $s->drawLine(320, 0);
 $s->drawLine(0, 240);
 $s->drawLine(-320, 0);
 $s->drawLine(0, -240);
 $m->add($s);
 header('Content-type: application/x-shockwave-flash');
 $m->output();
?>
```

## 6.21.51 swfmorph−>getshape1() : Sélectionne la forme de départ

mixed **swfmorph−>getshape1** ( void )

| Attention |
| --- |

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfmorph−>getshape1 sélectionne la forme de début de morphing. swfmorph−>getshape1 retourne un objet swfshape .

## 6.21.52 swfmorph−>getshape2() : Sélectionne la forme de fin

mixed **swfmorph−>getshape2** ( void )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfmorph−>getshape2 sélectionne la forme de début de morphing. swfmorph−>getshape2 retourne un objet swfshape .

## 6.21.53 swfmorph() : Crée un morphing

new **swfmorph** ( void )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfmorph crée un morphing.

swfmorph s'appelle aussi "shape tween". C'est cet objet qui permet toutes ces superbes animations qui mettent à genou votre ordinateur. Joie!

Les méthodes ici sont plutôt bizarres. Il serait tellement plus logique d'avoir seulement new SWFMorph(shape1, shape2); , mais, telles que sont les choses aujourd'hui, la deuxième forme a besoin de savoir qu'elle est l'aboutissement d'un morphing. (Tout cela, parceque Flash commence à dessiner aussitôt qu'il a les commandes de dessins. S'il conservait les descriptions de ses propres formes, et attendait leur totalité avant d'écrire, ceci et bien d'autres choses serait tellement plus simple).

SWFMorph a les méthodes suivantes : swfmorph−>getshape1 et swfmorph−>getshape2 .

Cet exemple simple effectue le morphing d'une gros carré rouge en un carré plus petit, bleu et bordé de noir.

Exemple avec swfmorph

```
<?php
 $p = new SWFMorph();
 $s = $p->getShape1();
 $s->setLine(0,0,0,0);
 /* Notez que cela se fait dans l'ordre inverse de l'ordre habituel
    (gauche au lieu de droite), mais je n'ai aucune idée de pourquoi... */
 $s->setLeftFill($s->addFill(0xff, 0, 0));
 $s->movePenTo(-1000,-1000);
 $s->drawLine(2000,0);
 $s->drawLine(0,2000);
```

```
  $s->drawLine(-2000,0);
  $s->drawLine(0,-2000);
  $s = $p->getShape2();
  $s->setLine(60,0,0,0);
  $s->setLeftFill($s->addFill(0, 0, 0xff));
  $s->movePenTo(0,-1000);
  $s->drawLine(1000,1000);
  $s->drawLine(-1000,1000);
  $s->drawLine(-1000,-1000);
  $s->drawLine(1000,-1000);
  $m = new SWFMovie();
  $m->setDimension(3000,2000);
  $m->setBackground(0xff, 0xff, 0xff);
  $i = $m->add($p);
  $i->moveTo(1500,1000);
  for($r=0.0; $r<=1.0; $r+=0.1)
  {
    $i->setRatio($r);
    $m->nextFrame();
  }
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

## 6.21.54 swfmovie->add() : Ajoute un objet dans une animation

void **swfmovie->add** ( resource instance )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfmovie->add ajoute l'objet instance dans l'animation courante. instance peut être de n'importe quel type : forme (shape), texte (text), police (font), etc... Ils doivent être ajouté à une animation pour être utilisé.

Pour les objets affichables (formes, textes, boutons, sprites), swfmovie->add retourne un objet swfdisplayitem de la liste d'affichage. Ainsi, vous pouvez ajouter la même forme plusieurs fois dans la même animation, et obtenir des ressources différentes pour chaque instance.

Voir aussi tous les autres objets et swfmovie->remove

Des exemples d'utilisation dans : swfdisplayitem->rotateto et swfshape->addfill .

## 6.21.55 swfmovie->nextframe() : Passe à l'image suivante

void **swfmovie->nextframe** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfmovie->setframes passe à l'image suivante de l'animation.

## 6.21.56 swfmovie->output() : Envoie votre animation au navigateur

void **swfmovie->output** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfmovie−>output envoie votre animation au navigateur. En PHP, faîte le précéder de la fonction header .

```
 <?php
  header('Content−type: application/x−shockwave−flash');
?>
```

Cela indique au navigateur que l'animation qui arrive est en Flash.

Voir aussi swfmovie−>save .

Des exemples d'utilisation dans : swfmovie−>streammp3 , swfdisplayitem−>rotateto , swfaction ... En fait, tous les exemples utilisent cette méthode.

## 6.21.57 swfmovie−>remove() : Supprime un objet d'une animation

void **swfmovie−>remove** ( resource instance )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfmovie−>remove supprime l'objet instance de la liste d'affichage, pour l'animation courante. L'objet ne sera plus disponible pour être affiché ou utilisé.

Voir aussi swfmovie−>add .

## 6.21.58 swfmovie−>save() : Sauve dans un fichier

void **swfmovie−>save** ( string filename )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfmovie−>save sauve votre animation dans le fichier filename .

Voir aussi swfmovie−>output .

## 6.21.59 swfmovie−>setbackground() : Modifie la couleur de fond

void **swfmovie−>setbackground** ( int red , int green , int blue )

| Attention |
| --- |
| |

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfmovie−>setbackground modifie la couleur de fond. Pourquoi est−ce que cette fonction n'accepte pas de canal alpha? (réflechissez quelques instants :−) ). En fait, cela ne serait pas si stupide : vous pouvez laisser apercevoir le fond HTML à travers l'animation. Il y a un moyen de faire cela, mais cela ne fonctionne qu'avec IE 4. Recherchez sur le site de http://www.macromedia.com/ pour plus de détails.

## 6.21.60 swfmovie−>setdimension() : Modifie les dimensions de l'animation

void **swfmovie−>setdimension** ( int width , int height )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfmovie−>setdimension modifie les dimensions de l'animation : width est la largeur et height la hauteur.

## 6.21.61 swfmovie−>setframes() : Modifie le nombre total d'images dans l'animation

void **swfmovie−>setframes** ( string numberofframes )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfmovie−>setframes modifie le nombre total d'images dans l'animation, et le fixe à numberofframes .

## 6.21.62 swfmovie−>setrate() : Modifie la vitesse de l'animation

void **swfmovie−>setrate** ( int rate )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfmovie−>setrate fixe la vitesse de l'animation à rate images par secondes. L'animation ralentira d'elle−même si le lecteur Flash ne peut pas afficher suffisamment rapidement, à moins qu'il n'y ait du son en stream, auquel cas les images sont sacrifiées pour garder un son fluide.

## 6.21.63 swfmovie−>streammp3() : Envoie un fichier MP3 en streaming

void **swfmovie−>streammp3** ( string mp3filename )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfmovie−>streammp3 envoie le fichier MP3 mp3filename en stream audio. swfmovie−>streammp3 n'est pas très robuste, et se prend facilement les pieds dans le tapis (elle peut éviter la balise initiale ID3, mais c'est bien tout). Tout comme swfshape−>addjpegfill(), ce n'est pas une fonction stable. Il faudra sûrement faire un objet séparé, pour gérer les types de son.

Notez que l'animation n'est pas suffisamment intelligente pour ajouter un nombre suffisant d'images, afin de correspondre à la durée totale du stream MP3. Il vous faudra ajouter des images jusqu'à durée de la musique multiplié par le nombre d'images par secondes.

Oui, vous pouvez utiliser Ming pour mettre un rock−'n−roll endiablé dans vos animation. Evitez d'en parler à la RIAA ou la SACEM.

Exemple avec swfmovie−>streammp3

```
<?php
  $m = new SWFMovie();
  $m->setRate(12.0);
  $m->streamMp3("distortobass.mp3");
// utilisez vos propres MP3
// assurez-vous d'avoir les droits
// 11.85 secondes avec 12.0 images par seconde = 142 frames
  $m->setFrames(142);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

## 6.21.64 swfmovie() : Crée un objet 'animation'

new **swfmovie** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfmovie Crée un objet 'animation', représentant une animation Flash version 4.

SWFMovie a les méthodes suivantes : swfmovie−>output , swfmovie−>save , swfmovie−>add , swfmovie−>remove , swfmovie−>nextframe , swfmovie−>setbackground , swfmovie−>setrate , swfmovie−>setdimension , swfmovie−>setframes et swfmovie−>streammp3 .

Des exemples d'utilisation dans : swfdisplayitem−>rotateto , swfshape−>setline , swfshape−>addfill ... En fait, tous les exemples utilisent cet objet.

## 6.21.65 swfshape−>addfill() : Ajoute un remplissage plein à la forme Flash

void **swfshape−>addfill** ( int red , int green , int blue , *int a* )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

void **swfshape−>addfill** ( SWFbitmap bitmap , *int* *flags* )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

void **swfshape−>addfill** ( SWFGradient gradient , *int* *flags* )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

swfshape−>addfill ajoute un remplissage plein à la forme. swfshape−>addfill accepte trois différents types d'arguments.

red , green , blue est une couleur (format RGB). Le dernier paramètre a est optionnel.

L'argument bitmap est un objet swfbitmap . Le paramètre flags peut être l'un des suivants : SWFFILL_CLIPPED_BITMAP ou SWFFILL_TILED_BITMAP. Par défault, c'est SWFFILL_TILED_BITMAP. Je crois.

L'argument gradient est un objet swfgradient . L'argument flags peut alors prendre l'une des valeurs suivantes : SWFFILL_RADIAL_GRADIENT ou SWFFILL_LINEAR_GRADIENT. Par défaut, c'est SWFFILL_LINEAR_GRADIENT. Cette fois ci, j'en suis sûr.

swfshape−>addfill retourne un objet swffill à utiliser avec swfshape−>setleftfill , et swfshape−>setrightfill décrite un peu plus loin.

Voir aussi swfshape−>setleftfill et swfshape−>setrightfill .

Ceci est un exemple simple qui affiche un cadre sur une bitmap. Ah, il y a un petit bug dans le lecteur Flash : il ne semble pas faire grand cas de la transformation de la seconde forme en morphing. Suivant les specs, la bitmap devrait s'étirer avec la forme dans cet exemple...

Exemple avec swfshape−>addfill

```
<?php
 $p = new SWFMorph();
 $b = new SWFBitmap("alphafill.jpg");
 // utilisez vos propres bitmaps!
 $width = $b->getWidth();
 $height = $b->getHeight();
 $s = $p->getShape1();
 $f = $s->addFill($b, SWFFILL_TILED_BITMAP);
 $f->moveTo(-$width/2, -$height/4);
 $f->scaleTo(1.0, 0.5);
 $s->setLeftFill($f);
 $s->movePenTo(-$width/2, -$height/4);
 $s->drawLine($width, 0);
 $s->drawLine(0, $height/2);
 $s->drawLine(-$width, 0);
 $s->drawLine(0, -$height/2);
 $s = $p->getShape2();
 $f = $s->addFill($b, SWFFILL_TILED_BITMAP);
 // ces déplacements n'ont aucun effet
 $f->moveTo(-$width/4, -$height/2);
 $f->scaleTo(0.5, 1.0);
 $s->setLeftFill($f);
 $s->movePenTo(-$width/4, -$height/2);
 $s->drawLine($width/2, 0);
 $s->drawLine(0, $height);
 $s->drawLine(-$width/2, 0);
 $s->drawLine(0, -$height);
```

```
  $m = new SWFMovie();
  $m->setDimension($width, $height);
  $i = $m->add($p);
  $i->moveTo($width/2, $height/2);
  for($n=0; $n<1.001; $n+=0.03)
  {
    $i->setRatio($n);
    $m->nextFrame();
  }
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

## 6.21.66 swfshape->drawcurve() : Dessine une courbe relativement

void **swfshape->drawcurve** ( int controldx , int controldy , int anchordx , int anchordy )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape->drawcurve dessine une courbe quadratique (avec le style courant de ligne, modifié par swfshape->setline ) depuis le point courant jusqu'au point ( anchordx , anchordy ) relativement au point courant, et en utilisant le point de contrôle ( controldx , controldy ). C'est-à-dire qu'il commence en allant vers le point de contrôle, puis se dirigie sur le point d'ancrrage.

Voir aussi swfshape->drawlineto , swfshape->drawline , swfshape->movepento et swfshape->movepen .

## 6.21.67 swfshape->drawcurveto() : Dessine une courbe

void **swfshape->drawcurveto** ( int controlx , int controly , int anchorx , int anchory )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape->drawcurveto dessine une courbe quadratique (avec le style courant de ligne, modifié par swfshape->setline ) depuis le point courant jusqu'au point ( anchorx , anchory ) en utilisant ( controlx , controly ) comme point de contrôle. C'est-à-dire qu'il commence en allant vers le point de contrôle, puis se dirigie sur le point d'ancrrage.

Voir aussi swfshape->drawlineto , swfshape->drawline , swfshape->movepento et swfshape->movepen .

## 6.21.68 swfshape->drawline() : Dessine une ligne relativement

void **swfshape->drawline** ( int dx , int dy )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape−>drawline dessine une ligne (avec le style courant de ligne, modifié par swfshape−>setline ) depuis le point courant, et sur le déplacement de ( dx , dy ).

Voir aussi swfshape−>movepento , swfshape−>drawcurveto , swfshape−>movepen et swfshape−>drawlineto .

## 6.21.69 swfshape−>drawlineto() : Dessine une ligne

void **swfshape−>drawlineto** ( int x , int y )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape−>drawlineto dessine une ligne (avec le style courant de ligne, modifié par swfshape−>setline ) depuis le point courant jusqu'au point ( x , y ) dans l'espace de coordonnées de la forme.

Voir aussi swfshape−>movepento , swfshape−>drawcurveto , swfshape−>movepen et swfshape−>drawline .

## 6.21.70 swfshape−>movepen() : Déplace le stylo relativement

void **swfshape−>movepen** ( int dx , int dy )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape−>movepen déplace le stylo dans la forme depuis les coordonnées (current x,current y) jusqu'au coordonnées (current x + dx , current y + dy ), dans l'espace de coordonnées de la forme.

Voir aussi swfshape−>movepento , swfshape−>drawcurveto , swfshape−>drawlineto et swfshape−>drawline .

## 6.21.71 swfshape−>movepento() : Déplace le stylo

void **swfshape−>movepento** ( int x , int y )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape−>movepento déplace le stylo dans la forme jusqu'au coordonnées globales ( x , y ).

Voir aussi swfshape−>movepen , swfshape−>drawcurveto , swfshape−>drawlineto et swfshape−>drawline .

## 6.21.72 swfshape−>setleftfill() : Modifie la couleur de rastérisation de gauche

void **swfshape−>setleftfill** ( swfgradient fill )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

void **swfshape−>setleftfill** ( int red , int green , int blue , *int a* )

Tout ce sac de noeud fait qu'il y a deux couleurs de remplissage des lignes. Lorsque l'objet est rasterisé, il est pratique de savoir à l'avance quelle sont les remplissages, et le format SWF les demande.

swfshape−>setleftfill affecte à la couleur de rastérisation de gauche, c'est−à−dire l'intérieur d'un objet, si vous définissez les contours d'un objet dans le sens inverse des aiguilles d'une montre. L'objet de remplissage est un objet swffill , retourné par la fonction swfshape−>addfill ci−dessus.

Cela semble être le contraire lorsque vous défnissez une forme dans un morphing. Si votre navigateur crashe, essayez de placer le remplissage sur l'autre côté.

Raccourci pour swfshape−>setleftfill($s−>addfill($r, $g, $b [, $a])); .

Voir aussi swfshape−>setrightfill .

## 6.21.73 swfshape−>setline() : Modifie le style de ligne de la forme

void **swfshape−>setline** ( int width , *int red* , *int green* , *int blue* , *int a* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape−>setline modifie le style de ligne de la forme. width est la largeur de la ligne. Si width vaut 0, le style est supprimé (et tous les autres arguments sont ignorés). Si width > 0, alors la couleur de la ligne devient ( red , green , blue ). Les couleurs sont représentées en RGB. Le dernier paramètre a est optionnel.

swfshape−>setline accepte 1, 4 ou 5 arguments (mais jamais 3 ou 2).

Vous devez déclarer un style avant de l'utiliser (voir exemple).

Cet exemple enfantin dessine une chaîne "!#%*@", dans des couleurs marrantes et un style rigolo.

Exemple swfshape−>setline

```
<?php
 $s = new SWFShape();
 $f1 = $s->addFill(0xff, 0, 0);
 $f2 = $s->addFill(0xff, 0x7f, 0);
 $f3 = $s->addFill(0xff, 0xff, 0);
 $f4 = $s->addFill(0, 0xff, 0);
 $f5 = $s->addFill(0, 0, 0xff);
 // erreur : il faut déclarer tous les styles avant
 // de les utiliser.
```

```
  $s->setLine(40, 0x7f, 0, 0);
  $s->setLine(40, 0x7f, 0x3f, 0);
  $s->setLine(40, 0x7f, 0x7f, 0);
  $s->setLine(40, 0, 0x7f, 0);
  $s->setLine(40, 0, 0, 0x7f);
  $f = new SWFFont('Techno.fdb');
  $s->setRightFill($f1);
  $s->setLine(40, 0x7f, 0, 0);
  $s->drawGlyph($f, '!');
  $s->movePen($f->getWidth('!'), 0);
  $s->setRightFill($f2);
  $s->setLine(40, 0x7f, 0x3f, 0);
  $s->drawGlyph($f, '#');
  $s->movePen($f->getWidth('#'), 0);
  $s->setRightFill($f3);
  $s->setLine(40, 0x7f, 0x7f, 0);
  $s->drawGlyph($f, '%');
  $s->movePen($f->getWidth('%'), 0);
  $s->setRightFill($f4);
  $s->setLine(40, 0, 0x7f, 0);
  $s->drawGlyph($f, '*');
  $s->movePen($f->getWidth('*'), 0);
  $s->setRightFill($f5);
  $s->setLine(40, 0, 0, 0x7f);
  $s->drawGlyph($f, '@');
  $m = new SWFMovie();
  $m->setDimension(3000,2000);
  $m->setRate(12.0);
  $i = $m->add($s);

// note la chaine est ici!!!
  $i->moveTo(1500-$f->getWidth("!#%*@")/2, 1000+$f->getAscent()/2);
  header('Content-type: application/x-shockwave-flash');
  $m->output();
?>
```

## 6.21.74 swfshape->setrightfill() : Modifie la couleur de rastériation de droite

void **swfshape->setrightfill** ( swfgradient fill )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

void **swfshape->setrightfill** ( int red , int green , int blue , *int* *a* )

Voir aussi swfshape->setleftfill .

Raccourci pour swfshape->setrightfill($s->addfill($r, $g, $b [, $a])); .

## 6.21.75 swfshape() : Crée une nouvelle forme

new **swfshape** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls. |

swfshape crée une nouvelle forme.

SWFShape a les méthodes suivantes : swfshape->setline , swfshape->addfill , swfshape->setleftfill , swfshape->setrightfill , swfshape->movepento , swfshape->movepen , swfshape->drawlineto , swfshape->drawline , swfshape->drawcurveto et swfshape->drawcurve .

Ce exemple simple dessine un quadrant d'ellipse rouge.

Exemple avec swfshape

```
<?php
 $s = new SWFShape();
 $s->setLine(40, 0x7f, 0, 0);
 $s->setRightFill($s->addFill(0xff, 0, 0));
 $s->movePenTo(200, 200);
 $s->drawLineTo(6200, 200);
 $s->drawLineTo(6200, 4600);
 $s->drawCurveTo(200, 4600, 200, 200);
 $m = new SWFMovie();
 $m->setDimension(6400, 4800);
 $m->setRate(12.0);
 $m->add($s);
 $m->nextFrame();
 header('Content-type: application/x-shockwave-flash');
 $m->output();
?>
```

## 6.21.76 swfsprite->add() : Ajoute un objet à un sprite

void **swfsprite->add** ( resource object )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swfsprite->add ajoute une swfshape , un swfbutton , un swftext , une swfaction ou une autre animation swfsprite .

Pour les objets affichables ( swfshape , swfbutton , swftext , swfaction or swfsprite ), cela retourne une ressource sur l'objet dans la liste d'affichage.

## 6.21.77 swfsprite->nextframe() : Va à la prochaine image du sprite

void **swfsprite->nextframe** ( void )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swfsprite->setframes se déplace à la prochaine image du sprite.

## 6.21.78 swfsprite->remove() : Supprime un objet dans un sprite

void **swfsprite->remove** ( resource object )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swfsprite->remove supprime une swfshape , un swfbutton , un swftext , une swfaction ou un swfsprite du sprite courant.

## 6.21.79 swfsprite−>setframes() : Fixe le nombre maximum d'image dans le sprite

void **swfsprite−>setframes** ( int numberofframes )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfsprite−>setframes fixe le nombre total d'images de l'animation à numberofframes .

## 6.21.80 swfsprite() : Crée un sprite

new **swfsprite** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swfsprite sont aussi connue sous le nom de "clip" : ils permettent la création d'objet animé dans une animation, avec un scénario propre. De ce fait, un sprite a les mêmes méthodes qu'une animation.

swfsprite a les méthodes suivantes : swfsprite−>add , swfsprite−>remove , swfsprite−>nextframe et swfsprite−>setframes .

Ce exemple pratique fait tourner un superbe carré rouge.

Exemple de swfsprite

```php
<?php
 $s = new SWFShape();
 $s->setRightFill($s->addFill(0xff, 0, 0));
 $s->movePenTo(-500,-500);
 $s->drawLineTo(500,-500);
 $s->drawLineTo(500,500);
 $s->drawLineTo(-500,500);
 $s->drawLineTo(-500,-500);
 $p = new SWFSprite();
 $i = $p->add($s);
 $p->nextFrame();
 $i->rotate(15);
 $p->nextFrame();
 $i->rotate(15);
 $p->nextFrame();
 $i->rotate(15);
 $p->nextFrame();
 $i->rotate(15);
 $p->nextFrame();
 $i->rotate(15);
 $p->nextFrame();
 $m = new SWFMovie();
 $i = $m->add($p);
 $i->moveTo(1500,1000);
 $i->setName("blah");
 $m->setBackground(0xff, 0xff, 0xff);
 $m->setDimension(3000,2000);
 header('Content-type: application/x-shockwave-flash');
 $m->output();
?>
```

## 6.21.81 swftext−>addstring() : Ajoute du texte

void **swftext−>addstring** ( string string )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext−>addstring ajoute le texte string au texte courant, et le dessine. Le stylo est situé sur la ligne de base du texte, c'est−à−dire que le texte sera écrit horizontalement.

## 6.21.82 swftext−>getwidth() : Calcule la longueur d'une chaîne

void **swftext−>getwidth** ( string string )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext−>addstring retourne la taille de la chaîne string , une fois qu'elle est dessinée avec la police et l'espaceement courant.

## 6.21.83 swftext−>moveto() : Déplace le stylo de texte

void **swftext−>moveto** ( int x , int y )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext−>moveto déplace le style (ou le curseur, si ça a un sens) jusqu'au coordonnées ( x , y ) dans l'espace de coordonnées du texte. Si x ou y vaut 0, la valeur de coordonnées de la dimension reste la même. C'est ennuyeux, et cela devrait être corrigé.

## 6.21.84 swftext−>setcolor() : Modifie la couleur de la police

void **swftext−>setcolor** ( int red , int green , int blue , *int a* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext−>setspacing change la couleur de la police courante. Par défaut, c'est noir. La couleur est représentée avec la convention RGB.

## 6.21.85 swftext−>setfont() : Sélectionne la police courante

void **swftext−>setfont** ( string font )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext−>setfont remplace la police courante par font .

## 6.21.86 swftext−>setheight() : Modifie la hauteur de la police courante

void **swftext−>setheight** ( int height )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext−>setheight fixe la hauteur courante de la police courante à height . Par défaut, c'est 240.

## 6.21.87 swftext−>setspacing() : Modifie l'espacement de police

void **swftext−>setspacing** ( float spacing )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext−>setspacing fixe l'espacement de police à spacing spacing . Par défaut, c'est 1.0. 0 signifie que toutes les lettres seront écrites au même point. Cela fonctionne pas terrible, car l'avance des lettres augmente, et l'espacement entre lettre n'est pas toujours le même. Il faudra que je l'explique plus clairement. Ou bien que je corrige les erreurs.

## 6.21.88 swftext() : Crée un nouvel objet texte

new **swftext** ( void )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftext crée un nouvel objet texte, prêt à être manipulé.

SWFText a les méthodes suivantes : swftext−>setfont , swftext−>setheight , swftext−>setspacing , swftext−>setcolor , swftext−>moveto , swftext−>addstring et swftext−>getwidth .

Cet exemple simple va afficher la phrase "PHP fait du Flash avec Ming" sur un fond blanc.

Exemple avec swftext

```
<?php
 $f = new SWFFont("Techno.fdb");
 $t = new SWFText();
 $t->setFont($f);
 $t->moveTo(200, 2400);
 $t->setColor(0xff, 0xff, 0);
 $t->setHeight(1200);
 $t->addString("PHP fait du Flash avec Ming!!");
 $m = new SWFMovie();
 $m->setDimension(5400, 3600);
 $m->add($t);
 header('Content-type: application/x-shockwave-flash');
 $m->output();
?>
```

## 6.21.89 swftextfield->addstring() : Ajoute au texte

void **swftextfield->addstring** ( string string )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swftextfield->setname concatène la chaîne string avec la chaîne courante.

## 6.21.90 swftextfield->align() : Modifie l'alignement du texte

void **swftextfield->align** ( int alignement )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swftextfield->align change l'alignement du texte par alignement . Les valeurs valides pour alignement sont : SWFTEXTFIELD_ALIGN_LEFT, SWFTEXTFIELD_ALIGN_RIGHT, SWFTEXTFIELD_ALIGN_CENTER et SWFTEXTFIELD_ALIGN_JUSTIFY.

## 6.21.91 swftextfield->setbounds() : Sélectionne la largeur et hauteur du champs

void **swftextfield->setbounds** ( int width , int height )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez-en conscient, et utilisez cette fonction à vos risques et périls.

swftextfield->setbounds fixe la longueur du champ à width et sa hauteur à height . Si vous ne fixez pas les bords vous-mêmes, Ming tentera de les deviner lui-même (mais ne le laissez pas faire!!).

## 6.21.92 swftextfield−>setcolor() : Modifie la couleur du champs texte

void **swftextfield−>setcolor** ( int red , int green , int blue , *int a* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setcolor modifie la couleur du champ texte, en la remplaçant par la couleur fournie. Par défaut, c'est noir opaque. Les couleurs sont représentées en convention RGB.

## 6.21.93 swftextfield−>setfont() : Modifie la police du champs

void **swftextfield−>setfont** ( string font )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setfont remplace la police courante par la police font (police client?).

## 6.21.94 swftextfield−>setheight() : Modifie la hauteur de la police du champs texte

void **swftextfield−>setheight** ( int height )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setheight modifie la hauteur de la police du champ texte par height . Par défaut, c'est 240.

## 6.21.95 swftextfield−>setindentation() : Modifie l'indentation de la première ligne

void **swftextfield−>setindentation** ( int width )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setindentation modifie l'indentation de la première ligne du champ texte, en la fixant à width .

## 6.21.96 swftextfield−>setleftmargin() : Modifie la marge de gauche

void **swftextfield−>setleftmargin** ( int width )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setleftmargin modifie la marge de gauche du champ texte à width . Par défaut, c'est 0.

## 6.21.97 swftextfield−>setlinespacing() : Modifie l'espacement de lignes

void **swftextfield−>setlinespacing** ( int height )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setlinespacing modifie l'espacement de lignes, en le fixant à height . Par défaut, c'est 40.

## 6.21.98 swftextfield−>setmargins() : Modifie les marges du champs texte Flash

void **swftextfield−>setmargins** ( int left , int right )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setmargins modifie les deux marges du champ texte : left sera la nouvelle largeur de la marge de gauche, et right , celle de gauche. Par défaut, elles sont toutes les deux à 0.

## 6.21.99 swftextfield−>setname() : Nomme le champs texte

void **swftextfield−>setname** ( string name )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setname baptise le champ texte name . Cela servira pour les formulaires et les actions.

## 6.21.100 swftextfield−>setrightmargin() : Modifie la marge de droite

void **swftextfield−>setrightmargin** ( int width )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield−>setrightmargin modifie la marge de gauche du champ texte à width . Par défaut, c'est zéro.

## 6.21.101 swftextfield() : Crée un nouveau champs texte

new **swftextfield** ( *int flags* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

swftextfield crée un nouveau champ texte. Les champs textes sont moins souples que les swftext , car ils ne peuvent être tournés, mis à l'échelle ou incliné, mais ils peuvent être utilisés sous forme de champs de formulaire, et ils peuvent utiliser des polices navigateur.

Les flags optionnels modifient les comportements du champ. Ils peuvent prendre les valeurs suivantes :

- SWFTEXTFIELD_NOEDIT : indique que le champ ne doit pas être éditable.
- SWFTEXTFIELD_PASSWORD : indique que c'est un champ mot de passe
- SWFTEXTFIELD_DRAWBOX : dessine le contour du champ
- SWFTEXTFIELD_MULTILINE : autorise les lignes multiples
- SWFTEXTFIELD_WORDWRAP : autorise la mise en forme du texte
- SWFTEXTFIELD_NOSELECT : rend le champ non−sélectionnable

Les flags peuvent être combinés avec l'opérateur OR . Par exemple :

```
 <?php
$t = newSWFTextField(SWFTEXTFIELD_PASSWORD | SWFTEXTFIELD_NOEDIT);
?>
```

crée un champ de mot de passe totalement inéditable (et inutile).

SWFTextField a les méthodes suivantes : swftextfield−>setfont , swftextfield−>setbounds , swftextfield−>align , swftextfield−>setheight , swftextfield−>setleftmargin , swftextfield−>setrightmargin , swftextfield−>setmargins , swftextfield−>setindentation , swftextfield−>setlinespacing , swftextfield−>setcolor , swftextfield−>setname et swftextfield−>addstring .

# 6.22 Fonctions diverses

## 6.22.1 Introduction

Ces fonctions ont été placées là, car elles ne rentraient dans aucune catégorie adéquate.

## 6.22.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.22.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.22.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| ignore_user_abort | "0" | PHP_INI_ALL |
| highlight.string | #DD0000 | PHP_INI_ALL |
| highlight.comment | #FF9900 | PHP_INI_ALL |
| highlight.keyword | #007700 | PHP_INI_ALL |
| highlight.bg | #FFFFFF | PHP_INI_ALL |
| highlight.default | #0000BB | PHP_INI_ALL |
| highlight.html | #000000 | PHP_INI_ALL |
| browscap | NULL | PHP_INI_SYSTEM |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*ignore_user_abort bool*
: TRUE par défaut. Si changée à FALSE les scripts seront terminés aussitôt qu'ils essaieront d'envoyer des données à un client qui a annulé la connexion.

  Voir aussi ignore_user_abort .

*highlight.xxx string*
: Couleurs utilisées pour le mode Syntax Highlighting. Ces options peuvent prendre n'importe quelles valeurs valides dans <font color="??????">.

*browscap string*
: Nom du fichier de descriptif des clients HTML. Voir aussi get_browser .

## 6.22.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.22.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*CONNECTION_ABORTED (integer)*
*CONNECTION_NORMAL (integer)*
*CONNECTION_TIMEOUT (integer)*


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- connection_aborted
- connection_status
- connection_timeout
- constant
- define
- defined
- die
- eval
- exit
- get_browser
- highlight_file
- highlight_string
- ignore_user_abort
- pack
- show_source
- sleep
- uniqid
- unpack
- usleep


## 6.22.8 connection_status() : Retourne les bits de status de la connexion HTTP

int **connection_status** ( void )

connection_status retourne les bits de statut de la connexion. Reportez−vous à la section gestion des connexions pour plus de détails.

Voir aussi connection_aborted et ignore_user_abort .

## 6.22.9 connection_timeout() : Indique si le script a expiré

bool **connection_timeout** ( void )

connection_timeout retourne TRUE si le script a expiré. Reportez−vous à la section gestion des connexions pour plus de détails.

| Attention |
| --- |
| Deprecated<br><br>This function is deprecated, and doesn't even exist anymore as of 4.0.5. |

Voyez la section sur la Gestion de la connexion dans la section Caractéristiques pour plus de détails.

## 6.22.10 constant() : Retourne la valeur d'une constante

mixed **constant** ( string name )

constant retourne la valeur de la constante name .

constant est pratique lorsque vous devez lire la valeur d'une constante, mais que vous ne savez pas son nom que durant l'exécution du script. Par exemple, ce nom peut être le résultat d'une fonction.

Exempe avec constant

```
<?php
 define ("MAXSIZE", 100);
 echo MAXSIZE;
 echo constant("MAXSIZE");   // identique a la ligne precedente
?>
```

Voir aussi define , defined et la section sur les constantes .

## 6.22.11 define() : Définit une constante

bool **define** ( string name , mixed value , *bool case_insensitive* )

define définit la constante de nom name et de valeur value . Voyez la section sur les constantes pour plus de détails sur les constantes.

Le nom de la constante est donné par le paramètre name ; sa valeur est donnée par value .

Le troisième paramètre optionnel case_insensitive est une valeur booléenne. S'il vaut TRUE , le nom de la constante sera insensible à la casse : CONSTANT et Constant représentent des valeurs identiques. Par défaut, ces constantes représenteront des valeurs différentes.

Définition d'une constante

```
<?php
define ("CONSTANT", "Bonjour à tous.");
echo CONSTANT; // affiche "Bonjour à tous."
echo Constant; // affiche "Constant" et une alerte

define ("BONJOUR", "Salut toi.",TRUE);
echo BONJOUR; // affiche "Salut toi."
echo Bonjour; // affiche "Salut toi."

?>
```

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi defined , constant et la section sur les constantes .

## 6.22.12 defined() : Vérifie l'existence d'une constante

bool **defined** ( string name )

defined retourne TRUE si la constante nommée name a été définie, et FALSE sinon. Pour tester l'existence d'une fonction utilisez function_exists .

Vérifier la présence de constantes avec defined

```
<?php
if (defined("CONSTANT")){
 // Notez que le nom de la constante est entre guillemets
    echo CONSTANT;
     }
?>
```

Voir aussi define , constant , get_defined_constants , la section sur les constantes et function_exists .

## 6.22.13 die() : Alias de la fonction exit

void **die** ( string message )

die est un alias de exit .

## 6.22.14 eval() : Exécute une chaîne comme un script PHP

mixed **eval** ( string code_str )

eval évalue la chaîne code_str comme un script PHP. Parmi les utilisations possibles, cette fonction permet de stocker du code dans une base de données, pour utilisation ultérieure.

Il faut bien garder en tête que le code passé à eval doit être valide, y compris les points virgules de fin de ligne et les séquences d'échappement, sinon l'exécution se terminera.

N'oubliez pas que les variables utilisées dans la fonction eval resteront accessibles dans le script principal.

Une comande return terminera l'évaluation de la chaîne immédiatement. En PHP 4, eval retourne NULL sauf si return est appelée dans le code évalué, auquel cas la valeur passée à return est retournée. En PHP 3, eval ne retourne par de valeur.

Exemple avec eval − concaténation de texte

```
<?php
 $string = 'tasse';
 $name = 'café';
 $str = 'Ceci est une $string avec mon $name dedans.<br />';
 echo $str;
 eval( "\$str = \"$str\";" );
 echo $str;
?>
```

L'exemple ci−dessus devrait afficher :
Ceci est une $string avec mon $name dedans.
Ceci est une tasse avec mon café dedans.

Array

## 6.22.15 exit() : Termine le script courant

void **exit** ( *string status* ) void **exit** ( int status )

| Note |
| --- |
| exit n'est pas une véritable fonction, mais un élément de langage. |

exit termine l'exécution du script courant. Elle n'a pas de valeur de retour (et pour cause!), mais elle utilisera le message status comme message de fin d'exécution.

| Note |
| --- |
| La fonction die est un alias de la fonction exit . |

Exemple avec exit

```
<?php
 $filename = '/chemin/jusqua/fichier';
 $file = fopen ($filename, 'r')
 or exit("Impossible d'ouvrir le fichier $filename");
?>
```

Voir aussi register_shutdown_function .

## 6.22.16 get_browser() : Indique de quoi est capable le navigateur client.

object **get_browser** ( *string user_agent* )

get_browser essaie de déterminer les capacités du navigateur client. Cela se fait en lisant les informations dans le fichier browscap.ini . Par défaut, la valeur de $HTTP_USER_AGENT est utilisée. Cependant, vous pouvez passer n'importe quelle valeur avec le paramètre optionnel user_agent à get_browser .

Les informations sont retournées sous forme d'un objet, dont les différents membres contiendront des informations, telles que les versions majeures et mineures et des chaînes d'identification; des booléens pour des caractéristiques telles que frames, JavaScript, et cookies; et ainsi de suite.

Même si browscap.ini contient des informations sur de nombreux clients, il compte sur les utilisateurs pour être mis à jour. Le format du fichier est facilement compréhensible.

L'exemple suivant montre comment on peut lister les informations disponibles :

Exemple avec get_browser

```
<?php
echo $_SERVER['HTTP_USER_AGENT'] . "<hr />\n";

$browser = get_browser();

foreach ($browser as $name => $value) {
    print "<b>$name</b> $value <br />\n";
}

?>
```

L'affichage devrait ressembler à ceci :

```
 Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586)<hr />
<b>browser_name_pattern:</b> Mozilla/4\.5.*<br />
<b>parent:</b> Netscape 4.0<br />
<b>platform:</b> Linux<br />
<b>majorver:</b> 4<br />
<b>minorver:</b> 5<br />
<b>browser:</b> Netscape<br />
<b>version:</b> 4<br />
<b>frames:</b> 1<br />
```

```
<b>tables:</b> 1<br />
<b>cookies:</b> 1<br />
<b>backgroundsounds:</b> <br />
<b>vbscript:</b> <br />
<b>javascript:</b> 1<br />
<b>javaapplets:</b> 1<br />
<b>activexcontrols:</b> <br />
<b>beta:</b> <br />
<b>crawler:</b> <br />
<b>authenticodeupdate:</b> <br />
<b>msn:</b> <br />
```

Pour fonctionner, votre configuration browscap dans le php.ini doit pointer sur le fichier browscap.ini de votre système. browscap.ini n'est pas fourni avec PHP mais vous pouvez trouver une version à jour sur ce site . Par défaut, la directive browscap est désactivée.

| Note |
| --- |
| La valeur cookies indique simplement que le navigateur est capable d'accepter les cookies, et n'indique pas si l'utilisateur les a activé sur son navigateur. Le seul moyen de tester l'activation des cookies est d'en poser un avec la fonction setcookie , de recharger la page et de vérifier que le cookie existe toujours. |

| Note |
| --- |
| Sur les versions de PHP plus ancienne que la 4.0.6, vous devez passer le nom du naviageteur comme troisième paramètre optionnel user_agent , et si la directive register_globals vaut off . Dans ce cas, vous devrez utiliser $HTTP_SERVER_VARS['HTTP_USER_AGENT'] . |

## 6.22.17 highlight_file() : Colorisation de la syntaxe d'un fichier

mixed **highlight_file** ( string filename , *bool return* )

highlight_file affiche la syntaxe colorisée du fichier filename , en utilisant les couleurs définies dans le moteur interne de PHP.

Si le second paramètre optionnel return vaut TRUE alors highlight_file retournera le code généré, au lieu de l'afficher. Si le second paramètre ne vaut pas TRUE alors highlight_file retournera TRUE en cas de succès, et FALSE sinon.

| Note |
| --- |
| Le paramètre return a été ajouét en PHP 4.2.0. Avant cela, la fonction se comportait comme si ce paramètre avait sa valeur par défaut de FALSE |

| Note |
| --- |
| Beaucoup de soin doit être apporté lors de l'utilisation de show_source et highlight_file pour s'assurer que vous ne révélez pas d'informations critiques telles que des mots de passe ou d'autres informations qui pourrait créer des fuites d'information. |

| Note |
| --- |
| Depuis PHP 4.2.1, cette fonction est aussi affectée par safe_mode et open_basedir . |

| Colorisation d'URL |
| --- |
| Pour configurer une URL qui peut coloriser n'importe quel script que vous lui passez, nous avons besoin d'utiliser la directive Apache "ForceType", pour générer une URL exploitable, puis utiliser la fonction highlight_file pour afficher un code propre.<br><br>Dans votre configuration HTTP httpd.conf , vous pouvez ajouter le code suivant :<br><br>`<Location /source>`<br>`  ForceType application/x-httpd-php` |

```
</Location>
```

Puis, faire un fichier appelé source , que vous placez dans votre racine de site web.

```
 <html>
<head>
<title>Coloration de la source</title>
</head>
<body bgcolor="white">
<?php
    $script = getenv("PATH_TRANSLATED");
    if (!$script) {
      echo "<br /><b>Erreur : Le nom du script est nécessaire</b><br>";
    } else {
      if (ereg("(\\.php|\\.inc)$",$script)) {
        echo "<h1>Source de : " . getenv('PATH_INFO') . "</h1>\n<hr />\n";
        highlight_file($script);
      } else {
        echo "<h1>Erreur: Seuls les noms de fichier PHP ou de fichiers PHP inclus sont autorisés</h1>";
      }
    }
    echo "<hr />Traitement : " . date("Y/M/d H:i:s", time());
?>
</body>
</html>
```

Alors, vous pourrez utiliser une URL telle que celle ci−dessous pour afficher une version colorisée de votre script /path/to/script.php .

```
 http://votre.serveur.com/source/jusqu/au/script.php
```

Voir aussi highlight_string et show_source .

## 6.22.18 highlight_string() : Applique la syntaxe colorisée à du code PHP

mixed **highlight_string** ( string str , *bool* *return* )

highlight_string affiche la version colorisée du code PHP contenu dans le paramètre str , en utilisant les couleurs du système interne de colorisation de PHP.

Si le second paramètre optionnel return est fourni, et vaut TRUE alors highlight_string retournera la chaîne colorisée au lieu de l'afficher immédaitement. Si le second paramètre ne vaut pas TRUE alors highlight_string retournera TRUE en cas de succès, et FALSE sinon.

| Note |
| --- |
| Le paramètre return est disponible depuis PHP 4.2.0. Auparavant, il prenait la valeur par défaut de FALSE |

Voir aussi highlight_file et show_source .

## 6.22.19 ignore_user_abort() : Active l'interruption de script sur déconnexion du visiteur

int **ignore_user_abort** ( *bool* *setting* )

ignore_user_abort active l'option décidant si, lors de la déconnexion du client, le script doit poursuivre son exécution ou non. La fonction renvoie le paramétrage précédent et elle peut être appelée sans argument pour ne pas changer le paramétrage courant. Voir le paragraphe sur la gestion des connexions dans le chapitre fonctionnalités pour une description plus complète des manipulations de connexion avec PHP .

Voir aussi connection_aborted et connection_status .

## 6.22.20 pack() : Compacte des données dans une chaîne binaire

string **pack** ( string format , *mixed* *args* )

pack compacte les arguments dans une chaîne binaire, suivant le format format . pack retourne la chaîne binaire.

L'idée vient du Perl et tout le formatage fonctionne de la même façon qu'en Perl, mais quelques formats manquent encore (comme, "u" ). La chaîne de format est composée d'une série de codes de formats, suivis par un quantificateur optionnel. Le quantificateur peut être un entier, ou * pour la répétition indéfinie. Pour les formats a, A, h et H, le quantificateur spécifie combien de caractères d'un argument sont pris; pour @, c'est la position absolue où placer les données, et pour le reste, c'est le nombre de répétitions. Actuellement, les formats suivants sont implémentés :

- Une chaîne complétée avec NULL
- Une chaîne complétée avec espace (SPACE)
- Chaîne hexadécimale h, bit de poids faible en premier.
- Chaîne hexadécimale H, bit de poids fort en premier.
- c caractère signé
- C caractère non signé
- s entier court signé (toujours sur 16 bits, ordre des bits dépendant de la machine).
- S entier court non signé (toujours 16 bits, ordre des bits dépendant de la machine).
- n entier court signé (toujours 16 bits, ordre des bits big endian)
- v entier cours non signé (toujours 16 bits, ordre des bits little endian)
- i entier signé (taille et ordre des bits dépendants de la machine)
- I entier non signé (taille et ordre des bits dépendants de la machine)
- l entier long signé (toujours 32 bits, ordre des bits dépendant de la machine)
- L entier long non signé (toujours 32 bits, ordre des bits dépendant de la machine)
- N entier long non signé (toujours 16 bits, ordre des bits big endian)
- V entier long non signé (toujours 16 bits, ordre des bits little endian)
- f nombre à virgule flottante (taille et représentation dépendantes de la machine)
- d nombre à virgule flottante double (taille et représentation dépendantes de la machine)
- x bit NULL
- X recule d'un octet
- @ rempli avec NULL , jusqu'à une position absolue

Compactage d'une chaîne

```
<?php
 $binarydata = pack ("nvc*", 0x1234, 0x5678, 65, 66);
?>
```

La chaîne binaire résultante aura 6 octets de long, et contiendra la séquence 0x12, 0x34, 0x78, 0x56, 0x41, 0x42.

Notez que la distinction entre signé et non signé n'affecte que la fonction unpack , tandis que la fonction pack fournira le même résultat pour les deux formats.

De plus, notez que PHP enregistre de manière interne et intégrale les valeurs : cette représentation dépend de la machine. Si vous essayez d'enregistrer une valeur trop grande, elle risque d'être convertie et de donner lieu à des effets de bords vicieux.

Voir aussi unpack .

## 6.22.21 show_source() : Génère la syntaxe colorisée d'un fichier PHP

bool **show_source** ( string filename , *bool return* )

show_source est un alias de highlight_file .

Voir aussi highlight_string et highlight_file .

## 6.22.22 sleep() : Retarde l'exécution

void **sleep** ( int seconds )

sleep retarde l'exécution du programme pendant seconds secondes.

Voir aussi usleep .

## 6.22.23 uniqid() : Génère un identifiant unique

string **uniqid** ( string prefix , *bool lcg* )

uniqid retourne un identifiant préfixé unique, basé sur l'heure courante, en micro−secondes. Le préfixe peut servir à identifier facilement différents hôtes, si vous générez simultanément des fichiers depuis plusieurs hôtes, à la même micro−seconde. prefix peut prendre jusqu'à 114 caractères.

Si le paramètre optionnel lcg est TRUE , uniqid ajoutera une entropie "combined LCG" à la fin de la valeur retournée, ce qui renforcera encore l'unicité de l'identifiant.

Sans prefix (préfixe vide), la chaîne retournée fera 13 caractères. Si lcg est à TRUE , elle fera 23 caractères.

| Note |
| --- |
| Le paramètre lcg est utilisé à partir de PHP 4 et PHP 3.0.13 et ultérieurs. |

Si vous voulez utiliser un identifiant unique, ou bien gérer des cookies, il est recommandé d'utiliser un code tel que celui−ci :

Exemple avec uniqid

```
<?php
 $token = md5 (uniqid (""));
  // pas de section aleatoire.
 $better_token = md5 (uniqid (rand()));
  // mieux, difficile a deviner
?>
```

Ceci va créer un identifiant de 32 caractères (un nombre hexadécimal de 128 ) qui sera très difficile à prédire.

## 6.22.24 unpack() : Déconditionne des données depuis une chaîne binaire.

array **unpack** ( string format , string data )

unpack déconditionne des données depuis une chaîne binaire avec le format format . unpack retourne un tableau contenant les éléments déconditionnés.

unpack se comporte légèrement différemment de la version Perl car les données déconditionnées sont stockées dans un tableau. Pour cela, il faut donner un nom à chaque format utilisé et les séparer par des slash (/).

Exemple avec unpack

```
<?php
  $array = unpack ("c2chars/nint", $binarydata);
?>
```

Le tableau résultant contiendra les entrées suivantes : "chars1", "chars2" et "int".

Pour plus de détails, reportez−vous à: pack

Il faut noter que PHP gère les valeurs en interne sous forme signée. Si vous déconditionnez une valeur qui est aussi grande que la taille utilisée en interne par PHP, le résultat se trouvera être un nombre négatif, même s'il a été déconditionné avec l'option " non signé ".

Voir aussi pack pour des informations sur les formats disponibles.

## 6.22.25 usleep() : Retarde l'exécution en micro−secondes

void **usleep** ( int micro_seconds )

sleep retarde l'exécution du programme pendant micro_seconds micro−secondes.

Voir aussi sleep .

Note

usleep est inopérante sous Windows

# 6.23 mnoGoSearch

## 6.23.1 Introduction

Ces fonctions donnent l'accès à mnoGoSearch (anciennement UdmSearch), moteur de recherche du monde libre. mnoGoSearch est une moteur de recherche complet, destinés aux intranet et serveurs web, distribué sous licence GNU. mnoGoSearch offre des fonctionnalités unique, qui en font un excellent outil pour un grand nombre d'applications de recherche dans votre site : recherche de recettes de cuisines ou dans les journaux, recherche dans un site FTP, dans les groupes de news, etc... Il offre un système d'indexation de textes pour les fichiers HTML, PDF et documents textes. mnoGoSearch est constitué de deux parties : l'indexeur, qui effectue les recherches et le moteur de recherche. L'indexeur passe en revue récursivement les sites HTTP, FTP, NEWS ou encore les fichiers locaux, et enregistre des méta−données dans les bases MySQL, pour optimiser les recherches ultérieures. Une fois que tous les documents ont été référéncés, ils sont accessibles au moteur de recherche. Celui−ci est utilisable par interface web. Les langages C CGI, Perl et PHP sont supportés pour effectuer les recherches.

Plus de détails sur le site officiel de mnoGoSearch : http://www.mnogosearch.ru/ .

| Note |
| --- |
| Cette extension n'est pas disponible sur les plate−formes Windows. |

## 6.23.2 Pré−requis

Téléchargez mnoGosearch sur le site http://www.mnogosearch.ru/ et installez le sur votre système. Vous aurez besoin de la version 3.1.10 ou plus récente de mnoGoSearch, pour pouvoir utiliser ces fonction.

## 6.23.3 Installation

Pour pouvoir les utiliser, vous devez inclure le support en ajoutant l'option −−with−mnogosearch . Si vous utilisez cette option sans indiquer le chemin jusqu'à mnogosearch, PHP essaiera de le trouver dans le dossier /usr/local/mnogosearch . Si vous avez installé mnogosearch dans un autre endroit, vous devez l'indiquer comme ceci : −−with−mnogosearch=DIR .

| Note |
| --- |
| PHP supporte naturellement MySQL. Il faut savoir que mnoGoSearch n'est pas compatible avec la librairie interne de PHP, et ne peut fonctionner qu'avec les librairies génériques MySQL. Par conséquent, si vous utilisez mnoGoSearch avec MySQL, indiquez le dossier d'installation de MySQL durant la configuration avec l'option : −−with−mnogosearch −−with−mysql=/usr . |

## 6.23.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.23.5 Types de ressources

## 6.23.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*UDM_FIELD_URLID (entier)*
*UDM_FIELD_URL (entier)*
*UDM_FIELD_CONTENT (entier)*
*UDM_FIELD_TITLE (entier)*
*UDM_FIELD_KEYWORDS (entier)*
*UDM_FIELD_DESC (entier)*
*UDM_FIELD_DESCRIPTION (entier)*
*UDM_FIELD_TEXT (entier)*
*UDM_FIELD_SIZE (entier)*
*UDM_FIELD_RATING (entier)*
*UDM_FIELD_SCORE (entier)*
*UDM_FIELD_MODIFIED (entier)*
*UDM_FIELD_ORDER (entier)*
*UDM_FIELD_CRC (entier)*
*UDM_FIELD_CATEGORY (entier)*
*UDM_FIELD_LANG (entier)*
*UDM_FIELD_CHARSET (entier)*
*UDM_PARAM_PAGE_SIZE (entier)*
*UDM_PARAM_PAGE_NUM (entier)*
*UDM_PARAM_SEARCH_MODE (entier)*
*UDM_PARAM_CACHE_MODE (entier)*
*UDM_PARAM_TRACK_MODE (entier)*
*UDM_PARAM_PHRASE_MODE (entier)*
*UDM_PARAM_CHARSET (entier)*
*UDM_PARAM_LOCAL_CHARSET (entier)*
*UDM_PARAM_BROWSER_CHARSET (entier)*
*UDM_PARAM_STOPTABLE (entier)*
*UDM_PARAM_STOP_TABLE (entier)*
*UDM_PARAM_STOPFILE (entier)*
*UDM_PARAM_STOP_FILE (entier)*
*UDM_PARAM_WEIGHT_FACTOR (entier)*
*UDM_PARAM_WORD_MATCH (entier)*
*UDM_PARAM_MAX_WORD_LEN (entier)*
*UDM_PARAM_MAX_WORDLEN (entier)*
*UDM_PARAM_MIN_WORD_LEN (entier)*
*UDM_PARAM_MIN_WORDLEN (entier)*
*UDM_PARAM_ISPELL_PREFIXES (entier)*
*UDM_PARAM_ISPELL_PREFIX (entier)*
*UDM_PARAM_PREFIXES (entier)*
*UDM_PARAM_PREFIX (entier)*
*UDM_PARAM_CROSS_WORDS (entier)*
*UDM_PARAM_CROSSWORDS (entier)*
*UDM_PARAM_VARDIR (entier)*
*UDM_PARAM_DATADIR (entier)*
*UDM_PARAM_HLBEG (entier)*
*UDM_PARAM_HLEND (entier)*

*UDM_PARAM_SYNONYM (entier)*
*UDM_PARAM_SEARCHD (entier)*
*UDM_PARAM_QSTRING (entier)*
*UDM_PARAM_REMOTE_ADDR (entier)*
*UDM_LIMIT_CAT (entier)*
*UDM_LIMIT_URL (entier)*
*UDM_LIMIT_TAG (entier)*
*UDM_LIMIT_LANG (entier)*
*UDM_LIMIT_DATE (entier)*
*UDM_PARAM_FOUND (entier)*
*UDM_PARAM_NUM_ROWS (entier)*
*UDM_PARAM_WORDINFO (entier)*
*UDM_PARAM_WORD_INFO (entier)*
*UDM_PARAM_SEARCHTIME (entier)*
*UDM_PARAM_SEARCH_TIME (entier)*
*UDM_PARAM_FIRST_DOC (entier)*
*UDM_PARAM_LAST_DOC (entier)*
*UDM_MODE_ALL (entier)*
*UDM_MODE_ANY (entier)*
*UDM_MODE_BOOL (entier)*
*UDM_MODE_PHRASE (entier)*
*UDM_CACHE_ENABLED (entier)*
*UDM_CACHE_DISABLED (entier)*
*UDM_TRACK_ENABLED (entier)*
*UDM_TRACK_DISABLED (entier)*
*UDM_PHRASE_ENABLED (entier)*
*UDM_PHRASE_DISABLED (entier)*
*UDM_CROSS_WORDS_ENABLED (entier)*
*UDM_CROSSWORDS_ENABLED (entier)*
*UDM_CROSS_WORDS_DISABLED (entier)*
*UDM_CROSSWORDS_DISABLED (entier)*
*UDM_PREFIXES_ENABLED (entier)*
*UDM_PREFIX_ENABLED (entier)*
*UDM_ISPELL_PREFIXES_ENABLED (entier)*
*UDM_ISPELL_PREFIX_ENABLED (entier)*
*UDM_PREFIXES_DISABLED (entier)*
*UDM_PREFIX_DISABLED (entier)*
*UDM_ISPELL_PREFIXES_DISABLED (entier)*
*UDM_ISPELL_PREFIX_DISABLED (entier)*
*UDM_ISPELL_TYPE_AFFIX (entier)*
*UDM_ISPELL_TYPE_SPELL (entier)*
*UDM_ISPELL_TYPE_DB (entier)*
*UDM_ISPELL_TYPE_SERVER (entier)*
*UDM_MATCH_WORD (entier)*
*UDM_MATCH_BEGIN (entier)*
*UDM_MATCH_SUBSTR (entier)*
*UDM_MATCH_END (entier)*


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies

- udm_add_search_limit
- udm_alloc_agent
- udm_api_version
- udm_cat_list
- udm_cat_path
- udm_check_charset
- udm_check_stored
- udm_clear_search_limits
- udm_close_stored
- udm_crc32
- udm_errno
- udm_error
- udm_find
- udm_free_agent
- udm_free_ispell_data
- udm_free_res
- udm_get_doc_count
- udm_get_res_field
- udm_get_res_param
- udm_load_ispell_data
- udm_open_stored
- udm_set_agent_param

## 6.23.8 udm_alloc_agent() : Alloue une session mnoGoSearch

resource **udm_alloc_agent** ( string dbaddr , *string dbmode* )

udm_alloc_agent retourne un agent mnogosearch en cas de succès, FALSE en cas d'erreur. udm_alloc_agent crée une session avec les paramètres de base de données.

dbaddr est une description de base de données formaté comme une URL. Les options (type, hôte, nom de base de données, port, utilisateur ou mot de passe) servent à se connecter à la base de données SQL. Ne passez aucune valeur si vous souhaitez utiliser le support des fichiers texte intégré. Sinon, utilisez le format : DBAddr DBType:[//[DBUser[:DBPass]@]DBHost[:DBPort]]/DBName/ . Actuellement, les valeurs de DBType possibles sont : mysql, pgsql, msql, solid, mssql, oracle, ibase. En fait, si vous avez ajouté un support natif, cette option est inutile. Mais les utilisateurs ODBC doivent spécifier une des valeurs supportées. Si votre type de base de données n'est pas supporté, utilisez le terme "unknown".

dbmode − Vous pouvez sélectionner le mode de stockage des mots dans la base de données. Si vous indiquez "single", tous les mots seront sotckés dans la même table. Si vous indiquez "multi", les mots seront situés dans différentes tables, suivant leur taille. Le mode "multi" est généralement plus rapide, mais requiert plus de tables. Si le mode "crc" est sélectionné, mnoGoSearch enregistrera un entier de 32 bits, calculé avec l'algorithme CRC32, plutot que des des mots. Ce mode requiert moins d'espace disque, et il est beaucoup plus rapide que les modes "single" et "multi". "crc−multi" utilise la même technique de stockage que le mode "crc", mais il stocke aussi les mots dans différentes tables suivant leur taille. Format: DBMode single/multi/crc/crc−multi.

| Note |
| --- |
| dbaddr et dbmode doit correspondre à ceux qui sont utilisés lors de l'indexation. |

| Note |
| --- |
| En réalité, udm_alloc_agent n'ouvre pas de connexion, et donc, ne vérifie ni le nom d'utilisateur, ni |

le mot de passe.

## 6.23.9 udm_api_version() : Lit la version des API mnoGoSearch

int **udm_api_version** ( void )

udm_api_version retourne le numéro de version des API mnoGoSearch. Par exemple, si mnoGosearch 3.1.10 est utilisé, udm_api_version retournera 30110 .

udm_api_version permet aux utilisateurs d'identifier quelles sont les API disponibles. Par exemple, udm_get_doc_count n'est disponible qu'à partir de mnoGoSearch 3.1.11.

Exemple avec udm_api_version

```
<?php
  if (udm_api_version() >= 30111) {
    print  "Nombre total d'URL dans la base : ".
      udm_get_doc_count($udm)."<br>\n";
  }
?>
```

## 6.23.10 udm_cat_list() : Liste toutes les catégories soeurs d'une catégorie

array **udm_cat_list** ( resource agent , string category )

udm_cat_list retourne un tableau contenant la liste de toutes les catégories de même niveau que la catégorie courante.

Cette fonction est pratique pour réaliser des arbres à partir des catégories.

udm_cat_list retourne un tableau avec le format suivant :

Le tableau est constitué de paires. Les index pairs contiennent les chemins de catégories, les index impairs contiennent les noms des catégories correspondantes.

```
$array[0] contiendra '020300'
  $array[1] contiendra 'Marseille'
  $array[2] contiendra '020301'
  $array[3] contiendra 'Lille'
  $array[4] contiendra '020302'
  $array[5] contiendra 'Lyon'
  ...
 etc.
```

Ce qui peut être affiché comme ceci :

```
Marseille
  Lille
  Lyon
  ...
```

Exemple avec udm_cat_list

```
<?php
$cat_list_arr = udm_cat_list($udm_agent,$cat);
$cat_list = '';
for ($i=0; $i<count($cat_list_arr); $i+=2) {
   $path = $cat_list_arr[$i];
```

```
    $name = $cat_list_arr[$i+1];
    $cat_list .= "<a href=\"$_SERVER[PHP_SELF]?cat=$path\">$name</a><br>";
 }
?>
```

Voir aussi udm_cat_path .

## 6.23.11 udm_cat_path() : Lit le chemin de la catégorie courante

array **udm_cat_path** ( resource agent , string category )

udm_cat_path retourne un tableau listant les catégories depuis la racine jusqu'à la catégorie courante.

Le paramètre agent est un identifiant de résultat, obtenu après un appel à udm_alloc_agent .

category − La catégorie courante : celle dont on veut le chemin.

udm_cat_path retourne un tableau avec le format suivant :

Le tableau est constitué de paires. Les index pairs contiennent les chemins de catégories, les index impairs contiennent les noms des catégories correspondantes.

Par exemple, l'appel $array=udm_cat_path($agent, '02031D'); peut retourner le tableau suivant :

```
$array[0] contiendra ''
 $array[1] contiendra 'Root'
 $array[2] contiendra '02'
 $array[3] contiendra 'Sport'
 $array[4] contiendra '0203'
 $array[5] contiendra 'Foot'
 $array[4] contiendra '02031D'
 $array[5] contiendra 'PSG'
```

Spécifier une catégorie avec un format hiérachisé

```
 <?php
//Spécifier le chemin de la catégorie courante avec le format suivant
 // '> Root > Sport > Foot > PSG'
  $cat_path_arr = udm_cat_path($udm_agent,$cat);
  $cat_path = '';
  for ($i=0; $i<count($cat_path_arr); $i+=2) {
    $path = $cat_path_arr[$i];
    $name = $cat_path_arr[$i+1];
    $cat_path .= " > <a href=\"$_SERVER[PHP_SELF]?cat=$path\">$name</a> ";
  }
?>
```

Voir aussi udm_cat_list .

## 6.23.12 udm_check_charset() : Vérifie si un jeu de caractère est supporté par mnogosearch

bool **udm_check_charset** ( resource agent , string charset )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

### 6.23.13 udm_check_stored() : Vérifie la connexion MnoGoSearch avec document stocké

int **udm_check_stored** ( resource agent , int link , string doc_id )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

### 6.23.14 udm_clear_search_limits() : Annule toutes les limitates de recherche MnoGoSearch

bool **udm_clear_search_limits** ( resource agent )

udm_clear_search_limits annule toutes les limitations de recherche imposées, et retourne TRUE .

Voir aussi udm_add_search_limit .

### 6.23.15 udm_close_stored() : Ferme la connexion MnoGoSearch avec le document enregistré

int **udm_close_stored** ( resource agent , int link )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

### 6.23.16 udm_crc32() : Retourne la somme de contrôle CRC32 d'une chaîne

int **udm_crc32** ( resource agent , string str )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

### 6.23.17 udm_errno() : Numéro d'erreur mnoGoSearch

int **udm_errno** ( resource agent )

udm_errno retourne le numéro d'erreur mnoGoSearch, ou bien 0 sinon.

Le paramètre agent est un identifiant de résultat, obtenu après un appel à udm_alloc_agent .

udm_errno retourne le numéro de l'erreur généré par l'agent agent .

### 6.23.18 udm_error() : Message d'erreur mnoGoSearch

string **udm_error** ( resource agent )

udm_errno retourne le message d'erreur mnoGoSearch, ou bien une chaîne vide sinon.

Le paramètre agent est un identifiant de résultat, obtenu après un appel à udm_alloc_agent .

udm_error retourne le numéro de l'erreur généré par l'agent agent .

## 6.23.19 udm_find() : Effectue une recherche

resource **udm_find** ( resource agent , string query )

udm_add_search_limit retourne TRUE en cas de succès, et FALSE en cas d'erreur.

La recherche en elle−même. Le premier argument agent est la session, le second est la query . Pour rechercher, entrez les mots avec lesquels que vous voulez faire une recherche, puis cliquez sur le bouton d'envoi. Par exemple, " mysql odbc ". Vous ne devez pas utiliser de guillemets doubles " , car ils sont utilisés par mnoGoSearch pour séparer une requête en mots. Avec l'exemple ci−dessus, mnoGoSearch va rechercher les pages contenant "mysql" et/ou "odbc". Les meilleures réponses seront classées en premier, et affichées en tête de liste. Si vous sélectionnez le mode de recherche "tous" ("ALL"), la recherche va retourner les documents qui contiennent l'un ou l'autre des mots que vous avez entré. Dans le cas où vous utilisez le mode "ANY", la recheche retourne la lsite des documents qui contiennent l'un ou l'autre des mots. Si vous voulez accéder aux fonctions avancées de recherche, vous pouvez utiliser le mode "BOOL", qui vous permet d'entrer directement des requêtes.

mnoGoSearch utilise les opérateurs booléean suivants :

& − AND, ET logique. Par exemple, "mysql & odbc". mnoGoSearch recherche toutes les URL qui contiennent à la fois les mots "mysql" et "odbc".

| − OR, OU logique. Par exemple, "mysql | odbc". mnoGoSearch recherche toutes les URL qui contiennent soit "mysql", soit "odbc".

~ − NOT, NON logique. Par exemple, "mysql & ~odbc". mnoGoSearch recherche toutes les URL qui contiennent le mot "mysql" mais ne contiennent pas le mot "odbc". Attention : la requête "~odbc" ne trouvera rien!

() − Groupage de commandes pour les requêtes complexes : par exemple, "(mysql | msql) & ~postgres". Le mode par requête est simple et puissant à la fois. Vous pouvez utiliser les commandes booléennes habituelles avec ce mode.

## 6.23.20 udm_free_agent() : Détruit une session mnoGoSearch

int **udm_free_agent** ( resource agent )

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre res est un identifiant de résultat, obtenu après un appel à udm−find .

udm_free_agent détruit l'agent de recherche créé par udm_alloc_agent .

## 6.23.21 udm_free_ispell_data() : Libère la mémoire allouée pour ispell par MnoGoSearch

bool **udm_free_ispell_data** ( int agent )

udm_free_ispell_data retourne toujours TRUE .

agent − Agent mnoGoSearch obtenu après un appel à udm_alloc_agent .

| Note |
| --- |
| udm_free_ispell_data est supportée à partir de la version 3.1.12 de mnoGoSearch et elle ne fait strictement rien avec les versions précédentes. |

## 6.23.22 udm_free_res() : Libère un résultat mnoGoSearch

bool **udm_free_res** ( resource res )

udm_free_res libère la mémoire de tous les résultats générés.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre res est un identifiant de résultat, obtenu après un appel à udm_find .

## 6.23.23 udm_get_doc_count() : Lit le nombre total de documents dans les bases MnoGoSearch

int **udm_get_doc_count** ( resource agent )

udm_get_doc_count retourne le nombre de document dans les bases de données.

agent − Agent mnoGoSearch obtenu après un appel à udm_alloc_agent .

| Note |
| --- |
| udm_get_doc_count est supporté à partir de la version mnoGoSearch 3.1.11 ou plus récent. |

## 6.23.24 udm_get_res_field() : Lit un champ de résultat mnoGoSearch

string **udm_get_res_field** ( resource res , int row , int field )

udm_alloc_agent retourne la valeur du champ field dans la ligne row , du résultat res , et FALSE sinon.

Le paramètre res est un identifiant de résultat, obtenu après un appel à udm−find .

Le paramètre row est le numéro du lien dans la page courante. Il peut valoir de 0 jusqu'à udm_param_num_rows .

Le paramètre field est l'identifiant de champ, et peut prendre l'une des valeurs suivantes :

- UDM_FIELD_URL : Champs URL

- UDM_FIELD_CONTENT : Champs "Content−type" (par exemple, "text/html").
- UDM_FIELD_TITLE : Titre du document.
- UDM_FIELD_KEYWORDS : Mots clés du document (balise META KEYWORDS).
- UDM_FIELD_DESC : Description du document (balise META DESCRIPTION).
- UDM_FIELD_TEXT : Corps du document (balise body, les premières lignes pour donner une idée du document).
- UDM_FIELD_SIZE : Taille du document.
- UDM_FIELD_URLID : Identifiant unique de l'URL.
- UDM_FIELD_RATING : Score de la page (calculé par mnoGoSearch).
- UDM_FIELD_MODIFIED : Date de modification au format unixtimestamp.
- UDM_FIELD_ORDER : Le nombre de documents trouvés.
- UDM_FIELD_CRC : La valeur CRC du document.

## 6.23.25 udm_get_res_param() : Lit les paramètres de résultats mnoGoSearch

string **udm_get_res_param** ( resource res , int param )

udm_get_res_param retourne les paramètres de résultat en cas de succès, FALSE en cas d'erreur.

Le paramètre res est un identifiant de résultat, obtenu après un appel à udm−find .

Le paramètre param peut prendre les valeurs suivantes :

- UDM_PARAM_NUM_ROWS − nombre de liens trouvés dans le groupe de résultat courant. C'est la valeur de UDM_PARAM_PAGE_SIZE pour tous les groupes, sauf le dernier.
- UDM_PARAM_FOUND − Nombre total de résultats trouvés.
- UDM_PARAM_WORDINFO − Informations sur les mots trouvés, c'est−à−dire que la recherche "un bon libre" retournera "un: stopword, bon:5637, livre: 120"
- UDM_PARAM_SEARCHTIME − Temps de recherche en secondes
- UDM_PARAM_FIRST_DOC − le numéro du premier document affiché dans le groupe.
- UDM_PARAM_LAST_DOC − le numéro du dernier document affiché dans le groupe.

## 6.23.26 udm_load_ispell_data() : Charge les données ispell

bool **udm_load_ispell_data** ( resource agent , int var , string val1 , string val2 , int flag )

udm_load_ispell_data charge des données ispell. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

agent − Agent mnoGoSearch obtenu après un appel à udm_alloc_agent .

var − paramètre indiquant la source des données ispell.

Après avoir utilisé cette fonction, pensez à libérer les données de la mémoire avec udm_free_ispell_data , même si vous utilisez le mode UDM_ISPELL_TYPE_SERVER .

Le mode de plus rapide est UDM_ISPELL_TYPE_SERVER . UDM_ISPELL_TYPE_TEXT est plus lent, et UDM_ISPELL_TYPE_DB est le plus lent. Ce classement est vrai pour mnoGoSearch 3.1.10 − 3.1.11. Il est prévu d'accélérer le mode DB dans les versions futures, et cela sera plus rapide que le mode TEXT.

- UDM_ISPELL_TYPE_DB indique que les données ispell doivent être chargée depuis la base SQL. Dans ce cas, les paramètres val1 et val2 sont ignorés et doivent être laissés vides. flag doit valoir 1 .

Note

flag indique qu'après le chargement des données ispell à partir de la source, elles doivent être triées (c'est nécessaire au bon fonctionnement d'ispell). Dans le cas où vous chargez les données depuis un fichier, il peut y avoir plusieurs appels à udm_load_ispell_data , et il ne vaut pas la peine de trier les valeurs après chaque appel, mais uniquement à la fin. Etant donné qu'en mode DB, toutes les données sont chargées en une seule fois, ce paramètre doit avoir la valeur de 1 . Dans ce mode, en cas d'erreur, par exemple si la table ispell est absente, la fonction retournera FALSE et le code d'erreur, avec son message, seront accessibles avec udm_error et udm_errno .

Exemple avec udm_load_ispell_data

```
 if (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_DB,'','',1)) {
  printf("Erreur #%d: '%s'\n", udm_errno($udm), udm_error($udm));
  exit;
}
```

- UDM_ISPELL_TYPE_AFFIX indique que les données ispell doivent être chargée depuis un fichier et initie le chargement. Dans ce cas, val1 définit le code de langue en deux lettre, et val2 est le chemin jusqu'aux fichiers. Notez que si vous utilisez un chemin relatif, le module recherche les fichiers non pas dans UDM_CONF_DIR , mais directement avec le chemin courant, où le script est exécuté. En cas d'erreur avec ce mode, si le fichier est absent, la fonction retourne FALSE , et un message d'erreur sera affiché. Les messages d'erreur ne sont pas accessibles avec udm_error et udm_errno , puisque ces fonctions ne traitent que les messages SQL. Reportez−vous à la description du paramètre flag .

Exemple avec udm_load_ispell_data et UDM_ISPELL_TYPE_AFFIX

```
 if ((! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_AFFIX,'en','/opt/ispell/en.aff',0)) ||
       (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_AFFIX,'ru','/opt/ispell/ru.aff',0)) ||
       (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_SPELL,'en','/opt/ispell/en.dict',0)) ||
       (! udm_load_ispell_data($udm,UDM_ISPELL_TYPE_SPELL,'ru','/opt/ispell/ru.dict',1))) {
    exit;
    }
```

Note

flag prend la valeur 1 si c'est le dernier appel à cette fonction.

- UDM_ISPELL_TYPE_SPELL indique que les données ispell doivent être chargées depuis un fichier, et initie le chargement du dictionnaire. Dans ce cas, val1 définit le code langue sur deux lettres, et val2 le chemin du fichier. Notez que si vous utilisez un chemin relatif, le module recherche les fichiers non pas dans UDM_CONF_DIR , mais directement avec le chemin courant, où le script est exécuté. En cas d'erreur avec ce mode, si le fichier est absent, la fonction retourne FALSE , et un message d'erreur sera affiché. Les messages d'erreur ne sont pas accessibles avec udm_error et udm_errno , puisque ces fonctions ne traitent que les messages SQL. Reportez−vous à la description du paramètre flag .

Exemple avec udm_load_ispell_data et UDM_ISPELL_TYPE_SPELL

```
 <?php
    if ((! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_AFFIX,'en','/opt/ispell/en.aff',0)) ||
       (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_AFFIX,'ru','/opt/ispell/ru.aff',0)) ||
       (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_SPELL,'en','/opt/ispell/en.dict',0)) ||
       (! Udm_Load_Ispell_Data($udm,UDM_ISPELL_TYPE_SPELL,'ru','/opt/ispell/ru.dict',1))) {
     exit;
     }
?>
```

Note

flag prend la valeur 1 si c'est le dernier appel à cette fonction.

- UDM_ISPELL_TYPE_SERVER active le support des serveurs ispell. val1 indique alors l'adresse de l'hôte qui supporte le serveur ispall. val2 n'est pas encore utilisé, mais dans les cas futurs, il indiquera le numéro de port utilisé par le serveur ispell. flag n'est pas utile, car les données sont déjà triées.

  Les serveurs Spelld lisent les données d'orthographe dans une configuration séparée (par défaut /usr/local/mnogosearch/etc/spelld.conf ), les trie et les stockes en mémoire. Avec les clients, le serveur communique de deux façons : vers les indexeurs, tout le contenu de la mémoire est transféré pour que l'indexeur travaille plus vite; vers le moteur de recherche, il

reçoit les mots à normaliser, et les rend au client corrigés. Cela permet une plus grande rapidité d'exécution, en comparaison des modes db et text (notamment, les tris et les chargements sont beaucoup plus rapides).

udm_load_ispell_data en mode UDM_ISPELL_TYPE_SERVER ne charge pas vraiment les données ispell, mais définit simplement l'adresse du serveur. En fait, le serveur sera automatiquement utilisé par udm_find lors des recherches. En cas d'erreur, (par exemple si le serveur ispell ne fonctionne pas ou que l'hôte indiqué est invalide), la conversion sera annulée, mais aucun message d'erreur ne sera affiché.

| Note |
| --- |
| Cette fonction est disponible à partir de mnoGoSearch 3.1.12. |

Exemple avec udm_load_ispell_data et UDM_ISPELL_TYPE_SERVER

```
<?php
 if (!udm_load_ispell_data($udm,UDM_ISPELL_TYPE_SERVER,'','',1)) {
   printf("Erreur au chargement de la librairie ispell sur le serveur<BR>\n");
   exit;
 }
?>
```

## 6.23.27 udm_open_stored() : Ouvre une connexion MnoGoSearch avec un document stocké

int **udm_open_stored** ( resource agent , string storedaddr )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.23.28 udm_set_agent_param() : Modifie les paramètres de l'agent MnoGoSearch

bool **udm_set_agent_param** ( resource agent , int var , string val )

udm_set_agent_param définit les paramètres de l'agent mnoGoSearch. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Les paramètres suivants et leurs valeurs sont disponibles :

- UDM_PARAM_PAGE_NUM − Utilisé pour choisir le numéro de groupe de résultat (les résultats sont retournés par groupe, commençant à 0, avec UDM_PARAM_PAGE_SIZE résultats par page).
- UDM_PARAM_PAGE_SIZE − Nombre de résultats affichés par page.
- UDM_PARAM_SEARCH_MODE − Mode de recherche. Les valeurs suivantes sont disponibles : UDM_MODE_ALL − recherche tous les mots; UDM_MODE_ANY − recherche l'un des mots; UDM_MODE_PHRASE − recherche une phrase; UDM_MODE_BOOL − recherche booléenne. Voir udm_find pour plus de détails sur les recherches booléennes.
- UDM_PARAM_CACHE_MODE − Active/désactive le cache. Lorsque le cache est activé, le moteur de recherche va stocker les résultats sur le disque. Lorque deux requête seront similaires, il pourra retourner les résultats plus rapidement, sans recherche. Valeurs disponibles : UDM_CACHE_ENABLED , UDM_CACHE_DISABLED .
- UDM_PARAM_TRACK_MODE − Active le mode de suivi de requête. Depuis la version 3.1.2, mnoGoSearch dispose d'un suivi de requête. Notez que ce suivi n'est implémenté qu'avec les versions SQL et n'est pas disponible avec les bases de données intégrées. Pour

utiliser ce suivi, vous devez créer des tables de suivi. Pour mysql, utilisez le script create/mysql/track.txt . Lorsque vous effectuez une recherche avec l'interface, ces tables stockeront les mots recherchés ainsi que le nombre de mots trouvés, et la date. Valeurs disponibles : UDM_TRACK_ENABLED , UDM_TRACK_DISABLED .

- UDM_PARAM_PHRASE_MODE − indique si les index des bases de données utilise des phrases( paramètre "phrase" dans indexer.conf). Valeurs disponibles : UDM_PHRASE_ENABLED and UDM_PHRASE_DISABLED . Notez bien que si la recherche par phrase est activé ( UDM_PHRASE_ENABLED ), il est toujours possible de faire des recherches dans d'autres modes, (ANY, ALL, BOOL ou PHRASE). En version 3.1.10 de mnoGoSearch, la recherche par phrase n'est supportée que pour les modes SQL et intégré, tandis qu'en 3.1.11, la recherche par phrase est supporté par le mode cache.

  Exemple de recherche par phrase :

  "Arizona desert" − Cette requête retourne tous les documents qui contiennent les mots "Arizona desert" comme une phrase. Notez que vous devez mettre des guillemets doubles autour des phrases.
- UDM_PARAM_CHARSET − Définit le jeu de caractères local. Valeurs disponibles : Tous les jeux supportés par mnoGoSearch. koi8−r, cp1251, ...
- UDM_PARAM_STOPFILE − Définit le nom et le chemin du fichier de mots ignorés. Il y a une petite différence avec mnoGoSearch : Avec mnoGoSearch, si le chemin est NULL ou relatif, il est utilisé à partir de UDM_CONF_DIR , alors qu'en PHP, le module va rechercher à partir du chemin courant, c'est−à−dire celui du script courant.
- UDM_PARAM_STOPTABLE − Charge la liste des mots ignorés depuis une table SQL. Vous pouvez utiliser plusieurs tables SQL. Cette commande n'a aucun effet si mnoGoSearch n'a pas été compilé avec le support de base de données.
- UDM_PARAM_WEIGHT_FACTOR − Représente le poids relatif des différentes parties d'un document. Actuellement, le corps, titre, mots clés, descriptions et url sont supportés. Pour activer cette fonctionnalité, utilisez le degré 2 de *Weight commands, dans le fichier indexer.conf . Imaginons que vous avez choisi les poids suivants :

  URLWeight 1

  BodyWeight 2

  TitleWeight 4

  KeywordWeight 8

  DescWeight 16

  Comme l'indexeur utiliser l'opérateur de bits OR pour mesurer le poids des mots, il est possible que le même mot soit trouvé plusieurs fois dans le même document lors des recherches. Un mot qui n'apparaît qu'une fois dans le corps sera défini par 00000010 (notation binaire). Un mot qui apparaîtra dans plusieurs parties pourra avoir la notation 00011111.

  La valeur de ce paramètre est une chaîne de chiffres hexadécimaux, sous la forme ABCDE . Chaque chiffre est un facteur correspondant à un poids affecté à une partie du document. Por la situation décrite ci−dessus,

  est le facteur de poids 1 (URL)

  est le facteur de poids 2 (Corps)

est le facteur de poids 4 (Titre)

est le facteur de poids 8 (Mots clés)

est le facteur de poids 16 (Description)

Exemples:

UDM_PARAM_WEIGHT_FACTOR =00001 ne recherche que dans les URL.

UDM_PARAM_WEIGHT_FACTOR =00100 ne recherche que dans les Titres.

UDM_PARAM_WEIGHT_FACTOR =11100 recherche dans les Titres,Mots−clés,Description mais pas dans le corps ou les URL.

UDM_PARAM_WEIGHT_FACTOR =F9421 recherche dans :

Description avec un poids de 15 (F hex)

Keywords avec un poids de 9

Title avec un poids de 4

Body avec un poids de 2

URL avec un poids de 1

Si UDM_PARAM_WEIGHT_FACTOR est omis, la valeur par défaut est utilisée.

- UDM_PARAM_WORD_MATCH − Recherche des mots. Vous pouvez utiliser ce paramètre pour choisir le type de recherche de mots. Cette fonctionnalité n'est valable qu'en mode "single" et "multi", avec les bases SQL ou intégrée. Elle ne fonctionne pas en mode intégré, ni avec d'autres modes, car les CRC ne supportent pas les recherches de sous−chaînes. Les valeurs disponibles sont :

  UDM_MATCH_BEGIN − début de mot;

  UDM_MATCH_END − fin de mot;

  UDM_MATCH_WORD − tout le mot;

  UDM_MATCH_SUBSTR − une sous−partie de mots.
- UDM_PARAM_MIN_WORD_LEN − définit les tailles extrêmes de mots. Tout mot plus court que la limite inférieur est ignoré. Notez que ce paramètre est inclusif, c'est−à−dire que si UDM_PARAM_MIN_WORD_LEN =3, un mot de 3 caractères ne sera pas ignoré, alors qu'un mot de 2 caractères sera ignoré. Par défaut, la valeur est de 1.
- UDM_PARAM_ISPELL_PREFIXES − Valeurs possibles : UDM_PREFIXES_ENABLED et UDM_PREFIXES_DISABLED . Ces valeurs activent et désactivent le support des préfixes. Par exemple, si le mot "testé" est placé dans la requête de recherche, les mots tels que "test", "tester", etc.. seront aussi recherchés. Les suffixes sont supportés par défaut. Les préfixes modifie généralement le sens des mots. Par exemple, si vous cherchez "testé", vous ne souhaitez pas trouver "protesté" ou "contesté". Le support des préfixes peut cependant être utilisé pour des raisons d'orthographe. Pour activer ispell, vous devez charger les données ispell avec la fonction udm_load_ispell_data .
- UDM_PARAM_CROSS_WORDS − Active ou désactive le support "CROSS_WORDS".

Valeurs possibles : UDM_CROSS_WORDS_ENABLED et UDM_CROSS_WORDS_DISABLED .

La fonctionnalité "CROSS_WORDS" vous permet d'effectuer des recherches dans les balises (entre <a href="xxx"> </a>), pour utiliser le nom du lien. Ce mode fonctionne avec les bases de données SQL et n'est pas supporté par les modes intégrés ou le cache.

| Note |
| --- |
| CROSS_WORDS est supporté à partir de mnoGoSearch 3.1.11. |

- UDM_PARAM_VARDIR − spécifie un chemin spécifique sur le disque où l'indexeur enregistre les données lorsqu'il utilise le cache et les bases de données internes. Par défaut, le dossier /var de l'installation de **MNOGOSEARCH** est utilisé. Ce paramètre est disponible en PHP 4.1.0 et plus récent.

# 6.24 MySQL

## 6.24.1 Introduction

Ces fonctions vous permettent d'accéder aux bases de données MySQL. Le site officiel de cette base est http://www.mysql.com/ .

La documentation de MySQL est disponible à http://www.mysql.com/documentation/ .

## 6.24.2 Pré−requis

Afin de pouvoir les utiliser, vous devez compiler PHP avec le support MySQL.

## 6.24.3 Installation

En utilisant l'option −−with−mysql . Si vous utilisez cette fonction sans préciser le chemin d'accès à la base MySQL, PHP utilisera les librairies clientes MySQL fournies en standard. Les utilisateurs qui font tourner d'autres applications qui utilisent elles−mêmes MySQL (par exemple, PHP 3 et PHP 4 utilisés comme des modules concurrents apache, ou encore auth−mysql), devraient toujours spécifier le chemin jusqu'à MySQL : −−with−mysql=/path/to/mysql . Cela va forcer PHP à utiliser les librairies clientes installées par MySQL et évitera les conflits.

En PHP 5, MySQL n'est plus activé par défaut, et la librairie MysQL n'est plus fournie avec PHP . Lisez la FAQ pour savoir pourquoi.

Cette extension MySQL extension ne fonctionnera pas avec les versions de MySQL supérieure à la version 4.1.0. Pour cela, il faut utiliser l'extension MySQLi .

| Attention |
| --- |
| Des crashes et des problèmes de démarrage de PHP peuvent être rencontré lorsque vous chargez cette fonction en même temps que l'extension recode. Voyez l'extension recode pour plus de détails. |

| Note |
| --- |
| Si vous avez besoin d'autres jeux de caractères que celui par défaut ( **latin** ), vous devez installer la librairie externe libmysql (non fournie), compilée avec ce jeu de caractères. |

## 6.24.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
| --- | --- | --- |
| mysql.allow_persistent | "On" | PHP_INI_SYSTEM |
| mysql.max_persistent | "−1" | PHP_INI_SYSTEM |
| mysql.max_links | "−1" | PHP_INI_SYSTEM |
| mysql.default_port | NULL | PHP_INI_ALL |
| mysql.default_socket | NULL | PHP_INI_ALL |
| mysql.default_host | NULL | PHP_INI_ALL |
| mysql.default_user | NULL | PHP_INI_ALL |
| mysql.default_password | NULL | PHP_INI_ALL |
| mysql.connect_timeout | "0" | PHP_INI_SYSTEM |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*mysql.allow_persistent bool*
: Active ou désactive les connexions persistantes à la base de données MySQL.

*mysql.max_persistent integer*
: Le nombre maximum de connexions persistantes aux bases de données MySQL, par processus.

*mysql.max_links integer*
: Le nombre maximum de connexions aux bases de données MySQL, y compris les connexions persistantes, par processus.

*mysql.default_port string*
: Le numéro de port TCP par défaut, utilisé pour se connecter à la base de données, lorsqu'aucun port n'est spécifié. Si aucun port par défaut est spécifié, le port sera alors obtenu en lisant la variable d'environnement MYSQL_TCP_PORT , l'entrée mysql−tcp dans le fichier /etc/services ou encore la constante de compilation MYSQL_PORT , dans cet order. Win32 n'utilisera aussi que la constante MYSQL_PORT .

*mysql.default_socket string*
: Le nom par défaut de la socket lorsque l'on se conecte au serveur local, si aucune autre socket n'est spécifiée.

*mysql.default_host string*
: Adresse par défaut du serveur, à utiliser lors de la connexion à un serveur MySQL, si aucun hôte n'est spécifié. Ne s'applique par lorsque le safe mode est activé.

*mysql.default_user string*
: Utilisateur par défaut, à utiliser lors de la connexion à un serveur MySQL, si aucun utilisateur n'est spécifié. Ne s'applique par lorsque le safe mode est activé.

*mysql.default_password string*
: Mot de passe par défaut, à utiliser lors de la connexion à un serveur MySQL, si aucun mot de passe n'est spécifié. Ne s'applique par lorsque le safe mode est activé.

*mysql.connect_timeout integer*
: Durée maximale d'attente de la réponse d'un serveur, en secondes. Sous Linux, cette durée sert aussi lors de l'échange du premier avec le serveur.

## 6.24.5 Types de ressources

Il y a deux types de ressources utilisées par le module MySQL. Le premier est un identifiant de connexion au serveur, appelé mysql link , et le second est un identifiant de résultat de requête, appelé mysql result .

## 6.24.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

Depuis PHP 4.3.0, il est possible de spécifier des options de client supplémentaires pour les fonctions mysql_connect et mysql_pconnect . Ces constantes sont les suivantes :

| Constante | Description |
|---|---|
| MYSQL_CLIENT_COMPRESS | Utilise le protocole avec compression |
| MYSQL_CLIENT_IGNORE_SPACE | Autorise les espaces après les noms de fonction |

| | |
|---|---|
| MYSQL_CLIENT_INTERACTIVE | Autorise interactive_timeout secondes d'inactivité sur la connexion (au lieu de wait_timeout). |

La fonction mysql_fetch_array utilise une constante pour spécifier les différents types de formats de réponse. Les constantes suivante sont utilisées :

| Constante | Description |
|---|---|
| MYSQL_ASSOC | Les colonnes sont retournées dans un tableau, dont les index sont les noms de colonnes. |
| MYSQL_BOTH | Les colonnes sont retournées dans un tableau ayant une indexation numérique et un système d'index correspondant au nom des colonnes. |
| MYSQL_NUM | Les colonnes sont retournées dans un tableau avec un index numérique. Les colonnes sont numérotées dans leur ordre d'apparition. L'index commence à zéro. |

## 6.24.7 Exemples

Cet exemple simple montre comment se connecter, exécuter une requête, lire les informations obtenues et se déconnecter d'une base de données MySQL.

Exemple de présentation de l'extension MySQL

```php
<?php
/* Connexion et sélection de la base */
$link = mysql_connect("mysql_host", "mysql_user", "mysql_password")
    or die("Could not connect");
echo "Connexion réussie";
mysql_select_db("my_database") or die("Could not select database");

/* Exécuter des requêtes SQL */
$query = "SELECT * FROM my_table";
$result = mysql_query($query) or die("Query failed");

/* Afficher des résultats en HTML */
echo "<table>\n";
while ($line = mysql_fetch_assoc($result)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

/* Libération des résultats */
mysql_free_result($result);

/* Fermeture de la connexion */
mysql_close($link);
?>
```


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- mysql_affected_rows
- mysql_change_user
- mysql_client_encoding
- mysql_close
- mysql_connect
- mysql_create_db
- mysql_data_seek
- mysql_db_name
- mysql_db_query
- mysql_drop_db
- mysql_errno
- mysql_error
- mysql_escape_string

- mysql_fetch_array
- mysql_fetch_assoc
- mysql_fetch_field
- mysql_fetch_lengths
- mysql_fetch_object
- mysql_fetch_row
- mysql_field_flags
- mysql_field_len
- mysql_field_name
- mysql_field_seek
- mysql_field_table
- mysql_field_type
- mysql_free_result
- mysql_get_client_info
- mysql_get_host_info
- mysql_get_proto_info
- mysql_get_server_info
- mysql_info
- mysql_insert_id
- mysql_list_dbs
- mysql_list_fields
- mysql_list_processes
- mysql_list_tables
- mysql_num_fields
- mysql_num_rows
- mysql_pconnect
- mysql_ping
- mysql_query
- mysql_real_escape_string
- mysql_result
- mysql_select_db
- mysql_stat
- mysql_tablename
- mysql_thread_id
- mysql_unbuffered_query

## 6.24.9 mysql_change_user() : Change le nom de session de l'utilisateur actif

int **mysql_change_user** ( string user , string password , *string database* , *resource link_identifier* )

mysql_change_user change l'utilisateur de la session courante, ou sur la connexion spécifiée avec l'option link_identifier . Si une base est spécifiée, elle deviendra la base par défaut de l'utilisateur. Si une erreur de connexion survient, la connexion en cours restera active.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| mysql_change_user a été introduite en PHP 3.0.13 et requiert MySQL 3.23.3 ou plus récent. |

## 6.24.10 mysql_client_encoding() : Retourne le nom du jeu de caractères utilisé par le client MySQL

string **mysql_client_encoding** ( *resource link_identifier* )

mysql_client_encoding retourne le nom du jeu de caractères utilisé par le client mysql, pour la connexion courante (si omise), ou pour la connexion link_identifier .

Exemple avec mysql_client_encoding

```
<?php
$link = mysql_connect('localhost', 'utilisateur_mysql', 'mot_de_passe');
$charset = mysql_client_encoding($link);
printf ("Le jeu de caractères actuel est %s.\n", $charset);
?>
```

L'exemple ci−dessus va afficher ceci :
Le jeu de caractères actuel est latin1.

Voir aussi mysql_real_escape_string .

## 6.24.11 mysql_close() : Ferme la connexion MySQL

bool **mysql_close** ( *resource link_identifier* )

mysql_close ferme la connexion au serveur MySQL associée à l'identifiant link_identifier . Si cet identifiant n'est pas spécifié, cette commande s'applique à la dernière connexion ouverte.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Note

Notez que cette commande n'est pas nécessaire, car toutes les connexions non persistantes seront automatiquement fermées à la fin du script.

mysql_close ne ferme pas les connexions persistantes générées par mysql_pconnect .

Exemple MySQL_close

```
<?php
 $link = mysql_connect("serveur.mysql.com", "utilisateur", "secret")
    or die("Impossible de se connecter");

 print("Connexion réussie");
 mysql_close($link);
?>
```

Voir aussi mysql_connect et mysql_pconnect .

## 6.24.12 mysql_connect() : Ouvre une connexion à un serveur MySQL

resource **mysql_connect** ( *string server* , *string username* , *string password* , *bool new_link* , *int client_flags* )

mysql_connect retourne une ressource de connexion au serveur MySQL en cas de succès, et sinon FALSE .

mysql_connect établit une connexion à un serveur MySQL. Tous les arguments sont optionnels, et s'ils manquent, les valeurs par défaut sont utilisées ('localhost', nom du propriétaire du processus,

mot de passe vide).

Le nom d'hôte peut aussi inclure un numéro de port, sous la forme : "hostname:port" ou un chemin jusqu'à une socket sous la forme ":/path/to/socket" pour l'hôte localhost.

Note

Le support des ":port" a été ajouté à partir de la version 3.0B4.

Le support de ":/path/to/socket" a été ajouté à partir de la version 3.0.10.

Vous pouvez supprimer le message d'erreur de connexion en ajoutant une arobase '@' au nom de la fonction.

Si un second appel à mysql_connect est fait avec les mêmes arguments, PHP ne va pas ouvrir une nouvelle connexion, mais va retourner l'identifiant de la connexion déjà ouverte. Le paramètre new_link modifie ce comportement et permet que mysql_connect ouvre une nouvelle connexion même si mysql_connect a été appelée auparavant avec les mêmes paramètres. Le paramètre client_flags peut être une combinaison des constantes suivantes : MYSQL_CLIENT_SSL , MYSQL_CLIENT_COMPRESS , MYSQL_CLIENT_IGNORE_SPACE et MYSQL_CLIENT_INTERACTIVE .

Note

La paramètre new_link a été ajouté en PHP 4.2.0.

La paramètre client_flags a été ajouté en PHP 4.3.0

Le lien sera fermé automatiquement dès que l'exécution du script sera terminée, à moins d'être fermé explicitement avec mysql_close .

Exemple avec mysql_connect

```
<?php
   $link = mysql_connect("localhost", "mysql_user", "mysql_password")
       or die("Impossible de se connecter : " . mysql_error());
   print ("Connexion réussie.");
   mysql_close($link);
?>
```

Voir aussi mysql_pconnect et mysql_close .

## 6.24.13 mysql_create_db() : Crée une base de données MySQL

bool **mysql_create_db** ( string database_name , *resource link_identifier* )

mysql_create_db tente de créer une nouvelle base de données nommée database_name sur le serveur associé à l'identifiant link_identifier , ou sur la dernière connexion ouverte.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple de création de base MySQL avec mysql_create_db

```
<?php
 $link = mysql_pconnect ("kron", "jutta", "geheim") {
   or die ("Connexion impossible");
 }
 if (mysql_create_db ("my_db")) {
   print ("Base de données créée\n");
 } else {
   echo "Erreur lors de la création de la base: ".mysql_error();
 }
?>
```

Pour des raisons de compatibilité ascendante, mysql_createdb() est toujours utilisable.

| Note |
| --- |
| La fonction mysql_create_db est obsolète. Il est préférable d'utiliser la fonction mysql_query , avec une commande SQL "CREATE DATABASE". |

Voir aussi mysql_drop_db et mysql_query

## 6.24.14 mysql_data_seek() : Déplace le pointeur interne de résultat

bool **mysql_data_seek** ( resource result_identifier , int row_number )

mysql_data_seek déplace le pointeur interne de résultat, dans le résultat associé à l'identifiant de résultat result_identifier . Il le fait pointer à la ligne row_number . Le prochain appel à mysql_fetch_row retournera cette ligne.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

row_number commence à 0. row_number doit être une valeur qui va de 0 à mysql_num_rows − 1.

| Note |
| --- |
| La fonction mysql_data_seek peut être utilisée en conjonction avec mysql_query , mais pas avec la fonction mysql_unbuffered_query . |

Exemple mysql_data_seek

```
 <?php
  $link = mysql_pconnect ("kron", "jutta", "geheim") {
    or die ("Connexion impossible");
  }
  mysql_select_db ("samp_db") {
    or die ("Sélection de base impossible");
  }
  $query = "SELECT last_name, first_name FROM friends";
  $result = mysql_query ($query) {
    or die ("Requête impossible");
  }
// récupère les lignes dans l'ordre inverse
  for ($i = mysql_num_rows ($result) - 1; $i >=0; $i--) {
    if (!mysql_data_seek ($result, $i)) {
      echo "Impossible d'atteindre la ligne $i\n";
      continue;
    }
    if(!($row = mysql_fetch_object ($result)))
      continue;
    echo "{$row->last_name} {$row->last_name$row->first_name}<BR>\n";
  }
  mysql_free_result ($result);
?>
```

Voir aussi mysql_query et mysql_num_rows .

## 6.24.15 mysql_db_name() : Lit les noms des bases de données

string **mysql_db_name** ( resource result , int row , *mixed field* )

mysql_db_name prend comme premier argument le pointeur de résultat result_identifier , issu de mysql_list_dbs . row est l'index dans le résultat.

Si une erreur survient, FALSE est retourné. Utilisez mysql_errno et mysql_error pour connaître la nature de l'erreur.

Exemple mysql_db_name

```
<?php
 error_reporting(E_ALL);
 mysql_connect('dbhost', 'username', 'password');
 $db_list = mysql_list_dbs();
 $i = 0;
 $cnt = mysql_num_rows($db_list);
 while ($i < $cnt) {
   echo mysql_db_name($db_list, $i) . "\n";
   $i++;
 }
?>
```

Pour des raisons de compatibilité ascendante, mysql_dbname() est aussi accepté, mais obsolète.

## 6.24.16 mysql_db_query() : Envoie une requête MySQL à un serveur MySQL

resource **mysql_db_query** ( string database , string query , *resource link_identifier* )

mysql_db_query retourne un identifiant de résultat si la requête réussit et FALSE sinon.

mysql_db_query sélectionne une base de données et exécute une requête. Si l'identifiant de lien link_identifier n'est pas précisé, mysql_db_query prendra par défaut la dernière connexion ouverte sur le serveur et si elle n'en trouve pas, elle tentera de se connecter, en utilisant la fonction mysql_connect , sans arguments.

Voir aussi mysql_connect et mysql_query .

Note

Cette fonction est obsolète, et abandonnée depuis PHP 4.0.6. Ne l'utilisez plus (ou pas!). Alternative recommandée : mysql_select_db et mysql_query .

## 6.24.17 mysql_drop_db() : Efface une base de données MySQL

bool **mysql_drop_db** ( string database_name , *resource link_identifier* )

mysql_drop_db essaie d'effacer une base de données entière sur le serveur associé à la ressource de connexion link_identifier .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Pour des raisons de compatibilité ascendante, mysql_dropdb() peut encore être utilisée. Son usage est déconseillé.à

Note

La fonction mysql_drop_db est obsolète. Il est préférable d'utiliser mysql_query avec la requête SQL DROP DATABASE à la place.

Voir aussi mysql_create_db et mysql_query .

## 6.24.18 mysql_errno() : Retourne le numéro d'erreur de la dernière commande MySQL

int **mysql_errno** ( *resource link_identifier* )

mysql_errno retourne le numéro de message d'erreur de la dernière opération MySQL sur la connexion courante, ou sur la connexion spécifiée avec l'option link_identifier . Les erreurs qui sont remontées depuis le serveur MySQL ne sont plus des alertes. A la place, il faut utiliser mysql_errno pour obtenir le numéro d'erreur.

Exemple avec mysql_errno

```
<?php
 mysql_connect("marliesle");
 echo mysql_errno()." : ".mysql_error()."<BR>";
 mysql_select_db("nonexistentdb");
 echo mysql_errno()." : ".mysql_error()."<BR>";
 $conn = mysql_query("SELECT * FROM nonexistenttable");
 echo mysql_errno()." : ".mysql_error()."<BR>";
?>
```

The above example would produce the following output:

```
1049: Unknown database 'nonexistentdb'
1146: Table 'kossu.nonexistenttable' doesn't exist
```

Voir aussi mysql_error .

## 6.24.19 mysql_error() : Retourne le texte associé avec l'erreur générée lors de la dernière requête.

string **mysql_error** ( *resource link_identifier* )

mysql_error retourne le message d'erreur généré par la dernière commande MySQL, ou bien '' (la chaîne vide) si aucune erreur n'est survenue.

Les erreurs retournées par le serveur MySQL ne génèrent plus de message d'alerte. A la place, vous devez utiliser la fonction mysql_error pour lire le contenu du message. Notez que cette fonction ne retourne que le texte de l'erreur la plus récente, ce qui fait que si vous souhaitez l'utiliser, vous devez vous assurer de sa valeur avant de lancer une autre requête.

Exemple avec mysql_error

```
<?php
   mysql_connect("localhost", "mysql_user", "mysql_password");

   mysql_select_db("nonexistentdb");
   echo mysql_errno() . ": " . mysql_error(). "\n";

   mysql_select_db("kossu");
   mysql_query("SELECT * FROM nonexistenttable");
   echo mysql_errno() . ": " . mysql_error() . "\n";
?>
```

L'exemple ci−dessus va afficher :

```
1049: Unknown database 'nonexistentdb'
1146: Table 'kossu.nonexistenttable' doesn't exist
```

Note

Si le paramètre optionnel link_identifier est utilisé, c'est le dernièr message d'erreur sur cette connexion qui sera lue. Sinon, c'est la dernière connexion MySQL ouverte qui est utilisée.

Voir aussi mysql_errno .

## 6.24.20 mysql_escape_string() : Protège une chaîne pour la passer à mysql_query()

string **mysql_escape_string** ( string unescaped_string )

mysql_escape_string va protéger tous les caractères de la chaîne unescaped_string , pour pouvoir l'utiliser directement dans une requête mysql_query . Elle retourne la chaîne modifiée.

| Note |
| --- |
| mysql_escape_string n'échappe pas les caractères pourcentage % ou _ . |

| Note |
| --- |
| mysql_escape_string est identique à la fonction mysql_real_escape_string hormi le fait que mysql_real_escape_string requiert une ressource de connexion et protège la chaîne en fonction du jeu de caractères courant. mysql_escape_string ne demande pas de connexion comme argument, et ne resepecte pas le jeu de caractères courant. |

Exemple avec mysql_escape_string

```
<?php
   $item = "C'est un exemple de Zak";
   $escaped_item = mysql_escape_string($item);
   printf ("Chaîne protégée : %s\n", $escaped_item);
?>
```

L'exemple ci−dessus affichera :
Chaîne protégée : C\'est un exemple de Zak

Voir aussi mysql_real_escape_string , addslashes et les magic_quotes_gpc directive.

## 6.24.21 mysql_fetch_array() : Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux

array **mysql_fetch_array** ( resource result , *int result_type* )

mysql_fetch_array retourne un tableau qui contient la ligne demandée, ou FALSE si il ne reste plus de ligne.

mysql_fetch_array est une version étendue de mysql_fetch_row . En plus d'enregistrer les données sous forme d'un tableau à indice numérique, elle peut aussi les enregistrer dans un tableau associatif, en utilisant les noms des champs comme indices.

Si plusieurs colonnes portent le même nom, la dernière colonne aura la priorité. Pour accéder aux autres colonnes du même nom, vous devez utiliser l'index numérique, ou faire un alias pour chaque colonne.

Requête avec duplication de nom de colonnes

```
select table1.field as foo, table2.field as bar from table1, table2
```

Il est important de souligner que mysql_fetch_array N'est PAS plus lente que mysql_fetch_row , tandis qu'elle ajoute un confort d'utilisation notable.

L'option result_type de mysql_fetch_array est une constante qui peut prendre les valeurs suivantes : MYSQL_ASSOC , MYSQL_NUM et MYSQL_BOTH .Cette fonctionnalité a été ajoutée en PHP 3.0.7. MYSQL_BOTH est la valeur par défaut de cet argument.

En utilisant MYSQL_BOTH , vos obtiendrez un tableau qui sera à la fois associatif et indexé numériquement. En utilisant MYSQL_ASSOC , vous obtiendrez un tableau associatif (tout comme le fait mysql_fetch_assoc ), et en utilisant MYSQL_NUM , vous obtiendrez un tableau indexé nuériquement (comme le fait mysql_fetch_row ).

| Note |
| --- |
| Les noms des champs retournés par cette fonction sont **sensibles à la casse** . |

Exemple de mysql_fetch_array avec MYSQL_NUM

```
<?php
   mysql_connect("localhost", "mysql_user", "mysql_password") or
       die("Impossible de se connecter :  " . mysql_error());
   mysql_select_db("mydb");

   $result = mysql_query("SELECT id, name FROM mytable");

   while ($row = mysql_fetch_array($result, MYSQL_NUM)) {
       printf ("ID: %s  Name: %s", $row[0], $row[1]);
   }

   mysql_free_result($result);
?>
```

Exemple de mysql_fetch_array avec MYSQL_ASSOC

```
<?php
   mysql_connect("localhost", "mysql_user", "mysql_password") or
       die("Impossible de se connecter :  " . mysql_error());
   mysql_select_db("mydb");

   $result = mysql_query("SELECT id, name FROM mytable");

   while ($row = mysql_fetch_array($result, MYSQL_ASSOC)) {
       printf ("ID: %s  Name: %s", $row["id"], $row["name"]);
   }

   mysql_free_result($result);
?>
```

Exemple de mysql_fetch_array avec MYSQL_BOTH

```
<?php
   mysql_connect("localhost", "mysql_user", "mysql_password") or
       die("Impossible de se connecter :  " . mysql_error());
   mysql_select_db("mydb");

   $result = mysql_query("SELECT id, name FROM mytable");

   while ($row = mysql_fetch_array($result, MYSQL_BOTH)) {
       printf ("ID: %s  Name: %s", $row[0], $row["name"]);
   }

   mysql_free_result($result);
?>
```

Voir aussi mysql_fetch_row et mysql_fetch_assoc .

## 6.24.22 mysql_fetch_assoc() : Lit une ligne de résultats dans un tableau associatif

array **mysql_fetch_assoc** ( resource result )

mysql_fetch_assoc retourne un tableau associatif qui contient la ligne lue dans le résultat result , ou bien FALSE , s'il ne reste plus de lignes à lire.

mysql_fetch_assoc est équivalente à mysql_fetch_array utilisée avec l'option MYSQL_ASSOC. Elle ne retourne qu'un tableau associatif. C'est le fonctionnement original de mysql_fetch_array . Si vous avez besoin d'indices numériques, utilisez mysql_fetch_array .

Si plusieurs colonnes portent le même nom, la dernière aura la priorité. Pour accéder aux autres colonnes du même nom, vous devez utiliser mysql_fetch_array et les indices numériques.

Une chose importante à noter est que mysql_fetch_assoc n'est PAS significativement plus lente que mysql_fetch_row , alors qu'elle apporte un confort d'utilisation important.

Pour plus de détails, reportez−vous à mysql_fetch_row et mysql_fetch_array .

| Note |
| --- |
| Les noms des champs retournés par cette fonction sont **sensibles à la casse** . |

Exemple avec mysql_fetch_assoc

```
<?php
   $conn = mysql_connect("localhost", "mysql_user", "mysql_password");

   if (!$conn) {
       echo "Impossible de se connecter à la base de données : " . mysql_error();
       exit;
   }

   if (!mysql_select_db("mydbname")) {
       echo "Impossible d'accéder à la base de données  : " . mysql_error();
       exit;
   }

   $sql = "SELECT id as userid, fullname, userstatus
            FROM   sometable
           WHERE  userstatus = 1";

   $result = mysql_query($sql);

   if (!$result) {
       echo "Impossible d'exécuter la requête ($sql) dans la base : " . mysql_error();
       exit;
   }

   if (mysql_num_rows($result) == 0) {
       echo "Aucune ligne trouvées, rien à afficher.";
       exit;
   }

   // Tant qu'une ligne existe, place cette ligne dans la variable $row
   // sous la forme d'un tableau associatif.
   // Note : Si vous n'attendez qu'une seule ligne, oubliez la boucle
   // Note : Si vous utilisez extract($row); dans la boucle suivante
   //       vous créerez then create $userid, $fullname, and $userstatus
   while ($row = mysql_fetch_assoc($result)) {
       echo $row["userid"];
       echo $row["fullname"];
       echo $row["userstatus"];
   }

   mysql_free_result($result);

?>
```

Voir aussi mysql_fetch_row , mysql_fetch_array , mysql_query et mysql_error .

## 6.24.23 mysql_fetch_field() : Retourne les données enregistrées dans une colonne sous forme d'objet.

object **mysql_fetch_field** ( resource result , *int* *field_offset* )

mysql_fetch_field retourne un objet contenant les données, lu dans le résultat result .

mysql_fetch_field sert à obtenir des informations à propos des champs, dans certaines requêtes. Si l'indice du champ n'est pas spécifié, le champ suivant le dernier champ retourné, est retourné.

Les propriétés de l'objet sont :

- name − nom de la colonne
- table − nom de la table de la colonne
- max_length − taille maximale de la colonne
- not_null − 1 si la colonne ne peut pas être NULL (attribut NOT NULL)

- primary_key − 1 si la colonne est une clé primaire (attribut PRIMARY KEY)
- unique_key − 1 si la colonne est une clé unique (attribut UNIQUE)
- multiple_key − 1 si la colonne est une clé non−unique
- numeric − 1 si la colonne est numérique
- blob − 1 si la colonne est BLOB
- type − le type de la colonne
- unsigned − 1 si la colonne est non signée
- zerofill − 1 si la colonne est complétée par des zéros.

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse** .

Exemple avec mysql_fetch_field

```
 <?php
mysql_connect('localhost:3306', $user, $password)
    or die("Impossible de se connecter : " . mysql_error());
mysql_select_db("database");
$result = mysql_query("select * from table")
    or die("Query failed: " . mysql_error());
/* Lecture des méta données de la colonne */
$i = 0;
while ($i < mysql_num_fields($result)) {
    echo "Détails sur la colonne $i :<br />\n";
    $meta = mysql_fetch_field($result, $i);
    if (!$meta) {
        echo "Aucun détail disponible<br />\n";
    }
    echo "<pre>
blob:         $meta->blob
max_length:   $meta->max_length
multiple_key: $meta->multiple_key
name:         $meta->name
not_null:     $meta->not_null
numeric:      $meta->numeric
primary_key:  $meta->primary_key
table:        $meta->table
type:         $meta->type
unique_key:   $meta->unique_key
unsigned:     $meta->unsigned
zerofill:     $meta->zerofill
</pre>";
    $i++;
}
mysql_free_result($result);
?>
```

Voir aussi mysql_field_seek .

## 6.24.24 mysql_fetch_lengths() : Retourne la taille de chaque colonne d'une ligne de résultat

array **mysql_fetch_lengths** ( resource result )

mysql_fetch_lengths retourne un tableau avec la taille de chaque colonne de la dernière ligne retournée par mysql_fetch_row , et lue dans le résultat result , sinon FALSE .

mysql_fetch_lengths stocke les tailles de chaque colonne de la dernière ligne retournée par mysql_fetch_row , mysql_fetch_array et mysql_fetch_object dans un tableau, en commençant à la position 0.

Voir aussi mysql_fetch_row .

## 6.24.25 mysql_fetch_object() : Retourne une ligne de résultat MySQL sous la forme d'un objet

object **mysql_fetch_object** ( resource result )

mysql_fetch_object retourne un objet dont les propriétés correspondent à une ligne du résultat result , ou FALSE si il n'y a plus d'autres lignes.

mysql_fetch_object est identique à mysql_fetch_array , à la différence qu'elle retourne un objet à la place d'un tableau. Vous pourrez ainsi accéder aux valeurs des champs par leur nom, mais plus par leur offset (les nombres ne sont pas des noms MySQL).

```
<?php

/* Ceci est valide */
echo $row->field;
/*  Ceci est invalide */
echo $row->0;

?>
```

| Note |
| --- |
| Les noms des champs retournés par cette fonction sont **sensibles à la casse** . |

L'argument optionnel result_type est une constante qui peut prendre les valeurs suivantes : MYSQL_ASSOC, MYSQL_NUM et MYSQL_BOTH.

Concernant la vitesse, mysql_fetch_object est aussi rapide que mysql_fetch_array et presque aussi rapide que mysql_fetch_row (la différence est insignifiante)

Exemple avec mysql_fetch_object

```
<?php
mysql_connect("hostname", "user", "password");
mysql_select_db($db);
$result = mysql_query("select * from table");
while ($row = mysql_fetch_object($result)) {
    echo $row->user_id;
    echo $row->fullname;
}
mysql_free_result($result);
?>
```

Voir aussi mysql_fetch_assoc , mysql_fetch_array et mysql_fetch_row .

## 6.24.26 mysql_fetch_row() : Retourne une ligne de résultat MySQL sous la forme d'un tableau

array **mysql_fetch_row** ( resource result )

mysql_fetch_row retourne un tableau indexé numériquement, qui correspond à la prochaine ligne du résultat MySQL result , ou bien FALSE s'il ne reste plus de ligne.

| Note |
| --- |
| Les noms des champs retournés par cette fonction sont **sensibles à la casse** . |

mysql_fetch_row va rechercher une ligne dans le résultat associé à l'identifiant de résultat spécifié. La ligne est retournée sous la forme d'un tableau. Chaque colonne est enregistrée sous la forme d'un tableau commençant à la position 0 .

Les appels suivants à mysql_fetch_row retourneront la ligne suivante dans le résultat, ou FALSE si il n'y a plus de ligne disponible.

Voir aussi mysql_fetch_assoc , mysql_fetch_array , mysql_fetch_object , mysql_data_seek , mysql_fetch_lengths et mysql_result .

## 6.24.27 mysql_field_flags() : Retourne des détails sur une colonne MySQL

string **mysql_field_flags** ( resource result , int field_offset )

mysql_field_flags retourne le sémaphore associé au champ spécifié par field_offset . Les sémaphores sont retournés comme des mots, séparés par des espaces, ce qui les rend faciles à séparer, avec la commande explode .

Les valeurs suivantes (pour une version suffisamment récente de MySQL) sont disponibles : "not_null", "primary_key", "unique_key", "multiple_key", "blob", "unsigned", "zerofill", "binary", "enum", "auto_increment", "timestamp".

Pour des raisons de compatibilité ascendante, mysql_fieldflags() peut encore être utilisé.

## 6.24.28 mysql_field_len() : Retourne la taille d'un champ de résultat MySQL

int **mysql_field_len** ( resource result , int field_offset )

mysql_field_len retourne la taille du champ field_offset −ième champ dans le résultat result . Les champs sont indexés à partir de 0.

Pour des raisons de compatibilité ascendante, mysql_fieldlen() peut encore être utilisé.

## 6.24.29 mysql_field_name() : Retourne le nom d'une colonne

string **mysql_field_name** ( resource result , int field_index )

mysql_field_name retourne le nom de la field_index −ième colonne du résulat MySQL result . result doit être une ressource de résultat MySQL et field_index est un offset numérique du champ.

Note

field_index commence à 0.

e.g. L'index du troisième champ sera en fait l'offset 2, et l'index du quatrième champ sera en fait l'offset 3, etc....

Note

Les noms des champs retournés par cette fonction sont **sensibles à la casse** .

Exemple avec mysql_field_name

```
 /* Supposons que la table utilisée contienne trois champs :
 *   user_id
 *   username
 *   password.
 */
$link = mysql_connect('localhost', "mysql_user", "mysql_password");
mysql_select_db($dbname, $link)
    or die("Impossible de se connecter à la base $dbname : " . mysql_error());
$res = mysql_query("select * from users", $link);

echo mysql_field_name($res, 0) . "\n";
echo mysql_field_name($res, 2);
```

L'exemple ci−dessus va afficher :
user_id
password

Pour des raisons de compatibilité ascendante, mysql_fieldname() peut encore être utilisé.

## 6.24.30 mysql_field_seek() : Déplace le pointeur de résultat

int **mysql_field_seek** ( resource result , int field_offset )

Place le pointeur de résultat sur le champ spécifié. Lors du prochain appel à mysql_fetch_field qui n'aura pas d'argument d'index de champ, le champ désormais pointé sera retourné.

Voir aussi mysql_fetch_field .

## 6.24.31 mysql_field_table() : Retourne le nom de la table où se trouve une colonne

string **mysql_field_table** ( resource result , int field_offset )

mysql_field_table retourne le nom de la table où se trouve la colonne d'offset field_offset , dans le résultat MySQL result .

Pour des raisons de compatibilité ascendante, mysql_fieldtable() peut encore être utilisé. Toutefois, elle est obsolète.

## 6.24.32 mysql_field_type() : Retourne le type de la colonne courante

string **mysql_field_type** ( resource result , int field_offset )

mysql_field_type est similaire à la fonction mysql_field_name . Les arguments sont identiques : c'est le type du champ qui est retourné. Il vaudra "int", "real", "string", "blob" ou d'autres, comme détaillé dans la documentation MySQL .

Types MySQL

```
<?php
   mysql_connect("localhost", "mysql_utilisateur", "mysql_mot_de_passe");
   mysql_select_db("mysql");
   $result = mysql_query("SELECT * FROM func");
   $fields = mysql_num_fields($result);
   $rows   = mysql_num_rows($result);
   $table = mysql_field_table($result, 0);
   echo "Votre table '".$table."' dispose de ".$fields." colonnes et ".$rows." ligne(s)\n";
   echo "Les colonnes sont les suivantes :  \n";
   for ($i=0; $i < $fields; $i++) {
       $type  = mysql_field_type($result, $i);
       $name  = mysql_field_name($result, $i);
       $len   = mysql_field_len($result, $i);
       $flags = mysql_field_flags($result, $i);
       echo $type." ".$name." ".$len." ".$flags."\n";
   }
   mysql_free_result($result);
   mysql_close();
?>
```

L'exemple ci−dessus produit l'affichage suivant :
Votre table 'func' dispose de 4 colonnes et 1 ligne(s)
Les colonnes sont les suivantes :

string name 64 not_null primary_key binary
int ret 1 not_null
string dl 128 not_null
string type 9 not_null enum

Pour des raisons de compatibilité ascendante, mysql_fieldtype() peut encore être utilisé. Toutefois, elle est obsolète.

## 6.24.33 mysql_free_result() : Efface le résultat de la mémoire

bool **mysql_free_result** ( resource result )

mysql_free_result libère toute la mémoire et les ressources utilisées par la ressource de résultat result .

mysql_free_result n'est à appeler que si vous avez peur d'utiliser trop de mémoire durant l'exécution de votre script. Toute la mémoire associée à l'identifiant de résultat sera automatiquement libérée.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Pour des raisons de compatibilité ascendante, mysql_freeresult() peut encore être utilisé.

## 6.24.34 mysql_get_client_info() : Lit les informations sur le client MySQL

string **mysql_get_client_info** ( void )

mysql_get_client_info retourne une chaîne qui représente le numéro de version du client utilisé par PHP.

Exemple avec mysql_get_client_info

```
<?php
   printf ("Détails sur le client MySQL : %s\n", mysql_get_client_info());
?>
```

L'exemple ci−dessus pourrait afficher par exemple :

```
Détails sur le client MySQL : 3.23.39
```

Voir aussi mysql_get_host_info , mysql_get_proto_info et mysql_get_server_info .

## 6.24.35 mysql_get_host_info() : Lit les informations sur l'hôte MySQL

string **mysql_get_host_info** ( *resource link_identifier* )

mysql_get_host_info retourne une chaîne qui représente le type de connexion utilisé avec la connexion link_identifier , y compris le nom du serveur hôte. Si link_identifier est omis, la dernière connexion ouverte est utilisée.

Exemple avec mysql_get_host_info

```
<?php
   mysql_connect("hote.mysql", "utilisateur", "mot de passe") or
       die("Imopssible de se connecter : " . mysql_error());
   printf ("Informations sur l'hôte MySQL : %s\n", mysql_get_host_info());
?>
```

L'exemple ci−dessus va afficher (par exemple) :

```
Informations sur l'hôte MySQL : Localhost via UNIX socket
```

Voir aussi mysql_get_client_info , mysql_get_proto_info et mysql_get_server_info .

## 6.24.36 mysql_get_proto_info() : Lit les informations sur le protocole MySQL

int **mysql_get_proto_info** ( *resource link_identifier* )

mysql_get_proto_info retourne une chaîne qui représente la version du protocole utilisé par la connexion link_identifier . Si link_identifier est omis, la dernière connexion ouverte est utilisée.

Exemple avec mysql_get_proto_info

```php
<?php
   mysql_connect("localhost", "mysql_user", "mysql_password") or
       die("Impossible de se connecter : " . mysql_error());
   printf ("Version du protocole MySQL : %s\n", mysql_get_proto_info());
?>
```

L'exemple ci−dessus pourrait afficher :

```
Version du protocole MySQL : 10
```

Voir aussi mysql_get_client_info , mysql_get_host_info et mysql_get_server_info .

## 6.24.37 mysql_get_server_info() : Lit les informations sur le serveur MySQL

string **mysql_get_server_info** ( *resource link_identifier* )

mysql_get_server_info retourne une chaîne qui représente la version du serveur, dont la connexion est Si link_identifier est omis, la dernière connexion ouverte est utilisée.

Exemple avec mysql_get_server_info

```php
<?php
   mysql_connect("localhost", "mysql_user", "mysql_password") or
       die("Could not connect: " . mysql_error());
   printf ("MySQL server version: %s\n", mysql_get_server_info());
?>
```

L'exemple ci−dessus va afficher (par exemple) :

```
MySQL server version: 4.0.1−alpha
```

Voir aussi mysql_get_client_info , mysql_get_host_info et mysql_get_proto_info .

## 6.24.38 mysql_info() : Lit des informations à propos de la dernière requête

string **mysql_info** ( *resource link_identifier* )

mysql_info retourne des informations détaillées sur la dernière requête exécutée sur la dernière connexion ouverte (si omise), ou sur la connexion link_identifier .

mysql_info retourne une chaîne avec tous les détails listés ci−dessous. Pour toutes les autres commandes, elle retourne FALSE . Le format de la chaîne dépend de la commande.

Commande utilisables avec mysql_info

```
 INSERT INTO ... SELECT ...
String format: Records: 23 Duplicates: 0 Warnings: 0
 INSERT INTO ... VALUES (...),(...),(...)...
String format: Records: 37 Duplicates: 0 Warnings: 0
 LOAD DATA INFILE ...
String format: Records: 42 Deleted: 0 Skipped: 0 Warnings: 0
 ALTER TABLE
String format: Records: 60 Duplicates: 0 Warnings: 0
 UPDATE
String format: Rows matched: 65 Changed: 65 Warnings: 0
```

Les nombres ci−dessus ne sont là que pour illustrer. Leur valeur s'adaptera à la situation de la requête.

Note

mysql_info retourne une valeur différente de FALSE pour les commandes INSERT ... VALUES uniquement si plusieurs jeux de valeurs à insérer ont été spécifiés dans la commande.

Voir aussi mysql_affected_rows .

## 6.24.39 mysql_insert_id() : Retourne l'identifiant généré par la dernière requête INSERT.

int **mysql_insert_id** ( *resource link_identifier* )

mysql_insert_id retourne le dernier identifiant généré par un champ de type AUTO_INCREMENT, sur la connexion MySQL courante, ou bien sûr la connexion spécifiée par link_identifier . Si link_identifier est omis, la dernière connexion ouverte est utilisée.

mysql_insert_id retourne 0 si la dernière requête SQL n'a pas généré de valeur AUTO_INCREMENT. Si vous devez lire cette valeur pour la conserver et l'utiliser ultérieurement, appelez la fonction mysql_insert_id juste après l'exécution de la requête, pour obtenir cette valeur non parasitée.

Note

La fonction MySQL LAST_INSERT_ID() contient toujours la valeur AUTO_INCREMENT la plus récente, et n'est pas remis à zéro entre deux requêtes.

Attention

mysql_insert_id convertit le type de valeur retourné par la fonction C de MySQL C mysql_insert_id() en type PHP long (appelé int en PHP). Si votre colonne de type AUTO_INCREMENT est une colonne de type BIGINT, la valeur retournée par mysql_insert_id sera incorrecte. A la place, utilisez la fonction interne MySQL LAST_INSERT_ID() dans une requête SQL.

Exemple avec mysql_insert_id

```php
<?php
   mysql_connect("localhost", "mysql_user", "mysql_password") or
       die("Impossible de se connecter : " . mysql_error());
   mysql_select_db("mydb");

   mysql_query("INSERT INTO mytable (product) values ('kossu')");
   printf ("Le dernier ID inséré dans est le %d\n", mysql_insert_id());
?>
```

Voir aussi mysql_query .

## 6.24.40 mysql_list_dbs() : Liste les bases de données disponibles sur le serveur MySQL

resource **mysql_list_dbs** ( *resource link_identifier* )

mysql_list_dbs retournera un identifiant de résultat, qui contiendra les noms des bases de données disponibles sur la connexion MySQL courante, ou bien sur la connexion spécifiée par link_identifier . La fonction mysql_tablename , ainsi que toutes les fonctions d'accès aux résultats comme mysql_fetch_array peuvent être utilisées pour déterminer le nom des bases dans le pointeur de résultat.

Exemple avec mysql_list_dbs

```
<?php
$link = mysql_connect('localhost', 'utilisateur_mysql', 'mot_de_passe_mysql');
$db_list = mysql_list_dbs($link);

while ($row = mysql_fetch_object($db_list)) {
    echo $row->Database . "\n";
}
?>
```

L'exemple ci−dessus va afficher :
database1
database2
database3
...

Note

The above code would just as easily work with mysql_fetch_row or other similar functions.

Pour des raisons de compatibilité ascendante, mysql_listdbs() est encore disponible.

Voir aussi mysql_db_name .

## 6.24.41 mysql_list_fields() : Liste les champs du résultat MySQL

resource **mysql_list_fields** ( string database_name , string table_name , *resource link_identifier* )

mysql_list_fields recherche les informations relatives à la table table_name sur la connexion MySQL courante, ou bien sûr la connexion spécifiée par link_identifier . Les arguments sont la base de données et le nom de la table. Un pointeur de résultat est retourné et pourra être passé à mysql_field_flags , mysql_field_len , mysql_field_name et mysql_field_type .

Un identifiant de résultat est un entier positif. La fonction retourne −1 si une erreur survient. Une chaîne décrivant le problème rencontré sera placée dans la variable $phperrmsg et, à moins que la fonction n'ait été appelée sous la forme @mysql() , cette erreur sera aussi affichée.

Exemple avec mysql_list_fields

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');

$fields = mysql_list_fields("database1", "table1", $link);
$columns = mysql_num_fields($fields);

for ($i = 0; $i < $columns; $i++) {
```

```
    echo mysql_field_name($fields, $i) . "\n";
}
```

Le script ci−dessus va afficher :
field1
field2
field3
...

Pour des raisons de compatibilité ascendante, mysql_listfields() est encore disponible.

## 6.24.42 mysql_list_processes() : Liste les processus MySQL

resource **mysql_list_processes** ( *resource link_identifier* )

mysql_list_processes retourne une ressource de résultat qui liste les processus courants sur le serveur.

Exemple avec mysql_list_processes

```
<?php
$link = mysql_connect('localhost', 'mysql_user', 'mysql_password');

$result = mysql_list_processes($link);
while ($row = mysql_fetch_row($result)){
    printf("%s %s %s %s %s\n", $row["Id"], $row["Host"], $row["db"],
       $row["Command"], $row["Time"]);
}
mysql_free_result ($result);
?>
```

L'exemple ci−dessus va afficher :
1 localhost test Processlist 0
4 localhost mysql sleep 5

Voir aussi mysql_thread_id .

## 6.24.43 mysql_list_tables() : Liste les tables d'une base de données

resource **mysql_list_tables** ( string database , *resource link_identifier* )

mysql_list_tables prend le nom d'une base de données database et retourne un identifiant de résultat, qui contiendra la liste des tables sur la connexion MySQL courante, ou bien sûr la connexion spécifiée par link_identifier . La fonction mysql_tablename , ainsi que toutes les fonctions d'accès aux résultats comme mysql_fetch_array peuvent être utilisées pour déterminer le nom des tables dans le pointeur de résultat.

Le paramètre database est le nom de la base de données à utiliser pour lister les tables. En cas d'échec, mysql_list_tables retourne FALSE .

Pour des raisons de compatibilité ascendante, la fonction mysql_listtables() est toujours disponible. Elle est désormais obsolète.

Exemple avec mysql_list_tables

```
<?php
   $dbname = 'test';

   if (!mysql_connect('serveur_mysql', 'utilisateur_mysql', 'mot_de_passe_mysql') {
       print 'Impossible de se connecter à MySQL';
       exit;
   }
```

```
    $result = mysql_list_tables($dbname);

    if (!$result) {
        print "Erreur : impossible de lister les bases de données\n";
        print 'Erreur MySQL : ' . mysql_error();
        exit;
    }

    while ($row = mysql_fetch_row($result)) {
        print "Table : $row[0]\n";
    }

    mysql_free_result($result);
?>
```

Voir aussi mysql_list_dbs et mysql_tablename .

## 6.24.44 mysql_num_fields() : Retourne le nombre de champs d'un résultat MySQL

int **mysql_num_fields** ( resource result )

mysql_num_fields retourne le nombre de champs du résultat MySQL result .

Voir aussi mysql_select_db , mysql_query , mysql_fetch_field et mysql_num_rows .

Pour des raisons de compatibilité ascendante mysql_numfields() est encore disponible.

## 6.24.45 mysql_num_rows() : Retourne le nombre de lignes d'un résultat

int **mysql_num_rows** ( resource result )

mysql_num_rows retourne le nombre de lignes d'un résultat. Cette commande n'est valide que pour les commandes SELECT . Pour connaître le nombre de lignes retournées par INSERT, UPDATE ou DELETE, utilisez mysql_affected_rows .

Exemple mysql_num_rows

```
 <?php

$link = mysql_connect("localhost", "mysql_user", "mysql_password");
mysql_select_db("database", $link);

$result = mysql_query("SELECT * FROM table1", $link);
$num_rows = mysql_num_rows($result);

echo "$num_rows Rows\n";

?>
```

Note

Si vous utilisez mysql_unbuffered_query , mysql_num_rows ne retournera pas le nombre correct de lignes lues tant que vous n'aurez pas lu toutes les lignes de la base.

Voir aussi mysql_affected_rows , mysql_connect , mysql_data_seek , mysql_select_db et mysql_query .

Pour des raisons de compatibilité ascendante mysql_numrows() est encore disponible.

## 6.24.46 mysql_pconnect() : Ouvre une connexion persistante à un serveur MySQL.

resource **mysql_pconnect** ( *string server* , *string username* , *string password* , *int client_flags* )

mysql_pconnect retourne un lien persistant positif en cas de succès et sinon FALSE en cas d'erreur.

mysql_pconnect établit une connexion persistante à un serveur MySQL. Tous les arguments sont optionnels et des valeurs par défaut seront utilisér en cas d'omission (' localhost ', nom d'utilisateur propriétaire du processus, mot de passe vide).

Le nom de l'hôte peut aussi inclure le numéro de port, c'est−à−dire "hostname:port" ou un chemin jusqu'à la socket :/path/to/socket pour l'hôte local.

| Note |
| --- |
| Le support de ":port" a été ajouté à partir de la version 3.0B4.<br><br>Le support de ":/path/to/socket" a été ajouté à partir de la version 3.0.10. |

mysql_pconnect se comporte exactement comme mysql_connect , mais avec deux différences majeures :

Premièrement, lors de la connexion, la fonction essaie de trouver une connexion permanente déjà ouverte sur cet hôte, avec le même nom d'utilisateur et de mot de passe. Si une telle connexion est trouvée, son identifiant est retourné, sans ouvrir de nouvelle connexion.

Deuxièmement, la connexion au serveur MySQL ne sera pas terminée avec la fin du script. Au lieu de cela, le lien sera conservé pour un prochain accès ( mysql_close ne terminera pas une connexion persistante établie par mysql_pconnect ).

C'est pourquoi ce type de connexion est dite 'persistante'.

| Note |
| --- |
| Notez que les connexions persitantes ne fonctionnent que si vous utilisez PHP en version module. Voyez la section sur les connexions persistantes aux bases de données pour plus d'informations. |

| Attention |
| --- |
| L'utilisation des connexions persistantes requiert des paramétrages d'Apache et de MySQL pour vous assurer que vous n'atteindrez pas la limite maximale de nombre de connexions simultanées. |

## 6.24.47 mysql_ping() : Pingue le serveur, et se reconnecte au besoin

bool **mysql_ping** ( *resource link_identifier* )

mysql_ping vérifie sir la connexion au serveur fonctionne. Si elle est tombée, une reconnexion automatique est effectuée. Cette fonction est utile pour les scripts qui restent inactifs durant un long moment. Ils peuvent ainsi vérifier si le serveur a fermé la connexion, pour cause d'inactivité. mysql_ping retourne TRUE si la connexion au serveur fonctionne, et FALSE sinon.

Voir aussi mysql_thread_id et mysql_list_processes .

## 6.24.48 mysql_query() : Envoie une requête SQL à un serveur MySQL

resource **mysql_query** ( string query , *resource link_identifier* )

mysql_query envoie une requête SQL à la base de données actuellement active sur le serveur MysQL. Si link_identifier n'est pas précisé, la dernière connexion est utilisée. Si aucune connexion n'a été ouverte, la fonction tentera d'en ouvrir une, avec la fonction mysql_connect mais sans aucun paramètre (c'est−à−dire avec les valeurs par défaut).

Le paramètre optionnel result_mode peut prendre les valeurs de MYSQL_USE_RESULT ou MYSQL_STORE_RESULT . Par défaut, il vaut MYSQL_STORE_RESULT , ce qui fait que le résultat est mis dans un buffer. Voyez aussi la fonction mysql_unbuffered_query pour comprendre cas où le résultat n'est pas mis en buffer.

| Note |
| --- |
| La requête ne doit pas être terminée par un point−virgule. |

Seules les requêtes SQL SELECT , SHOW , EXPLAIN et DESCRIBE , utilisées avec mysql_query retourne une ressource de résultat MySQL, ou bien FALSE si la requête n'a pu être exécuté correctement. Pour les autres types de requêtes SQL, mysql_query retourne TRUE en cas de succès et FALSE en cas d'erreur. Un résultat différent de FALSE , la valeur signifie que la requête soumise est valide et qu'elle a pu être exécutée correctement. Cela n'apporte aucune indication sur le nombre de lignes affectées. Il est parfaitement possible qu'une requête soit réussie, mais qu'elle ne modifie aucune ligne, ou qu'elle n'en retourne aucune.

L'exemple suivant est syntaxiquement invalide, ce qui conduit mysql_query à l'échec et retourne FALSE :

Exemple avec mysql_query

```
 <php
$result = mysql_query("SELECT * WHERE 1=1")
    or die ("Requête invalide");
?>
```

L'exemple suivant est sémantiquement invalide si ma_colonne n'est pas une colonne de la table my_table , ce qui conduit mysql_query à l'échec et retourne FALSE :

Exemple avec mysql_query (2)

```
 <?php
$result = mysql_query("SELECT ma_colonne FROM ma_table")
    or die ("Requête invalide");
?>
```

mysql_query échouera aussi et retournera aussi FALSE si les droits d'accès ne sont pas suffisants.

En supposant que la requête réussisse, vous pouvez appeler mysql_affected_rows pour connaître le nombre de lignes affectées (pour les commandes DELETE, INSERT, REPLACE, ou UPDATE ).

Pour les commandes SELECT , mysql_query retourne un identifiant de résultat que vous pouvez passer à mysql_fetch_array , ou n'importe quelle autre fonction de lecture de résultat. Lorsque vous avez terminé avec le résultat, libérez la mémoire avec mysql_free_result .

Voir aussi mysql_num_rows , mysql_affected_rows , mysql_unbuffered_query , mysql_free_result , mysql_fetch_array , mysql_fetch_row , mysql_fetch_assoc , mysql_result , mysql_select_db et mysql_connect .

## 6.24.49 mysql_real_escape_string() : Protège les caractères spéciaux d'une commande SQL.

string **mysql_real_escape_string** ( string unescaped_string , *resource link_identifier* )

mysql_real_escape_string protège les caractères spéciaux de la chaîne unescaped_string , en prenant en compte le jeu de caractères courant. Le résultat peut être utilisé sans problème avec la fonction mysql_query .

| Note |
| --- |
| mysql_real_escape_string ne remplace ni % , ni _ . |

Exemple avec mysql_real_escape_string

```
<?php
 $link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
 $item = "C'est la belle vie, n'est ce pas?";
 $escaped_item = mysql_real_escape_string($item);
 printf ("Chaîne résultat : : %s\n", $escaped_item);
?>
```

L'exemple ci−dessus va afficher ceci :
Chaîne résultat : C\'est la belle vie, n\'est ce pas?

Voir aussi mysql_escape_string et mysql_character_set_name .

## 6.24.50 mysql_result() : Retourne un champ d'un résultat

mixed **mysql_result** ( resource result , int row , *mixed field* )

mysql_result retourne le contenu d'un champ dans le résultat MySQL result_identifier . L'argument field peut être un indice de champ, ou le nom d'un champ, ou le nom de la table + point + le nom du champ ("table.champ"). Si la colonne a été aliasée, utilisez de préférence l'alias.

Lorsque vous travaillez sur des résultats de grande taille, il est conseillé d'utiliser une des fonctions qui vont rechercher une ligne entière dans un tableau. Ces fonctions sont NETTEMENT plus rapides. De plus, utiliser un offset numériques est aussi beaucoup plus rapide que spécifier un nom littéral.

Les appels mysql_result ne devraient pas être mélangés avec d'autres fonctions qui travaillent aussi sur le résultat.

Exemple avec mysql_result

```
<?php
   $link = mysql_connect("hote", "utilisateur", "mot_de_passe")
           or die("Impossible de se connecter : " . mysql_error());

    $result = mysql_query("SELECT name FROM work.employee")
           or die("Impossible d'exécuter la requête : . mysql_error());

   echo mysql_result($result,2); // Affiche le nom du troisième employé

   mysql_close($link);
?>
```

Alternatives à haut rendement, RECOMMANDEES : mysql_fetch_row , mysql_fetch_array et mysql_fetch_object .

## 6.24.51 mysql_select_db() : Sélectionne une base de données MySQL

bool **mysql_select_db** ( string database_name , *resource link_identifier* )

mysql_select_db change la base de données active sur la connexion représentée par link_identifier . Si aucun identifiant n'est spécifié, la dernière connexion est utilisée. S'il n'y a pas de dernière connexion, la fonction tentera de se connecter seule, avec mysql_connect et les paramètres par défaut.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Toutes les requêtes suivantes avec mysql_query seront faites avec la base de données active.

Voir aussi mysql_connect , mysql_pconnect et mysql_query .

Pour des raisons de compatibilité ascendante mysql_selectdb() est encore disponible. Elle est toutefois obsolète.

## 6.24.52 mysql_stat() : Retourne le statut courant du serveur

string **mysql_stat** ( *resource link_identifier* )

mysql_stat retourne le statut courant du serveur.

Note

mysql_stat actuellement ne retourne que le temps de fonctionnement, les threads, les requêtes, les tables ouvertes, les tables flushées, et le nombre de requêtes par secondes. Pour une liste complète des statuts du serveur, utilisez la requête SHOW STATUS.

Exemple mysql_stat

```
<?php
$link = mysql_connect('localhost', "mysql_user", "mysql_password");
$status = explode('  ',mysql_stat($link));
print_r($status);
?>
```

L'exemple ci−dessus va afficher :

```
Array
(
[0] => Uptime: 5380
[1] => Threads: 2
[2] => Questions: 1321299
[3] => Slow queries: 0
[4] => Opens: 26
[5] => Flush tables: 1
[6] => Open tables: 17
[7] => Queries per second avg: 245.595
)
```

## 6.24.53 mysql_tablename() : Lit le nom de la table qui contient un champ

string **mysql_tablename** ( resource result , int i )

mysql_tablename prend le pointeur de résultat obtenu avec mysql_list_tables ou bien un index entier et retourne le nom de la table. La fonction mysql_tablename , ainsi que toutes les fonctions d'accès aux résultats comme mysql_fetch_array peuvent être utilisées pour déterminer le nom des tables dans le pointeur de résultat.

Exemple avec mysql_tablename

```
<?php
   mysql_connect("localhost", "utilisateur_mysql", "mot_de_passe_mysql");
   $result = mysql_list_tables("ma_bdd");

   for ($i = 0; $i < mysql_num_rows($result); $i++)
       printf ("Table : %s\n", mysql_tablename($result, $i));

   mysql_free_result($result);
?>
```

Voir aussi mysql_list_tables .

## 6.24.54 mysql_thread_id() : Retourne l'identifiant du thread courant

int **mysql_thread_id** ( *resource link_identifier* )

mysql_thread_id retourne l'identifiant du thread courant. Si la connexion est perdue et que vous vous reconnectez (avec mysql_ping par exemple), alors l'identifiant de thread va changer. Cela signifie que vous devez lire l'identifiant de thread, et le garder pour plus tard. Lisez le lorsque vous en avez besoin.

Exemple avec mysql_thread_id

```
<?php
 $link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
 $thread_id = mysql_thread_id($link);
 if ($thread_id){
   printf ("L'identifiant du thread courant est %d\n", $thread_id);
 }
?>
```

L'exemple ci−dessus va afficher :
L'identifiant du thread courant est 73

Voir aussi mysql_ping et mysql_list_processes .

## 6.24.55 mysql_unbuffered_query() : Exécute une requête SQL sans mobiliser les résultats

resource **mysql_unbuffered_query** ( string query , *resource link_identifier* )

mysql_unbuffered_query envoie la requête SQL query au serveur MySQL identifié par link_identifier , sans préparer les résultats pour la lecture, comme le fait mysql_query . D'une part, cela réduit considérablement la consommation de mémoire par MySQL, lorsque les requêtes génère des résultats de grandes tailles. D'autre part, vous pourrez utiliser les résultats dès que la première ligne aura été lue : pas besoin d'attendre que la requête ait complètement été exécutée.

Note

L'intérêt de mysql_unbuffered_query est tempéré par une limitation : mysql_num_rows ne fonctionne pas sur une ressource retournée par mysql_unbuffered_query . Vous devez aussi lire tous les résultats d'une première requête exécutée avec mysql_unbuffered_query , avant de pouvoir en exécuter une autre.

Voir aussi mysql_query .

# 6.25 Réseau

## 6.25.1 Introduction

## 6.25.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.25.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.25.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| define_syslog_variables | "0" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, voyez ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*define_syslog_variables booléen*
: Définit si il faut ou pas définir les différentes variables de syslog, (e.g. $LOG_PID, $LOG_CRON, etc.). Eteindre cette fontionnalité est recommandé pour les performances. Durant l'exécution, vous pouvez définir ces variables en appelant define_syslog_variables .

## 6.25.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.25.6 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

| Constante | Description |
|---|---|
| LOG_CONS | Si il y a une erreur lors de l'envoi de données au système d'historique, l'erreur sera écrite directement sur la console système. |
| LOG_NDELAY | Ouvre la connexion à l'historique immédiatement |
| LOG_ODELAY | (par défaut) retarde l'ouverture de la connexion jusqu'à ce que le premier message soit archivé |
| LOG_PERROR | Affiche aussi le message sur la sortie standard |
| LOG_PID | Ajoute le PID dans chaque message |

| Constante | Description |
|---|---|
| LOG_EMERG | Le système est inutilisable |
| LOG_ALERT | Une mesure doit être prise immédiatement |
| LOG_CRIT | Conditions critiques |

| | |
|---|---|
| LOG_ERR | Erreurs |
| LOG_WARNING | Alertes |
| LOG_NOTICE | Notes (normales mais significantives) |
| LOG_INFO | Message d'informations |
| LOG_DEBUG | Message de débogage |

| Constante | Description |
|---|---|
| DNS_A | Adresse IPv4 |
| DNS_MX | Champs serveur Mail (MX) |
| DNS_CNAME | Champs alias (Nom canonique) (A) |
| DNS_NS | Champs de serveur autorité (NS) |
| DNS_PTR | Champs de pointeur (PTR) |
| DNS_HINFO | Champs d'informations d'hôte (HINFO) (voir la Operating System Names pour la signification de ces valeurs) |
| DNS_SOA | Champs de délégation d'autorité (SOA) |
| DNS_TXT | Texte |
| DNS_ANY | Tout les Champs. Sur la plupart des systèmes, cela retourne tous les Champs, mais il ne faut pas s'y fier. Utilisez DNS_ALL . |
| DNS_AAAA | Adresse IPv6 |
| DNS_ALL | Sollicite itérativement le serveur pour chaque type de champs. |


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- checkdnsrr
- closelog
- debugger_off
- debugger_on
- define_syslog_variables
- dns_check_record
- dns_get_mx
- dns_get_record
- fsockopen
- gethostbyaddr
- gethostbyname
- gethostbynamel
- getmxrr
- getprotobyname
- getprotobynumber
- getservbyname
- getservbyport
- ip2long
- long2ip
- openlog
- pfsockopen
- socket_get_status
- socket_set_blocking
- socket_set_timeout
- syslog


## 6.25.8 closelog() : Ferme la connexion à l'historique système

int **closelog** ( void )

closelog ferme le pointeur qui sert à écrire dans l'historique système. L'utilisation de closelog est optionnelle.

Voir aussi define_syslog_variables , syslog et openlog .

## 6.25.9 debugger_off() : Désactive le debugger interne de PHP.

int **debugger_off** ( void )

debugger_off désactive le debuggeur interne de PHP. Le debuggeur est toujours en cours de développement.

## 6.25.10 debugger_on() : Active le debugger interne de PHP

int **debugger_on** ( string address )

debugger_on active le debugger interne de PHP, et le connecte à l'adresse address . Le debuggeur est toujours en cours de développement.

## 6.25.11 define_syslog_variables() : Initialise toutes les constantes liées au syslog

void **define_syslog_variables** ( void )

define_syslog_variables initialise toutes les constantes utilisées par les fonctions de syslog.

Voir aussi openlog , syslog et closelog .

## 6.25.12 dns_check_record() : Synonyme de checkdnsrr

int **dns_check_record** ( string host , *string type* )

dns_check_record vérifie les enregistrements DNS correspondant à l'hôte internet host , donnés sous forme d'adresse IP ou de nom Internet.

## 6.25.13 dns_get_mx() : Synonyme de getmxrr

int **dns_get_mx** ( string hostname , array mxhosts , *array &weight* )

dns_get_mx lit le champ MX de l'hôte internet hostname .

## 6.25.14 dns_get_record() : Lit les données DNS associées à un hôte

array **dns_get_record** ( string hostname , *int type* , *array &authns* , array &addtl )

| Note |
| --- |
| **dns_get_record** n'est pas implémentée sur les plate−formes Windows. Utilisez la classe PEAR Net_DNS . |

**dns_get_record** retourne un tabluau associatif, contenant **au minimum** les index suivants : host , type , class , ttl .

| Attribut | Signification |
|---|---|
| host | L'enregistrement de l'espace de nom DNS qui est décrit par les autres données. |
| class | dns_get_record ne retourne que la classe d'enregistrement Internet et en tant que tel, cet index vaudra toujours IN . |
| type | Chaîne de caractère contenant le type d'enregistrement. Des attributs supplémentaires seront aussi disponbiles dans le tableau suivant la valeur de ce type. Reportez−vous à la table ci−dessous. |
| ttl | Time To Live : Durée avant expiration de l'enregistrement. Cette valeur est **différente** de la durée avant expiration originale, mais plutôt cette valeur moins la durée depuis la dernière interrogation du serveur DNS responsable. |

hostname doit être un nom d'hôte DNS valide, comme par exemple www.exemple.com . Des résolutions inversées peuvent être faites avec la notation in−addr.arpa, mais la fonction gethostbyaddr est plus efficace pour faire des résolutions inverses.

Par défaut, dns_get_record va rechercher toutes les ressources associées à hostname . Pour limiter la taille de la requête, le paramètre optionnel type peut prendre l'une des valeurs constantes suivantes : DNS_A , DNS_CNAME , DNS_HINFO , DNS_MX , DNS_NS , DNS_PTR , DNS_SOA , DNS_TXT , DNS_ALL et DNS_ANY . Par défaut, il vaut **DNS_ANY** .

| Note |
|---|
| A cause des performances excentriques de la librairie libresolv, suivant les plate−formes, DNS_ANY ne retournera pas toujours tous les enregistrements, et l'option DNS_ALL , bien que plus lente, le fera de manière plus sûre. |

Les paramètres optionnels authns et addtl sont passé par référence, et si fournis, ils recevront les enregistrements de ressources pour les **Authoritative Name Servers** , et tous les **Additional Records** respectivement. Voir l'exemple ci−dessous.

Suivant le type de valeur de l'index type , le tableau associatif peut aussi contenir les index suivants : ip , pri , target , cpu , os , mname , rname , serial , refresh , retry , expire , minimum−ttl , txt .

| Type | Valeur supplémentaire |
|---|---|
| A | ip : Une adresse IP v4, au format numérique. |
| MX | pri : Priorité du serveur de mail. Les nombres faibles indiquent une priorité importante. target : FQDN du serveur de mail. Voir aussi dns_get_mx . |
| CNAME | target : FQDN du nom de l'espace DNS qui sert d'alias à cet enregistrement. |
| NS | target : FQDN du nom de serveur qui est responsable de ce nom de domaine. |
| PTR | target : Nom de domaine sur lequel cet enregistrement pointe. |
| TXT | txt : Chaîne de caractères arbitrairement associée à cet enregistrement. |
| HINFO | cpu : Numéro IANA désignant le processeur de la machine référencée par cet enregistrement. os : Numéro IANA désignant le système d'exploitation de la machine référencée par cet enregistrement. Voir Operating System Names pour connaître la signification de ces valeurs. |
| SOA | mname : FQDN de la source de cet enregistrement. rname : Adresse email du contact administratif de ce domaine. serial : Numéro de série du nom de domaine. refresh : Interval de rafraîchissement (en secondes) que les serveurs de noms secondaires doivent utiliser pour mettre en cache ce nom de domaine. retry : Durée (en seconde) d'attente après un rafraîchissement échoué, avant de faire une seconde tentative. expire : Durée maximale (en seconde) de conservation d'une copie des données de zone sans pouvoir faire de rafraichissement. minimum−ttl : Durée minimale (en seconde) pour un client conserve des données de zone avant qu'il ne soumette une nouvelle requête. Cette configuration peut être annulée par d'autres enregistrements. |
| AAAA | ipv6 : adresse IPv6 |
| SRV | pri : (Priorité) les priorités les plus basses doivent être utilisées en premier. weight : classement pour choisir aléatoirement parmi les serveurs targets . target et port : nom d'hôte et port où le service est disponible. Pour plus d'informations, voir : RFC 2782 |
| NAPTR | order et pref : équivalent à pri et weight ci−dessus. flags , services , regex , et replacement : paramètres tels que définis dans la RFC 2915. |

| Note |
|---|
| En terme de standards DNS, les adresses email sont données au format utilisateur.hote (par exemple : webmestre.exemple.com au contraire du format webmestre@exemple.com). N'oubliez pas de vérifier cette adresse et de la modifier si nécessaire avant de la passer à la fonction mail . |

Exemple avec dns_get_record

```
 <?php
$result = dns_get_record("php.net");
print_r($result);
?>
```

```
/*
Produit un affichage similaire à ceci :
 ----------------------------------------

Array
(
    [0] => Array
        (
            [host] => php.net
            [type] => MX
            [pri] => 5
            [target] => pair2.php.net
            [class] => IN
            [ttl] => 6765
        )

    [1] => Array
        (
            [host] => php.net
            [type] => A
            [ip] => 64.246.30.37
            [class] => IN
            [ttl] => 8125
        )

)
*/
```

Comme il est très courant de rechercher l'IP d'un serveur, une fois que le champ MX a été résolu, dns_get_record retournera aussi un tableau dans le paramètre addtl qui contiendra les enregistrements associés. authns est aussi retourné en contenant une liste des serveurs autorité.

Exemple avec dns_get_record et DNS_ANY

```
 <?php
/* Demande tous ("ANY") les enregistrements pour php.net,
    pu crée les tableaus $authns et $addtl
   contenant une liste des noms de serveurs, et tous
   les enregistrements qui vont avec
    */
$result = dns_get_record("php.net",DNS_ANY,$authns,$addtl);
print "Résultat = ";
print_r($result);
print "Authorité NS = ";
print_r($authns);
print "Additionnel = ";
print_r($addtl);
?>

/*
Produit un affichage similaire à ceci :
 ----------------------------------------

Résultat = Array
(
    [0] => Array
        (
            [host] => php.net
            [type] => MX
            [pri] => 5
            [target] => pair2.php.net
            [class] => IN
            [ttl] => 6765
        )

    [1] => Array
        (
            [host] => php.net
            [type] => A
            [ip] => 64.246.30.37
            [class] => IN
            [ttl] => 8125
        )

)
Authortité NS = Array
(
    [0] => Array
        (
            [host] => php.net
            [type] => NS
            [target] => remote1.easydns.com
            [class] => IN
            [ttl] => 10722
        )

    [1] => Array
        (
            [host] => php.net
```

```
            [type] => NS
            [target] => remote2.easydns.com
            [class] => IN
            [ttl] => 10722
        )

    [2] => Array
        (
            [host] => php.net
            [type] => NS
            [target] => ns1.easydns.com
            [class] => IN
            [ttl] => 10722
        )

    [3] => Array
        (
            [host] => php.net
            [type] => NS
            [target] => ns2.easydns.com
            [class] => IN
            [ttl] => 10722
        )

)
Additionnel = Array
(
    [0] => Array
        (
            [host] => pair2.php.net
            [type] => A
            [ip] => 216.92.131.5
            [class] => IN
            [ttl] => 6766
        )

    [1] => Array
        (
            [host] => remote1.easydns.com
            [type] => A
            [ip] => 64.39.29.212
            [class] => IN
            [ttl] => 100384
        )

    [2] => Array
        (
            [host] => remote2.easydns.com
            [type] => A
            [ip] => 212.100.224.80
            [class] => IN
            [ttl] => 81241
        )

    [3] => Array
        (
            [host] => ns1.easydns.com
            [type] => A
            [ip] => 216.220.40.243
            [class] => IN
            [ttl] => 81241
        )

    [4] => Array
        (
            [host] => ns2.easydns.com
            [type] => A
            [ip] => 216.220.40.244
            [class] => IN
            [ttl] => 81241
        )

)
*/
```

Voir aussi dns_get_mx et dns_check_record

## 6.25.15 fsockopen() : Ouvre une socket de connexion Internet ou Unix

resource **fsockopen** ( string target , int port , *int errno* , *string errstr* , *float timeout* )

fsockopen crée un flux de connexion à l'Internet (AF_INET) ou à un domaine Unix (AF_UNIX). Via Internet, cette fonction va ouvrir une socket de connexion TCP avec l'hôte target sur le port port .

Pour les connexions UDP, vous devez explicitement spécifier le protocole : udp://target . Via un domaine Unix, target représente le chemin jusqu'à la socket, et port doit être mis à 0. L'option timeout sert à donner une durée maximale à cet appel.

fsockopen retourne un pointeur de fichier qui peut être utilisé avec d'autres fonctions fichiers, telles que fgets , fgetss , fputs , fclose et feof .

Si l'appel échoue, fsockopen retourne FALSE , et si les options errno et errstr ont été fournies, elles contiennent désormais les raisons de l'échec. Si l'erreur retournée est 0 et que la fonction retourne FALSE , c'est une indication d'erreur. C'est probablement du à une erreur d'initialisation de la socket. Notez que errno et errstr sont passées par référence.

Suivant les environnements, le type 'domaine Unix' ou l'option timeout ne sont pas toujours disponibles.

La socket sera ouverte par défaut en mode bloquant. Vous pouvez changer de mode en utilisant : stream_set_blocking .

Exemple avec fsockopen

```
<?php
$fp = fsockopen("www.php.net", 80, &$errno, &$errstr, 30);
if(!$fp) {
    echo "$errstr ($errno)<br>\n";
} else {
    fputs($fp,"GET / HTTP/1.0\n\n");
    while(!feof($fp)) {
        echo fgets($fp,128);
    }
    fclose($fp);
}
?>
```

L'exemple ci−dessous décrit comment lire la date et l'heure grâce à un service UDP "daytime" (port 13), sur votre propre machine.

Utilisation d'une connexion UDP

```
<?php
$fp = fsockopen("udp://127.0.0.1", 13, &$errno, &$errstr);
if (!$fp) {
    echo "ERREUR: $errno − $errstr<br>\n";
} else {
    fwrite($fp,"\n");
    echo fread($fp, 26);
    fclose($fp);
}
?>
```

Note

Lors de la spécification d'adresses IPv6 au format numérique (e.g. fe80::1) vous devez placer l'adresse IP entre crochets. Par exemple : tcp://[fe80::1]:80 .

Note

Le paramètre timeout a été introduit en PHP 3.0.9 et le support UDP en PHP 4.

Voir aussi pfsockopen , socket_set_blocking , socket_set_timeout , fgets , fgetss , fputs , fclose et feof .

## 6.25.16 gethostbyaddr() : Retourne le nom d'hôte correspondant à une IP

string **gethostbyaddr** ( string ip_address )

gethostbyaddr retourne le nom d'hôte correspondant à l'IP ip_address . Si une erreur survient, retourne ip_address .

Exemple avec gethostbyaddr

```
<?php
$hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);

print $hostname;
?>
```

Voir aussi gethostbyname .

## 6.25.17 gethostbyname() : Retourne l'adresse IP correspondant à un hôte

string **gethostbyname** ( string hostname )

gethostbyname retourne l'adresse IP correspondant à l'hôte hostname .

Exemple avec gethostbyaddr

```
<?php
$ip = gethostbyname('www.exemple.com');

print $ip;
?>
```

Voir aussi gethostbyaddr .

## 6.25.18 gethostbynamel() : Retourne la liste d'IP correspondante à un hôte.

array **gethostbynamel** ( string hostname )

gethostbynamel retourne la liste d'IP correspondant à l'hôte hostname .

Exemple avec gethostbynamel

```
<?php
 $hotes = gethostbynamel('www.yahoo.com');
 print_r($hotes);
?>
```

Le résultat du script ci−dessus sera :

```
Array
(
[0] => 216.109.125.67
[1] => 64.58.76.228
[2] => 216.109.125.65
[3] => 216.109.125.70
[4] => 64.58.76.226
[5] => 64.58.76.230
[6] => 64.58.76.222
[7] => 216.109.125.66
)
```

Voir aussi gethostbyname , gethostbyaddr , checkdnsrr , getmxrr et la page 8 du manuel.

## 6.25.19 getmxrr() : Retourne les enregistrements MX d'un hôte

int **getmxrr** ( string hostname , array mxhosts , *array weight* )

getmxrr effectue une recherche DNS pour obtenir les enregistrements MX de l'hôte hostname . Retourne TRUE si des enregistrements sont trouvés, et FALSE si une erreur est rencontrée, ou si la recherche échoue.

La liste des enregistrements MX est placée dans le tableau mxhosts . Si le tableau weight est fourni, il sera rempli par les informations de poids.

| Note |
| --- |
| Cette fonction ne doit pas être utilisée à des fin de vérification d'adresse. Seuls le serveur de mail trouvé dans les enregistrements DNS sont retournés. Suivant la RFC 974 lorsqu'aucun serveur de mail n'est listé, hostname doit être utilisé comme serveur de mail, avec la priorité 0 . |

| Note |
| --- |
| Cette fonction n'est pas disponible sur les machines Windows. Pour la remplacer, essayez la classe PEAR : Net_DNS . |

Voir aussi checkdnsrr , gethostbyname , gethostbynamel , gethostbyaddr , et la page 8 du manuel.

## 6.25.20 getprotobyname() : Retourne le numéro de protocole associé à un nom de protocole

int **getprotobyname** ( string name )

getprotobyname retourne le numéro de protocole associé avec le nom de protocole name , comme dans /etc/protocols .

Voir aussi getprotobynumber .

## 6.25.21 getprotobynumber() : Retourne le nom de protocole associé à un numéro de protocole

string **getprotobynumber** ( int number )

getprotobynumber retourne le nom de protocole associé avec le numéro de protocole name , comme dans /etc/protocols .

Voir aussi getprotobyname .

## 6.25.22 getservbyname() : Retourne le numéro de port associé à un service Internet et un protocole.

int **getservbyname** ( string service , string protocol )

getservbyname retourne le numéro de port associé au service service et au protocole protocol , comme dans /etc/services . protocol vaut soit tcp , soit udp .

Voir aussi getservbyport .

## 6.25.23 getservbyport() : Retourne le service Internet qui correspond au port et protocole.

string **getservbyport** ( int port , string protocol )

getservbyport le service internet associé au port port pour le protocole protocol comme dans /etc/services . protocol vaut soit tcp , soit udp .

Voir aussi getservbyname .

## 6.25.24 ip2long() : Convertit une chaîne contenant une adresse (IPv4) IP numérique en adresse littérale.

int **ip2long** ( string ip_address )

ip2long génère une adresse IPv4 à partir de son équivalent numérique.

Exemple ip2long

```
<?php
$ip = gethostbyname("www.php.net");
$out = "Les URLS suivantes sont équivalentes :<br>\n";
$out .= "http://www.php.net/, http://".$ip."/, et http://".ip2long($ip)."/<br>\n";
echo $out;
?>
```

Ce second exemple montre comment afficher une adresse convertie à l'aide de la fonction printf :

Affichage d'adresse IP

```
<?php
   $ip = gethostbyname("www.php.net");
   printf ("%u\n", ip2long ($ip));
   echo $out;
?>
```

Voir aussi long2ip et sprintf .

## 6.25.25 long2ip() : Convertit une adresse IP (IPv4) en adresse IP numérique

string **long2ip** ( int proper_address )

long2ip génère une adresse IP (format aaa.bbb.ccc.ddd) à partir de sa représentation littérale.

Voir aussi ip2long

## 6.25.26 openlog() : Ouvre la connexion à l'historique système

int **openlog** ( string ident , int option , int facility )

openlog ouvre la connexion à l'historique système. La chaîne ident sera ajoutée à chaque message. Les valeurs de option et facility sont données ci−dessous. L'utilisation de openlog est optionnelle;

cette fonction sera automatiquement appelée par syslog si nécessaire, et dans ce cas, l'identification sera mise par défaut à FALSE . facility sert à indiquer quel programme enregistre ce message. Cela vous permet de spécifier (sur la machine d'historique) comment traiter les messages venant de plusieurs serveurs. L'utilisation de openlog est optionnel. Cette fonction sera automatiquement appelée par la fonction syslog si nécessaire, auquel cas, ident vaudra par défaut FALSE .

| Constante | Description |
|---|---|
| LOG_CONS | Si une erreur survient lors de l'envoi des données au gestionnaire d'historique, écrire directement l'erreur sur la console. |
| LOG_NDELAY | Ouvre immédiatement une connexion au gestionnaire d'historique |
| LOG_ODELAY | Retarde l'ouverture de la connexion jusqu'à ce que le premier message soit enregistré (par défaut) |
| LOG_PERROR | Envoie le message au gestionnaire standard |
| LOG_PID | Inclut le PID à chaque message |

Vous pouvez utiliser une ou plusieurs de ces options. Pour les combiner, utiliser l'opérateur OR . Par exemple, pour ouvrir immédiatement la connexion, écrire sur la console et inclure le PID de chaque message, utilisez : LOG_CONS | LOG_NDELAY | LOG_PID .

| Constante | Description |
|---|---|
| LOG_AUTH | sécurité/messages d'autorisation (utilisez LOG_AUTHPRIV, pour remplacer cette constante sur les systèmes où elle est définie). |
| LOG_AUTHPRIV | sécurité/messages d'autorisation (privé) |
| LOG_CRON | démon horloge (cron et at) |
| LOG_DAEMON | autres démons système |
| LOG_KERN | noyau (kernel) |
| LOG_LOCAL0 ... LOG_LOCAL7 | réservé pour utilisation local. Inutilisable sous Windows |
| LOG_LPR | imprimante (line printer subsystem) |
| LOG_MAIL | messagerie mail |
| LOG_NEWS | USENET : groupes de news (newsgroup) |
| LOG_SYSLOG | messages générés en interne par syslogd |
| LOG_USER | messages utilisateurs générique |
| LOG_UUCP | UUCP subsystem |

Voir aussi define_syslog_variables , syslog et closelog .

## 6.25.27 pfsockopen() : Ouvre une socket de connexion Internet ou Unix persistante.

resource **pfsockopen** ( string hostname , int port , *int errno* , *string errstr* , *int timeout* )

pfsockopen se comporte exactement comme fsockopen mais la connexion ouverte le reste, même après la fin du script. C'est la version persistante de fsockopen .

## 6.25.28 socket_get_status() : Retourne les informations sur une socket

array **socket_get_status** ( resource socket_get_status )

**socket_get_status** est un alias de stream_get_meta_data .

## 6.25.29 socket_set_blocking() : Active/désactive le mode bloquant d'une socket

int **socket_set_blocking** ( int socket_descriptor , int mode )

Si mode est FALSE , la socket est mise en mode non bloquant, et si il est TRUE , la socket est mise en mode bloquant. Cela affecte des appels tels que fgets qui lisent depuis une socket. En mode non bloquant, un appel fgets retournera immediatement toujours TRUE tandis qu'en mode bloquant, elle va attendre que des données arrivent pour répondre TRUE .

## 6.25.30 socket_set_timeout() : Alias de stream_set_timeout

bool **socket_set_timeout** ( int socket_descriptor , int seconds , int microseconds )

socket_set_timeout est un alias de stream_set_timeout .

## 6.25.31 syslog() : Génère un message dans l'historique système

int **syslog** ( int priority , string message )

syslog génère un message qui sera inscrit dans l'historique par le système. priority est une combinaison des valeurs d'accès et de niveau, qui seront décrites dans la prochaine section. Le dernier argument est le message à envoyer. Attention : les caractères %m seront remplacés par l'erreur (sous forme de chaîne), présente dans errno .

| Constante | Description |
|---|---|
| LOG_EMERG | système inutilisable |
| LOG_ALERT | une décision doit être prise immédiatement |
| LOG_CRIT | conditions critiques |
| LOG_ERR | conditions d'erreur |
| LOG_WARNING | conditions d'alerte |
| LOG_NOTICE | condition normale, mais significative |
| LOG_INFO | message d'information |
| LOG_DEBUG | message de débuggage |

Utilisation de syslog

```php
 <?php
define_syslog_variables();
// ouverture de syslog, ajout du PID et envoi simultané du
// message à la sortie standard et à un mécanisme
// spécifique
openlog("myScripLog", LOG_PID | LOG_PERROR, LOG_LOCAL0);

// quelques lignes de code
if (authorized_client()) {
    // faire quelquechose
} else {
    // client non autorisé!
    // notation de la tentative
    $access = date("Y/m/d H:i:s");
    syslog(LOG_WARNING,"Client non autorisé: $access $REMOTE_ADDR ($HTTP_USER_AGENT)");
}

  closelog();
?>
```

Pour plus d'informations sur comment mettre en place un gestionnaire d'historique, reportez−vous au manuel Unix, page 5 syslog.conf

- 5

. D'autres informations sur les systèmes d'historique et leurs options sont aussi disponibles dans le manuel syslog

- 3

des machines Unix.

Avec Windows NT, l'historique est pris en charge par Event Log.

| Note |
| --- |
| L'utilisation de LOG_LOCAL0 à LOG_LOCAL7 pour le paramètre facility de la fonction openlog n'est pas disponible sous Windows. |

Voir aussi define_syslog_variables , openlog et closelog .

# 6.26 ODBC unifié

## 6.26.1 Introduction

En plus du support de l'ODBC normal, l'ODBC unifié de PHP vous donne accès à diverses bases de données qui ont emprunté la sémantique des API ODBC pour implémenter leur propres API. Au lieu de maintenir de multiples pilotes qui sont similaires, ces pilotes ont été rassemblés dans un jeu de fonctions ODBC uniques.

Les bases de données suivantes sont supportées par l'ODBC unifié : Adabas D , IBM DB2 , iODBC , Solid , et Sybase SQL Anywhere .

| Note |
| --- |
| Il n'y a pas d'ODBC utilisé lors des connexions aux bases de données ci−dessus. Les fonctions que vous utiliserez portent des noms évocateurs, et utilisent les mêmes syntaxes que leurs cousines d'ODBC. L'exception à ceci est iODBC. En compilant PHP avec le support iODBC, vous pourrez utiliser n'importe quel pilote compatible ODBC avec vos applications PHP. iODBC est mis à jour sur OpenLink Software . Plus d'informations sur iODBC, ainsi qu'un HOWTO (en anglais), sont disponibles sur www.iodbc.org . |

## 6.26.2 Pré−requis

Pour accéder à l'une des bases de données supportées, il faut que les librairies équivalentes soient installées.

## 6.26.3 Installation

*−−with−adabas[=dir]*
: Inclut le support Adabas D. DIR est le dossier d'installation d'Adabas D. Par défaut, c'est /usr/local .

*−−with−sapdb[=dir]*
: Inclut le support SAP DB. DIR est le dossier d'installation de SAP DB. Par défaut, c'est /usr/local .

*−−with−solid[=dir]*
: Inclut le support Solid. DIR est le dossier d'installation de Solid. Par défaut, c'est /usr/local/solid .

*−−with−ibm−db2[=dir]*
: Inclut le support IBM DB2. DIR est le dossier d'installation de DB2. Par défaut, c'est /home/db2inst1/sqllib .

*−−with−empress[=dir]*
: Inclut le support Empress. DIR est le dossier d'installation de Empress. Par défaut, c'est $EMPRESSPATH . Depuis PHP4, cette option ne supporte que Empress Version 8.60 et plus récent.

*−−with−empress−bcs[=dir]*
: Inclut le support Empress Local Access. DIR est le dossier d'installation de Empress. Par défaut, c'est $EMPRESSPATH . Depuis PHP4, cette option ne supporte que Empress Version 8.60 et plus récent.

*−−with−birdstep[=dir]*
: Inclut le support Birdstep. DIR est le dossier d'installation de Birdstep. Par défaut, c'est /usr/local/birdstep .

*−−with−custom−odbc[=dir]*
Inclut le support a user defined ODBC. DIR est le dossier d'installation d'ODBC. Par défaut, c'est /usr/local. Assurez vous que la variable CUSTOM_ODBC_LIBS est définie, et que le fichier odbc.h est dans votre chemin d'inclusion, c'est à dire que vous devriez définir les lignes suivantes pour Sybase SQL Anywhere 5.5.00 sous QNX, avant d'utiliser le script de configuration : CPPFLAGS="−DODBC_QNX −DSQLANY_BUG" LDFLAGS=−lunix CUSTOM_ODBC_LIBS="−ldblib −lodbc".

*−−with−iodbc[=dir]*
Inclut le support iODBC. DIR est le dossier d'installation de iODBC. Par défaut, c'est /usr/local .

*−−with−esoob[=dir]*
Inclut le support Easysoft OOB. DIR est le dossier d'installation de OOB. Par défaut, c'est /usr/local/easysoft/oob/client.

*−−with−unixodbc[=dir]*
Inclut le support unixODBC. DIR est le dossier d'installation d'unixODBC. Par défaut, c'est /usr/local .

*−−with−openlink[=dir]*
Inclut le support OpenLink ODBC. DIR est le dossier d'installation OpenLink. Par défaut, c'est /usr/local . C'est le même que pour iODBC.

*−−with−dbmaker[=dir]*
Inclut le support DBMaker. DIR est le dossier d'installation DBMaker. Par défaut, c'est le chemin de la dernière installation de DBMaker (par exemple /home/dbmaker/3.6 ).

To disable unified ODBC support in **PHP 3** add −−disable−unified−odbc to your configure line. Only applicable if iODBC, Adabas, Solid, Velocis or a custom ODBC interface is enabled.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

## 6.26.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| odbc.default_db * | NULL | PHP_INI_ALL |
| odbc.default_user * | NULL | PHP_INI_ALL |
| odbc.default_pw * | NULL | PHP_INI_ALL |
| odbc.allow_persistent | "1" | PHP_INI_SYSTEM |
| odbc.check_persistent | "1" | PHP_INI_SYSTEM |
| odbc.max_persistent | "−1" | PHP_INI_SYSTEM |
| odbc.max_links | "−1" | PHP_INI_SYSTEM |
| odbc.defaultlrl | "4096" | PHP_INI_ALL |
| odbc.defaultbinmode | "1" | PHP_INI_ALL |

| Note |
|---|
| Les entrées marquées avec une étoile ne sont pas encore implémentées. |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*odbc.default_db chaîne de caractères*
La source de données ODBC si aucune n'est spécifiée dans les fonctions odbc_connect ou odbc_pconnect .

*odbc.default_user chaîne de caractères*

Nom d'utilisateur à utiliser si aucun n'est spécifié dans les fonctions odbc_connect ou odbc_pconnect .

*odbc.default_pw chaîne de caractères*
: Mot de passe à utiliser si aucun n'est spécifié dans les fonctions odbc_connect ou odbc_pconnect .

*odbc.allow_persistent booléen*
: Est ce que les connexions ODBC persistantes sont autorisées ou pas.

*odbc.check_persistent booléen*
: Vérifie que la connexion est valide avant de l'utiliser.

*odbc.max_persistent entier*
: Nombre maximum de connexions persistantes par processus.

*odbc.max_links entier*
: Le nombre maximum de connexions ODBC par processus, y compris les connexions persistantes.

*odbc.defaultlrl entier*
: Gestion des champs de type LONG. Spécifie le nombre d'octets retournés dans les variables.

*odbc.defaultbinmode entier*
: Gestion des données binaires.

## 6.26.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.26.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*ODBC_TYPE (entier)*
*ODBC_BINMODE_PASSTHRU (entier)*
*ODBC_BINMODE_RETURN (entier)*
*ODBC_BINMODE_CONVERT (entier)*
*SQL_ODBC_CURSORS (entier)*
*SQL_CUR_USE_DRIVER (entier)*
*SQL_CUR_USE_IF_NEEDED (entier)*
*SQL_CUR_USE_ODBC (entier)*
*SQL_CONCURRENCY (entier)*
*SQL_CONCUR_READ_ONLY (entier)*
*SQL_CONCUR_LOCK (entier)*
*SQL_CONCUR_ROWVER (entier)*
*SQL_CONCUR_VALUES (entier)*
*SQL_CURSOR_TYPE (entier)*
*SQL_CURSOR_FORWARD_ONLY (entier)*
*SQL_CURSOR_KEYSET_DRIVEN (entier)*
*SQL_CURSOR_DYNAMIC (entier)*
*SQL_CURSOR_STATIC (entier)*
*SQL_KEYSET_SIZE (entier)*
*SQL_CHAR (entier)*
*SQL_VARCHAR (entier)*
*SQL_LONGVARCHAR (entier)*
*SQL_DECIMAL (entier)*

*SQL_NUMERIC (entier)*
*SQL_BIT (entier)*
*SQL_TINYINT (entier)*
*SQL_SMALLINT (entier)*
*SQL_INTEGER (entier)*
*SQL_BIGINT (entier)*
*SQL_REAL (entier)*
*SQL_FLOAT (entier)*
*SQL_DOUBLE (entier)*
*SQL_BINARY (entier)*
*SQL_VARBINARY (entier)*
*SQL_LONGVARBINARY (entier)*
*SQL_DATE (entier)*
*SQL_TIME (entier)*
*SQL_TIMESTAMP (entier)*
*SQL_TYPE_DATE (entier)*
*SQL_TYPE_TIME (entier)*
*SQL_TYPE_TIMESTAMP (entier)*
*SQL_BEST_ROWID (entier)*
*SQL_ROWVER (entier)*
*SQL_SCOPE_CURROW (entier)*
*SQL_SCOPE_TRANSACTION (entier)*
*SQL_SCOPE_SESSION (entier)*
*SQL_NO_NULLS (entier)*
*SQL_NULLABLE (entier)*
*SQL_INDEX_UNIQUE (entier)*
*SQL_INDEX_ALL (entier)*
*SQL_ENSURE (entier)*
*SQL_QUICK (entier)*


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- odbc_autocommit
- odbc_binmode
- odbc_close_all
- odbc_close
- odbc_columnprivileges
- odbc_columns
- odbc_commit
- odbc_connect
- odbc_cursor
- odbc_data_source
- odbc_do
- odbc_error
- odbc_errormsg
- odbc_exec
- odbc_execute
- odbc_fetch_array
- odbc_fetch_into
- odbc_fetch_object

- odbc_fetch_row
- odbc_field_len
- odbc_field_name
- odbc_field_num
- odbc_field_precision
- odbc_field_scale
- odbc_field_type
- odbc_foreignkeys
- odbc_free_result
- odbc_gettypeinfo
- odbc_longreadlen
- odbc_next_result
- odbc_num_fields
- odbc_num_rows
- odbc_pconnect
- odbc_prepare
- odbc_primarykeys
- odbc_procedurecolumns
- odbc_procedures
- odbc_result_all
- odbc_result
- odbc_rollback
- odbc_setoption
- odbc_specialcolumns
- odbc_statistics
- odbc_tableprivileges
- odbc_tables

## 6.26.8 odbc_binmode() : Modifie la gestion des colonnes de données binaires

int **odbc_binmode** ( resource result_id , int mode )

Types ODBC SQL affectés: BINARY, VARBINARY, LONGVARBINARY.

- ODBC_BINMODE_PASSTHRU: Mode Passthru
- ODBC_BINMODE_RETURN: Retourne tel quel.
- ODBC_BINMODE_CONVERT: Convertit en char et retourne la valeur.

Lorsqu'une donnée SQL est convertie en caractère C, les 8 bits du caractère source sont représentés par deux caractères ASCII. Ces caractères sont des représentations ASCII des nombres au format hexadécimal. Par exemple, le binaire 00000001 est converti en "01" et le binaire 11111111 est converti en "FF" .

| mode | longueur | résultat |
|---|---|---|
| ODBC_BINMODE_PASSTHRU | 0 | passthru |
| ODBC_BINMODE_RETURN | 0 | passthru |
| ODBC_BINMODE_CONVERT | 0 | passthru |
| ODBC_BINMODE_PASSTHRU | 0 | passthru |
| ODBC_BINMODE_PASSTHRU | >0 | passthru |
| ODBC_BINMODE_RETURN | >0 | Tel quel |
| ODBC_BINMODE_CONVERT | >0 | Caractère |

Si odbc_fetch_into est utilisé, passthru signifie qu'une chaîne vide sera retournée pour ces colonnes.

Si result_id vaut 0 , ces paramètres seront appliqués aux nouveaux résultats.

| Note |
| --- |
| La valeur par défaut de longreadlen est 4096 et la valeur par défaut de odbc_binmode est ODBC_BINMODE_RETURN . La gestion des colonnes binaires est aussi modifiée par odbc_longreadlen . |

## 6.26.9 odbc_close_all() : Ferme toutes les connexions ODBC

void **odbc_close_all** ( void )

odbc_close_all ferme toutes les connexions ODBC à des sources de données.

| Note |
| --- |
| odbc_close_all échouera s'il y a des transactions en cours sur cette connexion. Dans ce cas, la connexion restera ouverte. |

## 6.26.10 odbc_close() : Ferme une connexion ODBC

void **odbc_close** ( resource connection_id )

odbc_close ferme la connexion avec la source de données représentée par l'identifiant de connexion connection_id .

| Note |
| --- |
| odbc_close échouera s'il y a des transactions en cours sur cette connexion. Dans ce cas, la connexion restera ouverte. |

## 6.26.11 odbc_columnprivileges() : Liste les colonnes et leurs droits associés

int **odbc_columnprivileges** ( resource connection_id , *string qualifier* , *string owner* , *string table_name* , *string column_name* )

odbc_columnprivileges liste les colonnes et leurs droits associés pour la table table_name . odbc_columnprivileges retourne un identifiant de résultat ODBC, ou bien FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- GRANTOR
- GRANTEE
- PRIVILEGE
- IS_GRANTABLE

Le résultat est ordonné par TABLE_QUALIFIER, TABLE_OWNER et TABLE_NAME.

Le paramètre column_name accepte des masques de recherche ('%' pour remplacer zéro ou plus

caractères, et '_' pour n'en remplacer qu'un seul).

## 6.26.12 odbc_columns() : Liste les colonnes d'une table

resource **odbc_columns** ( resource connection_id , *string qualifier* , *string schema* , *string table_name* , *string column_name* )

odbc_columns liste toutes les colonnes de la source de données. odbc_columns retourne un identifiant de résultat ODBC, ou bien FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- COLUMN_NAME
- DATA_TYPE
- TYPE_NAME
- PRECISION
- LENGTH
- SCALE
- RADIX
- NULLABLE
- REMARKS

Le résultat est ordonné par TABLE_QUALIFIER, TABLE_OWNER et TABLE_NAME.

Les paramètres owner , column_name et table_name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

Voir aussi odbc_columnprivileges pour connaître les droits associés.

## 6.26.13 odbc_commit() : Valide une transaction ODBC

bool **odbc_commit** ( resource connection_id )

odbc_commit valide toutes les transactions en cours sur la connexion connection_id . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.26.14 odbc_connect() : Connexion à une source

resource **odbc_connect** ( string dsn , string user , string password , *int cursor_type* )

odbc_connect retourne un identifiant de connexion ODBC ou 0 ( FALSE ) en cas d'erreur.

L'identifiant de connexion retourné par cette fonction est nécessaire pour toutes les autres fonctions ODBC. Vous pouvez avoir de multiples connexions en même temps. Le quatrième paramètre fixe le type de pointeur de résultat utilisé pour cette connexion. Ce paramètre n'est généralement pas nécessaire, mais il peut être utile pour contourner certains problèmes ODBC.

Avec certains pilotes ODBC, l'exécution de procédures enregistrées complexes peut produire l'erreur suivante : "Cannot open a cursor on a stored procedure that has anything other than a single select statement in it", ce qui signifie : "Impossible de créer un pointeur de résultat dans une procédure enregistrée qui est réduite à une simple sélection (SELECT)). Utiliser l'option SQL_CUR_USE_ODBC permet d'éviter cette erreur. De plus, certains pilotes ne supportent pas le paramètre optionnel de numéro de ligne dans odbc_fetch_row . SQL_CUR_USE_ODBC peut aussi permettre de résoudre ces problèmes.

Les constantes suivantes sont définies comme types de pointeur :

- SQL_CUR_USE_IF_NEEDED
- SQL_CUR_USE_ODBC
- SQL_CUR_USE_DRIVER
- SQL_CUR_DEFAULT

Pour les connexions persistantes, reportez−vous à odbc_pconnect .

## 6.26.15 odbc_cursor() : Lecture du pointeur de fiche courante (cursorname)

string **odbc_cursor** ( resource result_id )

odbc_cursor lit le pointeur de fiche courante (cursorname) pour le résultat result_id .

## 6.26.16 odbc_data_source() : Retourne des informations sur la connexion courante

resource **odbc_data_source** ( resource connection_id , constant fetch_type )

odbc_data_source retourne FALSE en cas d'erreur, et un tableau en cas de réussite.

odbc_data_source retourne des informations sur la connexion identifiée par la ressource connection_id , suivies d'informations sur le DSN. Le paramètre connection_id est une connexion ODBC valide. Le paramètre fetch_type peut être l'une des deux constantes suivantes : SQL_FETCH_FIRST ou SQL_FETCH_NEXT . Utilisez SQL_FETCH_FIRST la première fois que la fonction est appelée, et puis SQL_FETCH_NEXT .

## 6.26.17 odbc_do() : Synonyme de odbc_exec

resource **odbc_do** ( resource conn_id , string query )

odbc_do exécute la requête query sur la connexion connection_id .

## 6.26.18 odbc_error() : Lit le dernier code d'erreur

string **odbc_error** ( *resource connection_id* )

odbc_error retourne un état ODBC sur 6 chiffres, ou une chaîne vide s'il n'y avait plus d'erreurs. Si connection_id est spécifié, le dernier état ODBC de cette connexion est retourné. Si connection_id

est omis, c'est le dernier état de n'importe quelle connexion qui est retourné.

Voir aussi odbc_errormsg et odbc_exec .

## 6.26.19 odbc_errormsg() : Lit le dernier message d'erreur

string **odbc_errormsg** ( *resource connection_id* )

odbc_errormsg retourne une chaîne contenant le dernier message d'erreur ODBC, ou une chaîne vide s'il n'y avait pas d'erreur. Si connection_id est spécifié, le dernier état ODBC de cette connexion est retourné. Si connection_id est omis, c'est le dernier état de n'importe quelle connexion qui est retourné.

Voir aussi odbc_error et odbc_exec .

## 6.26.20 odbc_exec() : Prépare et exécute une requête SQL

resource **odbc_exec** ( resource connection_id , string query_string )

odbc_exec retourne FALSE en cas d'erreur, ou bien retourne un identifiant de résultat ODBC en cas d'exécution réussie.

odbc_exec envoie une commande SQL à la source de données représentée par connection_id . Ce paramètre doit être un identifiant valide de connexion, retourné par odbc_connect ou odbc_pconnect .

Voir aussi odbc_prepare et odbc_execute pour les exécutions multiples de requêtes SQL.

## 6.26.21 odbc_execute() : Exécute une requête SQL préparée.

bool **odbc_execute** ( resource result_id , *array parameters_array* )

odbc_execute exécute une requête SQL préparée par odbc_prepare . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Le tableau de paramètres parameters_array ne sert que si vous avez besoin de paramètrer votre requête.

Les valeurs du paramètre parameter_array seront substituées dans les variables de requêtes de la requête préparée.

Tout paramètre de parameter_array qui commence et termine par des guillements simples sera considéré comme un nom de fichier à lire et envoyer à la base de données, avec la variable de requête appropriée.

| Note |
| --- |
| Depuis PHP 4.1.1, cette fonction de lecture de fichier a les limitations suivantes :<br>• La lecture du fichier **n'est pas** sujette à aucune restriction de la page du safe mode ou de la directive open-basedir . Cela a été corrigé en PHP 4.2.0.<br>• Les fichiers distants ne sont pas supportés.<br>• Si vous voulez stocker une chaîne de caractères qui commence et se finit réellement par des guillemets, vous devez les protéger avec des anti slash, ou bien en ajoutant un espace au début ou à la fin de la chaîne, pour éviter que ce paramètre soit confondu avec un nom de fichier. Si ce n'est pas possible dans le cadre de votre application, vous devrez passer par la fonction odbc_exec . |

## 6.26.22 odbc_fetch_array() : Lit une ligne de résultat dans un tableau associatif

array **odbc_fetch_array** ( resource result , *int* *rownumber* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.26.23 odbc_fetch_into() : Lit une ligne de résultat, et la place dans un tableau.

bool **odbc_fetch_into** ( resource result_id , *int* *rownumber* , array result_array ) resource **odbc_fetch_into** ( resource result_id , array result_array , *int* *rownumber* )

odbc_fetch_into retourne le nombre de colonnes dans le résultat, ou FALSE en cas d'erreur. result_array doit avoir été passé par référence, mais il peut être de n'importe quel type, étant donné qu'il sera converti en tableau. Le tableau contiendra les valeurs des colonnes, ces dernières étant numérotées à partir de 0.

Exemple avec odbc_fetch_into (avant PHP 4.0.6)

```
<?php
 $rc = odbc_fetch_into($res_id, $my_array);
?>
```

ou

```
<?php
 $rc = odbc_fetch_into($res_id, $row, $my_array);
 $rc = odbc_fetch_into($res_id, 1, $my_array);
?>
```

Jusqu'en PHP 4.0.5, le paramètre result_array n'a plus besoin d'être passé par référence.

Depuis PHP 4.0.6, le paramètre rownumber ne peut pas être passé comme une constante, mais comme une variable.

Depuis PHP 4.2.0, les paramètres result_array et rownumber ont été inversés. Cela permet à rownumber d'être à nouveau une constante. Cette modification est la dernière pour cette fonction.

Exemple avec odbc_fetch_into (après PHP 4.0.6)

```
<?php
 $rc = odbc_fetch_into($res_id, $my_array);
?>
```

ou

```
<?php
 $row = 1;
 $rc = odbc_fetch_into($res_id, $row, $my_array);
?>
```

Exemple avec odbc_fetch_into en PHP 4.2.0

```
<?php
 $rc = odbc_fetch_into($res_id, $my_array);
?>
```

ou

```
<?php
 $rc = odbc_fetch_into($res_id,$my_array, 2);
?>
```

## 6.26.24 odbc_fetch_object() : Lit une ligne de résultat dans un objet

object **odbc_fetch_object** ( resource result , *int rownumber* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.26.25 odbc_fetch_row() : Lit une ligne de résultat

bool **odbc_fetch_row** ( resource result_id , *int row_number* )

Si odbc_fetch_row a réussi, TRUE est retourné. S'il n'y avait plus de ligne, ou en cas d'erreur, FALSE est retourné.

odbc_fetch_row lit une ligne dans le résultat identifié par result_id et retourné par odbc_do ou odbc_exec . Après odbc_fetch_row , les champs seront accessibles avec la fonction odbc_result .

Si row_number est omis, row_number va tenter de lire la prochaine ligne dans le résultat. Des appels répétés à odbc_fetch_row avec et sans paramètre row_number peuvent être combinés librement.

Pour passer en revue toutes les lignes d'un résultat plusieurs fois, vous pouvez appeler odbc_fetch_row avec row_number = 1, puis continue à appeler odbc_fetch_row sans le paramètre row_number pour passer en revue tout le résultat. Si un pilote ne supporte pas la lecture des lignes par numéro, le paramètre sera ignoré.

## 6.26.26 odbc_field_len() : Lit la longueur d'un champ

int **odbc_field_len** ( resource result_id , int field_number )

odbc_field_len retourne la longueur du champ référence par le nombre field_number , dans la connexion ODBC result_id . Les numéros de champs commencent à 1.

## 6.26.27 odbc_field_name() : Lit le nom de la colonne

string **odbc_field_name** ( resource result_id , int field_number )

odbc_field_name lit le nom de la colonne dont l'index est field_number . La numérotation des colonnes commence à 1. FALSE est retourné en cas d'erreur.

## 6.26.28 odbc_field_num() : Numéro de colonne

int **odbc_field_num** ( resource result_id , string field_name )

odbc_field_num retourne le numéro de la colonne nommée field_name . Ce numéro correspond à l'index du champ dans le résultat ODBC. La numérotation commence à 1. FALSE est retourné en cas d'erreur.

## 6.26.29 odbc_field_precision() : Alias de odbc_field_len

string **odbc_field_precision** ( resource result_id , int field_number )

odbc_field_precision retourne la précision du champ référencé par son numéro field_number , dans le résultat ODBC result_id .

Voir aussi : odbc_field_scale pour connaître l'échelle d'un nombre à virgule flottante.

## 6.26.30 odbc_field_scale() : Lit l'échelle d'un champ

string **odbc_field_scale** ( resource result_id , int field_number )

odbc_field_precision retourne l'échelle du champ référencé par son numéro de champ field_number dans le résultat ODBC result_id .

## 6.26.31 odbc_field_type() : Type de données d'un champ

string **odbc_field_type** ( resource result_id , int field_number )

odbc_field_type retourne le type de données SQL d'un champ, identifié par son index. La numérotation des champs commence à 1.

## 6.26.32 odbc_foreignkeys() : Liste les clés étrangères

resource **odbc_foreignkeys** ( resource connection_id , string pk_qualifier , string pk_owner , string pk_table , string fk_qualifier , string fk_owner , string fk_table )

odbc_foreignkeys liste les clés étrangères utilisées dans la table pk_table . odbc_foreignkeys retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- PKTABLE_QUALIFIER
- PKTABLE_OWNER
- PKTABLE_NAME
- PKCOLUMN_NAME
- FKTABLE_QUALIFIER
- FKTABLE_OWNER
- FKTABLE_NAME
- FKCOLUMN_NAME
- KEY_SEQ
- UPDATE_RULE
- DELETE_RULE
- FK_NAME
- PK_NAME

Si pk_table contient un nom de table, odbc_foreignkeys retourne la clé primaire de la table pk_table , et toutes les clés étrangères qui y font référence.

Si fk_table contient un nom de table, odbc_foreignkeys retourne la liste des clés étrangères de la table fk_table , et les clés primaires (d'autres tables) qui y font référence.

Si pk_table et fk_table contiennent des noms de tables, odbc_foreignkeys retourne la liste des clés étrangères de la table fk_table qui utilisent la clé primaire de la table pk_table . Cette liste devrait ne contenir qu'une clé au mieux.

## 6.26.33 odbc_free_result() : Libère les ressources associées à un résultat

bool **odbc_free_result** ( resource result_id )

odbc_free_result retourne toujours TRUE .

odbc_free_result n'est nécessaire que si vous craignez d'utiliser trop de mémoire lors de l'exécution de votre script. Tous les résultats en mémoire seront libérés dès la fin du script. Mais, si vous êtes sûr que vous n'aurez plus besoin d'un résultat jusqu'à la fin de votre script, vous pouvez appeler odbc_free_result , et la mémoire associée à result_id sera libérée.

| Note |
| --- |
| Si l'auto−validation est désactivée (voir odbc_autocommit ) et que vous appelez odbc_free_result avant de valider vos requêtes, toutes les transactions préparées seront annulées. |

## 6.26.34 odbc_gettypeinfo() : Liste les types de données supportés par une source

int **odbc_gettypeinfo** ( resource connection_id , *int data_type* )

odbc_gettypeinfo liste les types de données qui sont supportées par une source. odbc_gettypeinfo retourne un identifiant de résultat, ou FALSE en cas d'erreur. L'argument optionnel data_type peut être utilisé pour restreindre les informations à un seul type de données.

Le résultat possède les colonnes suivantes :

- TYPE_NAME
- DATA_TYPE
- PRECISION
- LITERAL_PREFIX
- LITERAL_SUFFIX
- CREATE_PARAMS
- NULLABLE
- CASE_SENSITIVE
- SEARCHABLE
- UNSIGNED_ATTRIBUTE
- MONEY
- AUTO_INCREMENT
- LOCAL_TYPE_NAME
- MINIMUM_SCALE
- MAXIMUM_SCALE

Le résultat est ordonné par DATA_TYPE et TYPE_NAME.

## 6.26.35 odbc_longreadlen() : Gestion des colonnes de type LONG

int **odbc_longreadlen** ( resource result_id , int length )

Types ODBC SQL affectés: LONG, LONGVARBINARY.

Le nombre d'octets retournés à PHP est contrôlé par le paramètre length . Si sa valeur est 0, les colonnes de type Long seront transformées en chaînes vides.

| Note |
| --- |
| La gestion des types LONGVARBINARY est aussi affectée par odbc_binmode . |

## 6.26.36 odbc_next_result() : Vérifie si plusieurs résultats sont disponibles

bool **odbc_next_result** ( resource result_id )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.26.37 odbc_num_fields() : Nombre de colonnes dans un résultat

int **odbc_num_fields** ( resource result_id )

odbc_num_fields retourne le nombre de colonnes dans un résultat ODBC. odbc_num_fields retournera −1 en cas d'erreur. L'argument est un identifiant de résultat valide, retourné par odbc_exec .

## 6.26.38 odbc_num_rows() : Nombre de lignes dans un résultat

int **odbc_num_rows** ( resource result_id )

odbc_num_rows retourne le nombre de lignes dans un résultat ODBC. odbc_num_rows retournera −1 en cas d'erreur. Pour les commandes INSERT, UPDATE et DELETE, odbc_num_rows retourne le nombre de ligne affectées. Pour les commandes SELECT, cela PEUT être le nombre de lignes disponibles, mais ce n'est pas certain.

Note: odbc_num_rows après un SELECT retournera −1 avec de nombreux pilotes.

## 6.26.39 odbc_pconnect() : Ouvre une connexion persistante à une source de données.

resource **odbc_pconnect** ( string dsn , string user , string password , *int cursor_type* )

odbc_pconnect retourne un identifiant de connexion ODBC ou 0 ( FALSE ) en cas d'erreur. odbc_pconnect se comporte de manière similaire à odbc_connect , mais la connexion ouverte n'est pas vraiment terminée lorsque le script est terminé. Les prochaines requêtes qui se feront sur une connexion dont les dsn , user , password sont les mêmes que celle−ci (avec odbc_connect et odbc_pconnect ) réutiliseront la connexion ouverte.

| Note |
| --- |
| Les connexions persistantes n'ont aucun effet si PHP est utilisé comme CGI. |

Pour plus de détails sur le paramètre optionnel cursor_type , voyez odbc_connect . Pour plus de détails sur les connexions persistantes, reportez−vous à la FAQ PHP .

## 6.26.40 odbc_prepare() : Prépare une commande pour l'exécution

resource **odbc_prepare** ( resource connection_id , string query_string )

odbc_prepare prépare une commande pour l'exécution.

odbc_prepare retourne un identifiant de résultat ODBC si la commande SQL a été préparée avec succès. L'identifiant peut être utilisé plus tard pour exécuter la commande avec odbc_execute .

## 6.26.41 odbc_primarykeys() : Liste les colonnes utilisées dans une clé primaire

resource **odbc_primarykeys** ( resource connection_id , string qualifier , string owner , string table )

odbc_primarykeys liste les colonnes utilisées dans une clé primaire de la table table . odbc_primarykeys retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- COLUMN_NAME
- KEY_SEQ
- PK_NAME

## 6.26.42 odbc_procedurecolumns() : Liste les paramètres des procédures

resource **odbc_procedurecolumns** ( resource connection_id , *string qualifier* , *string owner* , *string proc* , *string column* )

odbc_procedurecolumns liste les paramètres d'entrée et de sortie, ainsi que les colonnes utilisées dans les procédures désignées par les paramètres. odbc_procedurecolumns retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- PROCEDURE_QUALIFIER
- PROCEDURE_OWNER
- PROCEDURE_NAME
- COLUMN_NAME
- COLUMN_TYPE

- DATA_TYPE
- TYPE_NAME
- PRECISION
- LENGTH
- SCALE
- RADIX
- NULLABLE
- REMARKS

Le résultat est ordonné par PROCEDURE_QUALIFIER, PROCEDURE_OWNER, PROCEDURE_NAME et COLUMN_TYPE.

Les paramètres owner , proc et column acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

## 6.26.43 odbc_procedures() : Liste les procédure stockées

resource **odbc_procedures** ( resource connection_id , *string qualifier* , *string owner* , *string name* )

odbc_procedures liste toutes les procédures stockées dans la source de données. odbc_procedures retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- PROCEDURE_QUALIFIER
- PROCEDURE_OWNER
- PROCEDURE_NAME
- NUM_INPUT_PARAMS
- NUM_OUTPUT_PARAMS
- NUM_RESULT_SETS
- REMARKS
- PROCEDURE_TYPE

Les paramètres owner et name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

## 6.26.44 odbc_result_all() : Affiche le résultat sous la forme d'une table HTML.

int **odbc_result_all** ( resource result_id , *string format* )

odbc_result_all retourne le nombre de lignes dans le résultat, ou FALSE en cas d'erreur.

odbc_result_all affiche toutes les lignes d'un résultat. L'affichage se fait au format HTML. Avec l'option format , il est possible de modifier l'aspect global de la table.

## 6.26.45 odbc_result() : Lit un champ de résultat UODBC

string **odbc_result** ( resource result_id , mixed field )

odbc_result retourne le contenu d'un champ.

field peut être aussi bien un entier, contenant le numéro de colonne du champ, dans le résultat, ou bien une chaîne de caractère, qui représente le nom du champ. Par exemple :

Exemple avec odbc_result

```
<?php
 $item_3 = odbc_result($Query_ID, 3 );
 $item_val = odbc_result($Query_ID, "val");
?>
```

Le premier appel à odbc_result retourne la valeur du troisième champ de la ligne courante, du résultat result_id . Le deuxième appel à odbc_result retourne la valeur du troisième champ dont le nom est "val" de la ligne courante, du résultat result_id . Une erreur survient si le paramètre de colonne est inférieur à 1, ou dépasse le nombre de colonnes du résultat. De la même manière, une erreur survient si le nom du champ passé ne correspond à aucun champ dans le résultat.

Les index de champs commencent à 1. Pour plus d'informations sur la façon de lire des colonnes de type binaire ou long, reportez−vous à odbc_binmode et odbc_longreadlen .

## 6.26.46 odbc_rollback() : Annule une transaction

int **odbc_rollback** ( resource connection_id )

odbc_rollback annule toutes les transactions sur la connexion connection_id . odbc_rollback retourne TRUE en cas de succès, et FALSE en cas d'échec.

## 6.26.47 odbc_setoption() : Modifie les paramètres ODBC.

int **odbc_setoption** ( resource id , int function , int option , int param )

odbc_setoption donne accès aux options ODBC pour une connexion particulière ou un résultat de requête. Elle a été écrite pour aider à la résolution de problèmes liés aux pilotes ODBC récalcitrants. Vous aurez sûrement à utiliser odbc_setoption si vous êtes un programmeur ODBC et que vous comprenez les divers effets des options disponibles. Vous aurez aussi besoin d'un bon manuel de référence pour comprendre les options et leur usage. Différentes versions de pilotes supportent différentes versions d'options.

Etant donné que les effets peuvent varier d'un pilote à l'autre, l'utilisation de odbc_setoption dans des scripts voués à être livrés au public est très fortement déconseillée. De plus, certaines options ODBC ne sont pas disponibles car elles doivent être fixées avant l'établissement de la connexion. Cependant, si dans un cas bien spécifique, odbc_setoption vous permet d'utiliser PHP sans que votre patron ne vous pousse à utiliser un produit commercial, alors cela n'a pas d'importance.

id est un identifiant de connexion, ou un identifiant de résultat, pour lequel vous souhaitez modifier des options. Pour SQLSetConnectOption(), c'est un identifiant de connexion. Pour SQLSetStmtOption(), c'est un identifiant de résultat.

function est la fonction ODBC à utiliser. La valeur doit être de 1 pour utiliser SQLSetConnectOption() et 2 pour SQLSetStmtOption().

Le paramètre option est l'option à modifier.

Le paramètre param est la valeur de l'option option .

Exemple de modification d'option ODBC

```
<?php
// 1. L'option 102 de SQLSetConnectOption() est SQL_AUTOCOMMIT.
// 1 de SQL_AUTOCOMMIT est SQL_AUTOCOMMIT_ON.
//    Cet exemple a le meme effet que
//    odbc_autocommit($conn, true);
odbc_setoption($conn, 1, 102, 1);
// 2. Option 0 de SQLSetStmtOption() est SQL_QUERY_TIMEOUT.
//    Cet exemple fixe le délai d'expiration à 30 secondes.
$result = odbc_prepare($conn, $sql);
odbc_setoption($result, 2, 0, 30);
odbc_execute($result);
?>
```

## 6.26.48 odbc_specialcolumns() : Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table

resource **odbc_specialcolumns** ( resource connection_id , int type , string qualifier , string owner , string table , int scope , int nullable )

Lorsque le type est SQL_BEST_ROWID, odbc_specialcolumns retourne la ou les colonnes qui permettent de repérer uniquement chaque ligne d'une table.

Lorsque le type type est SQL_ROWVER, odbc_specialcolumns retourne l'ensemble optimal de colonnes tel qu'en lisant les valeurs de ces colonnes, on puisse spécifier n'importe quelle ligne de manière unique.

odbc_specialcolumns retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- SCOPE
- COLUMN_NAME
- DATA_TYPE
- TYPE_NAME
- PRECISION
- LENGTH
- SCALE
- PSEUDO_COLUMN

Le résultat est ordonné par SCOPE.

## 6.26.49 odbc_statistics() : Calcule des statistiques sur une table

resource **odbc_statistics** ( resource connection_id , string qualifier , string owner , string table_name , int unique , int accuracy )

odbc_statistics effectue quelques statistiques sur une table et ses index. odbc_statistics retourne un identifiant de résultat, ou FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- NON_UNIQUE
- INDEX_QUALIFIER

- INDEX_NAME
- TYPE
- SEQ_IN_INDEX
- COLUMN_NAME
- COLLATION
- CARDINALITY
- PAGES
- FILTER_CONDITION

Le résultat est ordonné par NON_UNIQUE, TYPE, INDEX_QUALIFIER, INDEX_NAME et SEQ_IN_INDEX.

## 6.26.50 odbc_tableprivileges() : Liste les tables et leurs privilèges

int **odbc_tableprivileges** ( resource connection_id , *string qualifier* , *string owner* , *string name* )

odbc_tableprivileges liste les tables de la source et leurs droits associés. odbc_tableprivileges retourne un identifiant de résultat ODBC, ou bien FALSE en cas d'erreur.

Le résultat possède les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- GRANTOR
- GRANTEE
- PRIVILEGE
- IS_GRANTABLE

Le résultat est ordonné par TABLE_QUALIFIER, TABLE_OWNER et TABLE_NAME.

Les paramètres owner et name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

## 6.26.51 odbc_tables() : Liste les tables d'une source.

int **odbc_tables** ( resource connection_id , *string qualifier* , *string owner* , *string name* , *string types* )

odbc_tables liste toutes les tables de la source et retourne un identifiant de résultat ODBC, ou bien FALSE en cas d'erreur.

Le résultat contient les colonnes suivantes :

- TABLE_QUALIFIER
- TABLE_OWNER
- TABLE_NAME
- TABLE_TYPE
- REMARKS

Le résultat est ordonné grâce aux options TABLE_TYPE, TABLE_QUALIFIER, TABLE_OWNER et TABLE_NAME.

Les paramètres owner et name acceptent des masques de recherche ('%' pour remplacer zéro ou plus caractères, et '_' pour n'en remplacer qu'un seul).

Pour supporter les énumérations de qualificateurs propriétaires et types de table, la sémantique suivante pour les paramètres qualifier , owner , name et table_type sont disponibles :

- Si qualifier est un signe de pourcentage (%), et owner et name sont des chaînes vides, alors le résultat contient la liste des qualifiers valides pour la source. (toutes les colonnes hormis TABLE_QUALIFIER contiennent NULL).
- Si owner est un signe de pourcentage (%), et qualifier et name sont des chaînes vides, alors le résultat contient la liste des propriétaires de la source (toutes les colonnes hormis TABLE_OWNER contiennent NULL).
- Si table_type est un signe de pourcentage (%), et qualifier , owner et name sont des chaînes vides, alors le résultat contient la liste des types de tables de la source (toutes les colonnes hormis TABLE_TYPE contiennent NULL).

Si table_type n'est pas une chaîne vide, il doit contenir une liste de valeurs, séparées par des virgules, qui representent les types recherchés. Chaque valeur peut être insérée entre guillemets simples ('), ou sans guillemets. Par exemple "'TABLE','VIEW'" ou "TABLE, VIEW". Si la source de données ne supporte pas un type de table donné, odbc_tables ne retournera aucun résultat pour ce type.

Voir aussi odbc_tableprivileges pour connaître les droits associés.

# 6.27 Oracle 8

## 6.27.1 Introduction

Ces fonctions vous permettront d'accéder aux serveurs Oracle8 et Oracle7. Elles utilisent l'interface Oracle8 Call−Interface (oci8). Vous aurez donc besoin des librairies clientes Oracle8 pour pouvoir les utiliser.

Il faut noter que cette extension est plus souple que l'extension Oracle officielle. Elle supporte notamment les liaisons entre les variables globales et locales de PHP avec des emplacements Oracle; elle supporte complètement les types LOB, FILE et ROWID et vous permet d'utiliser des variables de définitions personnalisables.

## 6.27.2 Pré−requis

Avant d'utiliser cette extension, assurez−vous que vous avez bien paramétré vos variables d'environnement Oracle, ainsi que votre démon utilisateur. Les variables dont vous pouvez avoir besoin sont :

- ORACLE_HOME
- ORACLE_SID
- LD_PRELOAD
- LD_LIBRARY_PATH
- NLS_LANG
- ORA_NLS33

Après avoir configuré ces variables pour votre utilisateur "serveur web", assurez−vous aussi d'ajouter cet utilisateur (nobody, www) au group Oracle.

Note

Si votre serveur web ne démarre pas, ou crashe au démarrage

Vérifiez que Apache a bien été compilé avec la librairie pthread :

Configuration Apache avec pthread

```
 # ldd /www/apache/bin/httpd
     libpthread.so.0 => /lib/libpthread.so.0 (0x4001c000)
    libm.so.6 => /lib/libm.so.6 (0x4002f000)
    libcrypt.so.1 => /lib/libcrypt.so.1 (0x4004c000)
    libdl.so.2 => /lib/libdl.so.2 (0x4007a000)
    libc.so.6 => /lib/libc.so.6 (0x4007e000)
    /lib/ld−linux.so.2 => /lib/ld−linux.so.2 (0x40000000)
```

Si la libpthread n'est pas listée, vous devez réinstaller Apache :

Reinstaller Apache avec pthread

```
 # cd /usr/src/apache_1.3.xx
# make clean
# LIBS=−lpthread ./config.status
# make
# make install
```

Notez bien que sur certains systèmes comme UnixWare il faut utiliser libthread au lieu de libpthread. PHP et Apache doivent alors être configurés avec EXTRA_LIBS=−lthread.

## 6.27.3 Installation

Vous devez compiler PHP avec l'option −−with−oci8[=DIR] , où DIR vaut par défaut la variable d'environnement ORACLE_HOME.

## 6.27.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.27.5 Types de ressources

## 6.27.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*OCI_DEFAULT (entier)*
*OCI_DESCRIBE_ONLY (entier)*
*OCI_COMMIT_ON_SUCCESS (entier)*
*OCI_EXACT_FETCH (entier)*
*SQLT_BFILEE (entier)*
*SQLT_CFILEE (entier)*
*SQLT_CLOB (entier)*
*SQLT_BLOB (entier)*
*SQLT_RDD (entier)*
*OCI_B_SQLT_NTY (entier)*
*OCI_SYSDATE (entier)*
*OCI_B_BFILE (entier)*
*OCI_B_CFILEE (entier)*
*OCI_B_CLOB (entier)*
*OCI_B_BLOB (entier)*
*OCI_B_ROWID (entier)*
*OCI_B_CURSOR (entier)*
*OCI_B_BIN (entier)*
*OCI_FETCHSTATEMENT_BY_COLUMN (entier)*
*OCI_FETCHSTATEMENT_BY_ROW (entier)*
*OCI_ASSOC (entier)*
*OCI_NUM (entier)*
*OCI_BOTH (entier)*
*OCI_RETURN_NULLS (entier)*
*OCI_RETURN_LOBS (entier)*
*OCI_DTYPE_FILE (entier)*
*OCI_DTYPE_LOB (entier)*
*OCI_DTYPE_ROWID (entier)*
*OCI_D_FILE (entier)*
*OCI_D_LOB (entier)*
*OCI_D_ROWID (entier)*

## 6.27.7 Exemples

Exemple avec OCI8

```
 <?php
// par sergo@bacup.ru

// Utilisez l'option: OCI_DEFAULT pour exécuter la commande avec exécution
// retardée.
OCIExecute($stmt, OCI_DEFAULT);

// pour lire les données, après un fetch :

 $result = OCIResult($stmt, $n);
if (is_object ($result)) $result = $result->load();

// Pour les commandes INSERT et UPDATE utilisez :

$sql = "insert into table (field1, field2) values (field1 = 'value',
 field2 = empty_clob()) returning field2 into :field2";
OCIParse($conn, $sql);
$clob = OCINewDescriptor($conn, OCI_D_LOB);
OCIBindByName ($stmt, ":field2", &$clob, -1, OCI_B_CLOB);
OCIExecute($stmt, OCI_DEFAULT);
$clob->save ("some text");
OCICommit($conn);

?>
```

Vous pouvez facilement accéder aux procédures stockées, de la même façon que vous le feriez par ligne de commande :

Utilisation de procédures stockées

```
 <?php
// par webmaster@remoterealty.com
$sth = OCIParse ( $dbh, "begin sp_newaddress( :address_id, '$firstname',
 '$lastname', '$company', '$address1', '$address2', '$city', '$state',
 '$postalcode', '$country', :error_code );end;" );

// Cela appelle la procédure stockée sp_newaddress, avec la variable :address_id
 // pour les entrées/sorties et :error_code comme variable de sortie.
// Ensuite, vous faites les liaisons suivantes :

    OCIBindByName ( $sth, ":address_id", $addr_id, 10 );
   OCIBindByName ( $sth, ":error_code", $errorcode, 10 );
   OCIExecute ( $sth );

?>
```


- Introduction
- Pré-requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- ocibindbyname
- ocicancel
- ocicloselob
- ocicollappend
- ocicollassign
- ocicollassignelem
- ocicollgetelem
- ocicollmax
- ocicollsize
- ocicolltrim
- ocicolumnisnull
- ocicolumname
- ocicolumnprecision
- ocicolumnscale
- ocicolumnsize

- ocicolumntype
- ocicolumntyperaw
- ocicommit
- ocidefinebyname
- ocierror
- ociexecute
- ocifetch
- ocifetchinto
- ocifetchstatement
- ocifreecollection
- ocifreecursor
- ocifreedesc
- ocifreestatement
- ociinternaldebug
- ociloadlob
- ocilogoff
- ocilogon
- ocinewcollection
- ocinewcursor
- ocinewdescriptor
- ocinlogon
- ocinumcols
- ociparse
- ociplogon
- ociresult
- ocirollback
- ocirowcount
- ocisavelob
- ocisavelobfile
- ociserverversion
- ocisetprefetch
- ocistatementtype
- ociwritelobtofile
- ociwritetemporarylob

## 6.27.9 ocicancel() : Termine la lecture de ressources Oracle

bool **ocicancel** ( resource stmt )

ocicancel détruit les ressources liées au dernier résultat stmt . Si vous ne souhaitez plus lire d'informations dans ce résultat, utilisez cette fonction.

## 6.27.10 ocicloselob() : Ferme un descripteur de lob

bool **ocicloselob** ( )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.11 ocicollappend() : Ajoute un objet à une collection Oracle

bool **ocicollappend** ( string value )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.12 ocicollassign() : Assigne une collection depuis une autre collection Oracle

bool **ocicollassign** ( object from )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.13 ocicollassignelem() : Assigne l'élément val à une collection Oracle à l'index ndx

bool **ocicollassignelem** ( int ndx , string val )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.14 ocicollgetelem() : Lit la valeur d'un élément d'une collection Oracle à l'index ndx

string **ocicollgetelem** ( int ndx )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.15 ocicollmax() : Retourne la plus grande valeur d'une collection Oracle

int **ocicollmax** ( )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Pour un varray, c'est la taille maximale du tableau.

## 6.27.16 ocicollsize() : Retourne la taille d'une collection Oracle

int **ocicollsize** ( )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.17 ocicolltrim() : Supprime num éléments à la fin de la collection Oracle

bool **ocicolltrim** ( int num )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.18 ocicolumnisnull() : Teste si la valeur d'une colonne Oracle est NULL

bool **ocicolumnisnull** ( resource stmt , mixed col )

ocicolumnisnull retourne TRUE si la colonne column du résultat stmt est NULL. Vous pouvez utiliser le numéro de colonne (l'indexation des colonnes commence à 1) ou le nom de la colonne, pour le paramètre column .

## 6.27.19 ocicolumname() : Retourne le nom d'une colonne Oracle

string **ocicolumnname** ( resource stmt , int col )

ocicolumnname retourne le nom de la colonne numéro col (l'indexation des colonnes commence à 1).

Exemple avec ocicolumnname

```
<?php
   print "<HTML><PRE>\n";
   $conn = OCILogon("scott", "tiger");
   $stmt = OCIParse($conn,"select * from emp");
   OCIExecute($stmt);
   print "<TABLE BORDER=\"1\">";
   print "<TR>";
   print "<TH>Nom</TH>";
   print "<TH>Type</TH>";
   print "<TH>Taille</TH>";
   print "</TR>";
   $ncols = OCINumCols($stmt);
   for ( $i = 1; $i <= $ncols; $i++ ) {
       $column_name  = OCIColumnName($stmt,$i);
       $column_type  = OCIColumnType($stmt,$i);
       $column_size  = OCIColumnSize($stmt,$i);
       print "<TR>";
       print "<TD>$column_name</TD>";
       print "<TD>$column_type</TD>";
       print "<TD>$column_size</TD>";
       print "</TR>";
   }
   print "</TABLE>\n";
    OCIFreeStatement($stmt);
   OCILogoff($conn);
   print "</PRE>";
   print "</HTML>\n";
?>
```

Voir aussi ocinumcols , ocicolumntype et ocicolumnsize .

## 6.27.20 ocicolumnprecision() : Lit la précision d'une colonne Oracle

int **ocicolumnprecision** ( resource stmt , int col )

| Attention |
| --- |

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.27.21 ocicolumnscale() : Lit la taille d'une colonne Oracle

int **ocicolumnscale** ( resource stmt , int col )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.27.22 ocicolumnsize() : Retourne la taille d'une colonne Oracle

int **ocicolumnsize** ( resource stmt , mixed column )

ocicolumnsize retourne la taille de la colonne. Vous pouvez utiliser l'index de colonne (l'indexation commence à 1) ou le nom de la colonne dans le paramètre column .

Exemple avec ocicolumnsize

```
<?php
   print "<HTML><PRE>\n";
   $conn = OCILogon("scott", "tiger");
   $stmt = OCIParse($conn,"select * from emp");
   OCIExecute($stmt);
   print "<TABLE BORDER=\"1\">";
   print "<TR>";
   print "<TH>Name</TH>";
   print "<TH>Type</TH>";
   print "<TH>Length</TH>";
   print "</TR>";
   $ncols = OCINumCols($stmt);
   for ( $i = 1; $i <= $ncols; $i++ ) {
       $column_name  = OCIColumnName($stmt,$i);
       $column_type  = OCIColumnType($stmt,$i);
       $column_size  = OCIColumnSize($stmt,$i);
       print "<TR>";
       print "<TD>$column_name</TD>";
       print "<TD>$column_type</TD>";
       print "<TD>$column_size</TD>";
       print "</TR>";
   }
   print "</TABLE>";
   OCIFreeStatement($stmt);
   OCILogoff($conn);
   print "</PRE>";
   print "</HTML>\n";
?>
```

Voir aussi ocinumcols , ocicolumnname et ocicolumnsize .

## 6.27.23 ocicolumntype() : Retourne le type de données d'une colonne Oracle

mixed **ocicolumntype** ( resource stmt , int col )

ocicolumntype retourne le type de données de la colonne correspondant au numéro de colonne col dans le résultat stmt (les colonnes sont indexées à partir de 1).

Exemple avec ocicolumntype

```
<?php
   print "<HTML><PRE>\n";
   $conn = OCILogon("scott", "tiger");
   $stmt = OCIParse($conn,"select * from emp");
   OCIExecute($stmt);
   print "<TABLE BORDER=\"1\">";
   print "<TR>";
   print "<TH>Nom</TH>";
   print "<TH>Type</TH>";
```

```
    print "<TH>Longueur</TH>";
    print "</TR>";
    $ncols = OCINumCols($stmt);
    for ( $i = 1; $i <= $ncols; $i++ ) {
        $column_name  = OCIColumnName($stmt,$i);
        $column_type  = OCIColumnType($stmt,$i);
        $column_size  = OCIColumnSize($stmt,$i);
        print "<TR>";
        print "<TD>$column_name</TD>";
        print "<TD>$column_type</TD>";
        print "<TD>$column_size</TD>";
        print "</TR>";
    }
    print "</TABLE>\n";
     OCIFreeStatement($stmt);
    OCILogoff($conn);
    print "</PRE>";
    print "</HTML>\n";
 ?>
```

Voir aussi ocinumcols , ocicolumnname et ocicolumnsize .

## 6.27.24 ocicolumntyperaw() : Lit directement le type de colonne Oracle

mixed **ocicolumntyperaw** ( resource stmt , int col )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.25 ocicommit() : Valide les transactions Oracle en cours

bool **ocicommit** ( resource connection )

ocicommit valide toutes les transactions en cours sur la connexion Oracle connection . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

This example demonstrates how OCICommit is used.

OCICommit

```
 <?php
    // Connexion au serveur Oracle
    $conn = OCILogon('scott', 'tiger');

    // Analyse de la requête SQL
    $stmt = OCIParse($conn, "INSERT INTO employees (name, surname) VALUES ('Maxim', 'Maletsky')");

    // Exécution de la requête
    OCIExecute($stmt);

    // Validation de la transaction
    $committed = OCICommit($conn);

    // Vérification de la validation : si une erreur est survenue, afficher
    // le message d'erreur
    if(!$committed) {
        $error = OCIError($conn);
        echo 'Lavalidation a échoué. Oracle dit : ' . $error['message'];
    }

    // Fermeture de la connexion
    OCILogoff($conn);
?>
```

Voir aussi ocirollback .

## 6.27.26 ocidefinebyname() : Utilise une variable PHP pour la phase de définition, dans une commande SELECT Oracle

bool **ocidefinebyname** ( resource stmt , string column_name , mixed &variable , *int type* )

ocidefinebyname copie les valeurs issues de colonnes SQL column−name dans les variables PHP. Méfiez−vous des colonnes Oracle qui sont toutes en majuscule, tandis que dans les SELECT, vous pouvez aussi les écrire en minuscules. ocidefinebyname s'attends à ce que column−name soit en majuscules. Si vous définissez une variable qui n'existe pas dans la commande SELECT, vous ne serez pas prévenu par une erreur.

Si vous avez besoin de définir un type de données abstrait, tel que (LOB/ROWID/BFILE), vous devez lui allouer la mémoire avec ocinewdescriptor . Reportez−vous aussi à ocibindbyname .

Exemple avec ocidefinebyname

```
 <?php
$conn = OCILogon("scott","tiger");

$stmt = OCIParse($conn,"select empno, ename from emp");

/* La définition DOIT intervenir AVANT l'exécution! */

OCIDefineByName($stmt,"EMPNO",$empno);
OCIDefineByName($stmt,"ENAME",$ename);

OCIExecute($stmt);

while (OCIFetch($stmt)) {
    echo "empno :".$empno."\n";
    echo "ename :".$ename."\n";
}

OCIFreeStatement($stmt);
OCILogoff($conn);
?>
```

## 6.27.27 ocierror() : Retourne la dernière erreur de stmt|conn|global sur Oracle

array **ocierror** ( *resource stmt|conn|global* )

ocierror retourne la dernière erreur trouvée. Si l'option stmt|conn n'est pas fournie, la dernière erreur rencontrée est retournée. Si aucune erreur n'est trouvée, ocierror retourne FALSE . ocierror retourne l'erreur sous la forme d'un tableau associatif. Dans ce tableau, l'élément code contient le numéro d'erreur Oracle et message contient le message d'erreur.

## 6.27.28 ociexecute() : Exécute une commande SQL Oracle

bool **ociexecute** ( resource stmt , *int mode* )

ociexecute exécute une commande déjà préparée (voir ociparse ). L'option mode vous permet de spécifier le mode d'exécution (par défaut, il est à OCI_COMMIT_ON_SUCCESS ). Si vous ne voulez pas que la commande soit automatiquement validée, utilisez le mode OCI_DEFAULT .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.27.29 ocifetch() : Lit la prochaine ligne dans le résultat Oracle

bool **ocifetch** ( resource stmt )

ocifetch lit la prochaine ligne (d'une commande SELECT) et la place dans le buffer interne de résultat. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.27.30 ocifetchinto() : Retourne la ligne suivante d'un résultat Oracle dans un tableau

int **ocifetchinto** ( resource stmt , array &result , *int mode* )

ocifetchinto retourne la ligne suivante (pour une commande SELECT) dans le tableau result . ocifetchinto écrasera le contenu de result . Par défaut, result sera un tableau à index numérique, commençant à 1, et qui contiendra toute les colonnes qui ne sont pas NULL.

L'option mode vous permet de modifier le comportement par défaut de la fonction. Vous pouvez passer plusieurs modes simplement en les additionnant (i.e. OCI_ASSOC+OCI_RETURN_NULLS). Les modes valides sont :

- OCI_ASSOC retourne un tableau associatif.
- OCI_NUM retourne un tableau à index numérique (DEFAULT, valeur par défaut)
- OCI_RETURN_NULLS retourne les colonnes vides.
- OCI_RETURN_LOBS retourne la valeur des objets LOB plutôt que leur descripteur.

Exemple avec ocifetchinto

```
 <?php
$conn = ocilogon("username","password");

$query = "SELECT pommes FROM arbre";

$statement = OCIParse ($conn, $query);
OCIExecute ($statement);

while (OCIFetchInto ($statement, $row, OCI_ASSOC)) {
    print $row['pommes'];
}
?>
```

Voir aussi ocifetch et ociexecute .

## 6.27.31 ocifetchstatement() : Retourne toutes les lignes d'un résultat Oracle

int **ocifetchstatement** ( resource stmt , array &output , *int skip* , *int maxrows* , *int flags* )

ocifetchstatement retourne toutes les lignes d'un résultat dans le tableau variable. ocifetchstatement retourne le nombre de lignes retournées. skip est le nombre de lignes initiales à ignorer (par défaut, 0). maxrows est le nombre de lignes à lire, en commençant à la skip −ième ligne (par défaut, −1, qui représente toute les lignes).

flags représente les options disponibles pour cette fonction, et qui peuvent être la combinaison des constantes suivantes :

- OCI_FETCHSTATEMENT_BY_ROW
- OCI_FETCHSTATEMENT_BY_COLUMN (valeur par défaut)

- OCI_NUM
- OCI_ASSOC

Exemple avec ocifetchstatement

```
 <?php
$conn = OCILogon("scott","tiger");

$stmt = OCIParse($conn,"select * from emp");

OCIExecute($stmt);

$nrows = OCIFetchStatement($stmt,$results);
if ( $nrows > 0 ) {
   print "<TABLE BORDER=\"1\">\n";
   print "<TR>\n";
   while ( list( $key, $val ) = each( $results ) ) {
      print "<TH>$key</TH>\n";
   }
   print "</TR>\n";

   for ( $i = 0; $i < $nrows; $i++ ) {
      reset($results);
      print "<TR>\n";
      while ( $column = each($results) ) {
         $data = $column['value'];
         print "<TD>$data[$i]</TD>\n";
      }
      print "</TR>\n";
   }
   print "</TABLE>\n";
} else {
   echo "Aucune données trouvées<BR>\n";
}
print "$nrows lignes lues<BR>\n";

 OCIFreeStatement($stmt);
OCILogoff($conn);
?>
```

## 6.27.32 ocifreecollection() : Supprime une collection Oracle

bool **ocifreecollection** ( )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.27.33 ocifreecursor() : Libère toutes les ressources réservées par un résultat Oracle

bool **ocifreecursor** ( resource stmt )

ocifreecursor libère toutes les ressources réservées par le résultat Oracle stmt . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.27.34 ocifreedesc() : Détruit un pointeur de LOB Oracle

bool **ocifreedesc** ( )

ocifreedesc détruit le pointeur de LOB lob . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.27.35 ocifreestatement() : Libère toutes les ressources occupées par une commande Oracle

bool **ocifreestatement** ( resource stmt )

ocifreestatement libère toutes les ressources occupées par la commande stmt . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.27.36 ociinternaldebug() : Active ou désactive l'affichage des données de déboggage Oracle

void **ociinternaldebug** ( int onoff )

ociinternaldebug active ou désactive l'affichage des informations de déboggage. Pour les afficher, mettez onoff à 1, ou sinon mettez onoff à 0 pour les masquer.

## 6.27.37 ociloadlob() : Charge un LOB Oracle

string **ociloadlob** ( )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.38 ocilogoff() : Déconnecte d'un serveur Oracle

bool **ocilogoff** ( resource connection )

ocilogoff ferme la connexion Oracle connection .

Utiliser ocilogoff n'est pas nécessaire, car les connexions non persistantes seront automatiquement fermées à la fin du script. Voyez aussi le chapitre sur la libération des ressources .

## 6.27.39 ocilogon() : Etablit une connexion à un serveur Oracle

resource **ocilogon** ( string username , string password , *string db* )

ocilogon retourne un identifiant de connexion, nécessaire à la plupart des fonctions oci. Le paramètre optionnel db peut contenir le nom local du serveur Oracle, ou bien son nom dans la table d'entrées tnsnames.ora à laquelle vous voulez vous connecter. Si le troisième paramètre n'est pas spécifié, PHP utilisera la variable d'environnement ORACLE_SID (instance Oracle) ou TWO_TASK (tnsnames.ora) pour déterminer la base à laquelle se connecter.

Les connexions sont partagées, à l'intérieur d'une même page avec ocilogon . Cela signifie que COMMIT et ROLLBACK s'appliquent à toutes les transactions commencées à l'intérieur d'une même page, même si vous avez créé de multiples connexions.

Les connexions sont partagées au niveau de la page, lorsque vous utilisez ocilogon . Cela signifie que les validations et annulations s'appliquent à toutes les transactions de la page, même si vous avez créé de multiples connexions à un même serveur Oracle.

Cet exemple montre comment les connexions sont partagées :

Exemple avec ocilogon

```
 <?php
print "<HTML><PRE>";
$db = "";

$c1 = ocilogon("scott","tiger",$db);
$c2 = ocilogon("scott","tiger",$db);

function create_table($conn)
{ $stmt = ociparse($conn,"create table scott.hallo (test varchar2(64))");
  ociexecute($stmt);
  echo $conn." a créé une table\n\n";
}

function drop_table($conn)
{ $stmt = ociparse($conn,"drop table scott.hallo");
  ociexecute($stmt);
  echo $conn." a effacé une table\n\n";
}

function insert_data($conn)
{ $stmt = ociparse($conn,"insert into scott.hallo
             values('$conn' || ' ' || to_char(sysdate,'DD-MON-YY HH24:MI:SS'))");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a inséré dans hallo\n\n";
}

function delete_data($conn)
{ $stmt = ociparse($conn,"delete from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a effacé dans hallo\n\n";
}

function commit($conn)
{ ocicommit($conn);
  echo $conn." validé\n\n";
}

function rollback($conn)
{ ocirollback($conn);
  echo $conn." annulé\n\n";
}

function select_data($conn)
{ $stmt = ociparse($conn,"select * from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn."----selection\n\n";
  while (ocifetch($stmt))
    echo $conn." <".ociresult($stmt,"TEST").">\n\n";
  echo $conn."----fin\n\n";
}

create_table($c1);
insert_data($c1);   // Insertion avec c1
insert_data($c2);   // Insertion avec c2

select_data($c1);   // Resultats des insertions
select_data($c2);

rollback($c1);      // Annulation sur c1

select_data($c1);   // Les deux insertions ont été annulées
select_data($c2);

insert_data($c2);   // Insertion avec c2
commit($c2);        // Validation avec c2

select_data($c1);   // Le résultat de l'insertion via c2

delete_data($c1);   // Effacement de toutes les lignes avec c1
select_data($c1);   // Aucune ligne lue
select_data($c2);   // Aucune ligne lue
commit($c1);        // Validation avec c1

select_data($c1);   // Aucune ligne lue
select_data($c2);   // Aucune ligne lue

drop_table($c1);
print "</PRE></HTML>";
?>
```

Voir aussi ociplogon et ocinlogon .

## 6.27.40 ocinewcollection() : Initialise une nouvelle collection

object **ocinewcollection** ( resource connection , string tdo , *string schema* )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.41 ocinewcursor() : Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références

resource **ocinewcursor** ( resource conn )

ocinewcursor alloue un nouveau pointeur de commande, pour la connexion conn .

Utiliser un REF CURSOR issue d'une procédure enregistrée

```
 <?php
// supposons que votre procédure stockée info.output retourne un curseur de référence dans :data

$conn = OCILogon("scott","tiger");
$curs = OCINewCursor($conn);
$stmt = OCIParse($conn,"begin info.output(:data); end;");

ocibindbyname($stmt,"data",&$curs,-1,OCI_B_CURSOR);
ociexecute($stmt);
ociexecute($curs);

while (OCIFetchInto($curs,&$data)) {
    var_dump($data);
}

 OCIFreeStatement($stmt);
OCIFreeCursor($curs);
OCILogoff($conn);
?>
```

Utiliser un REF CURSOR issue d'une commande SELECT

```
 <?php
print "<HTML><BODY>";
$conn = OCILogon("scott","tiger");
$count_cursor = "CURSOR(select count(empno) num_emps from emp " .
                "where emp.deptno = dept.deptno) as EMPCNT from dept";
$stmt = OCIParse($conn,"select deptno,dname,$count_cursor");

ociexecute($stmt);
print "<TABLE BORDER=\"1\">";
print "<TR>";
print "<TH>DEPT NAME</TH>";
print "<TH>DEPT #</TH>";
print "<TH># EMPLOYEES</TH>";
print "</TR>";

while (OCIFetchInto($stmt,&$data,OCI_ASSOC)) {
    print "<TR>";
    $dname  = $data["DNAME"];
    $deptno = $data["DEPTNO"];
    print "<TD>$dname</TD>";
    print "<TD>$deptno</TD>";
    ociexecute($data["EMPCNT"]);
    while (OCIFetchInto($data["EMPCNT"],&$subdata,OCI_ASSOC)) {
        $num_emps = $subdata["NUM_EMPS"];
        print  "<TD>$num_emps</TD>";
    }
    print "</TR>";
}
print "</TABLE>";
print "</BODY></HTML>";
OCIFreeStatement($stmt);
OCILogoff($conn);
?>
```

## 6.27.42 ocinewdescriptor() : Initialise un nouveau pointeur vide de LOB/FILE

object **ocinewdescriptor** ( resource connection , *int type* )

ocinewdescriptor alloue l'espace nécessaire pour stocker un descripteur, ou un pointeur de LOB. Les valeurs acceptées pour type sont OCI_D_FILE, OCI_D_LOB et OCI_D_ROWID. Pour les pointeurs de LOB, les méthodes load, save, et savefile sont associées avec le pointeur. Pour les pointeurs de type BFILE, seule la méthode load existe. Voyez le second exemple pour une illustration.

ocinewdescriptor

```
<?php
   /* Ce script est fait pour être appelé dans un formulaire HTML
    * Il attends les variables $user, $password, $table, $where, et $commitsize
    * Le script efface alors les lignes sélectionnées avec ROWID et valide
    * l'effacement après chaque groupe de $commitsize lignes.
    * (Utilisez avec prudence, car il n'y a pas d'annulation possible).
    */
   $conn = OCILogon($user, $password);
   $stmt = OCIParse($conn,"select rowid from $table $where");
   $rowid = OCINewDescriptor($conn,OCI_D_ROWID);
   OCIDefineByName($stmt,"ROWID",&$rowid);
   OCIExecute($stmt);
   while ( OCIFetch($stmt) ) {
      $nrows = OCIRowCount($stmt);
      $delete = OCIParse($conn,"delete from $table where ROWID = :rid");
      OCIBindByName($delete,":rid",&$rowid,-1,OCI_B_ROWID);
      OCIExecute($delete);
      print "$nrows\n";
      if ( ($nrows % $commitsize) == 0 ) {
          OCICommit($conn);
      }
   }
   $nrows = OCIRowCount($stmt);
   print "$nrows deleted...\n";
   OCIFreeStatement($stmt);
   OCILogoff($conn);
?>


<?php
   /* Ce script est fait pour être appelé depuis un formulaire HTML.
    * Il attends les variables $user, $password, $table, $where, et $commitsize,
    * données par le formulaire. Le script efface
    * les lignes sélectionnées avec ROWID est valide les transactions
    * à chaque jeu de $commitsize lignes. (Attention : il n'y plus d'annulation) */
  if(!isset($lob_upload) || $lob_upload == 'none'){
?>
<form action="upload.php" method="post" enctype="multipart/form-data">
Fichier à télécharger : <input type="file" name="lob_upload"><br>
<input type="submit" value="Envoyer"> - <input type="reset">
</form>
<?php
  } else {

     // $lob_upload contient le nom temporaire du fichier téléchargé

     // voir aussi la section fonctionnalités sur les téléchargements de
     // fichiers si vous voulez utiliser des téléchargements sécurisés

     $conn = OCILogon($user, $password);
     $lob = OCINewDescriptor($conn, OCI_D_LOB);
     $stmt = OCIParse($conn,"insert into $table (id, the_blob)
                values(my_seq.NEXTVAL, EMPTY_BLOB()) returning the_blob into :the_blob");
     OCIBindByName($stmt, ':the_blob', &$lob, -1, OCI_B_BLOB);
     OCIExecute($stmt, OCI_DEFAULT);
     if($lob->savefile($lob_upload)){
        OCICommit($conn);
        echo "Blob bien téléchargé\n";
     }else{
        echo "Impossible de télécharger le Blob\n";
     }
     OCIFreeDesc($lob);
     OCIFreeStatement($stmt);
     OCILogoff($conn);
  }
?>
```

Exemple avec ocinewdescriptor

```
<?php
   /* Appel d'une procédure PL/SQL stockée qui prend un clobs
    * en entrée (PHP 4 >= 4.0.6).
```

```
     * Exemple de signateure de procédure stockée PL/SQL :
    *
    * PROCEDURE save_data
    *   Nom de l'argument              Type                     In/Out Default?
    *   ------------------------------ ------------------------ ------ --------
    *   KEY                            NUMBER(38)               IN
    *   DATA                           CLOB                     IN
    *
    */

   $conn = OCILogon($user, $password);
   $stmt = OCIParse($conn, "begin save_data(:key, :data); end;");
   $clob = OCINewDescriptor($conn, OCI_D_LOB);
   OCIBindByName($stmt, ':key', $key);
   OCIBindByName($stmt, ':data', $clob, -1, OCI_B_CLOB);
   $clob->WriteTemporary($data);
   OCIExecute($stmt, OCI_DEFAULT);
   OCICommit($conn);
   $clob->close();
   $clob->free();
   OCIFreeStatement($stmt);
?>
```

## 6.27.43 ocinlogon() : Etablit une nouvelle correction avec un serveur Oracle

resource **ocinlogon** ( string username , string password , *string db* )

ocinlogon crée une nouvelle connexion à un serveur Oracle et s'identifie. Le paramètre optionnel db peut contenir le nom local du serveur Oracle, ou bien son nom dans la table d'entrées tnsnames.ora à laquelle vous voulez vous connecter. Si le troisième paramètre n'est pas spécifié, PHP utilisera la variable d'environnement ORACLE_SID (instance Oracle) ou TWO_TASK (tnsnames.ora) pour déterminer la base à laquelle se connecter.

ocinlogon force le serveur à établir une nouvelle connexion. Cette fonction ne doit être utilisée que si vous voulez isoler un ensemble de transactions. Par défaut, les connexions sont partagées au niveau de la page, si vous utilisez la fonction ocinlogon ou bien au niveau du processus web, si vous utilisez ociplogon . Si vous avez de multiples connexions ouvertes avec ocinlogon , les validations et annulations ne s'appliquent qu'à la connexion spécifée.

L'exemple ci−dessous montre l'utilisation des connexions séparées.

Exemple avec ocinlogon

```
 <?php
print "<HTML><PRE>";
$db = "";

$c1 = ocilogon("scott","tiger",$db);
$c2 = ocinlogon("scott","tiger",$db);


function create_table($conn)
{ $stmt = ociparse($conn,"create table scott.hallo (test varchar2(64))");
  ociexecute($stmt);
  echo $conn." a créé une table\n\n";
}

function drop_table($conn)
{ $stmt = ociparse($conn,"drop table scott.hallo");
  ociexecute($stmt);
  echo $conn." a effacé une table\n\n";
}

function insert_data($conn)
{ $stmt = ociparse($conn,"insert into scott.hallo
             values('$conn' || ' ' || to_char(sysdate,'DD-MON-YY HH24:MI:SS'))");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a inséré dans hallo\n\n";
}

function delete_data($conn)
{ $stmt = ociparse($conn,"delete from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn." a effacé dans hallo\n\n";
}
```

```
function commit($conn)
{ ocicommit($conn);
  echo $conn." validé\n\n";
}

function rollback($conn)
{ ocirollback($conn);
  echo $conn." annulé\n\n";
}

function select_data($conn)
{ $stmt = ociparse($conn,"select * from scott.hallo");
  ociexecute($stmt,OCI_DEFAULT);
  echo $conn."----selection\n\n";
  while (ocifetch($stmt))
    echo $conn." <".ociresult($stmt,"TEST").">\n\n";
  echo $conn."----fin\n\n";
}

create_table($c1);
insert_data($c1);

select_data($c1);
select_data($c2);

rollback($c1);

select_data($c1);
select_data($c2);

insert_data($c2);
commit($c2);

select_data($c1);

delete_data($c1);
select_data($c1);
select_data($c2);
commit($c1);

select_data($c1);
select_data($c2);

drop_table($c1);
print "</PRE></HTML>";
?>
```

Voir aussi ocilogon et ociplogon .

## 6.27.44 ocinumcols() : Retourne le nombre de colonnes dans un résultat

int **ocinumcols** ( resource stmt )

ocinumcols retourne le nombre de colonnes dans le résultat stmt .

Exemple avec ocinumcols

```
<?php
   print "<HTML><PRE>\n";
   $conn = OCILogon("scott", "tiger");
   $stmt = OCIParse($conn,"select * from emp");
   OCIExecute($stmt);
   while ( OCIFetch($stmt) ) {
       print "\n";
       $ncols = OCINumCols($stmt);
       for ( $i = 1; $i <= $ncols; $i++ ) {
           $column_name  = OCIColumnName($stmt,$i);
           $column_value = OCIResult($stmt,$i);
           print $column_name . ': ' . $column_value . "\n";
       }
       print "\n";
   }
   OCIFreeStatement($stmt);
   OCILogoff($conn);
   print "</PRE>";
   print "</HTML>\n";
?>
```

## 6.27.45 ociparse() : Analyse une requête SQL avec Oracle

resource **ociparse** ( resource conn , string query )

ociparse analyse la requête query sur la connexion conn , et retourne TRUE si la requête query est valide, et FALSE , si ce n'est pas le cas. query peut être n'importe quelle requête SQL.

## 6.27.46 ociplogon() : Ouvre une connexion persistante à un serveur Oracle

resource **ociplogon** ( string username , string password , *string db* )

ociplogon crée une connexion persistante à un serveur Oracle 8 et s'authentifie. Le paramètre optionnel db peut contenir le nom local du serveur Oracle, ou bien son nom dans la table d'entrées tnsnames.ora à laquelle vous voulez vous connecter. Si le troisième paramètre n'est pas spécifié, PHP utilisera la variable d'environnement ORACLE_SID (instance Oracle) ou TWO_TASK (tnsnames.ora) pour déterminer la base à laquelle se connecter.

Voir aussi ocilogon et ocinlogon .

## 6.27.47 ociresult() : Retourne la valeur d'une colonne dans un résultat Oracle

mixed **ociresult** ( resource statement , mixed col )

ociresult retourne les données de la colonne column dans la ligne courante (voir ocifetch ). ocifetch retournera tout les types, sauf les types abstraits (ROWIDs, LOBs et FILEs).

## 6.27.48 ocirollback() : Annule les transactions Oracle en cours

bool **ocirollback** ( resource connection )

ocirollback annule les transactions en cours sur la connexion Oracle connection . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi ocicommit .

## 6.27.49 ocirowcount() : Retourne le nombre de lignes affectées par Oracle

int **ocirowcount** ( resource stmt )

ocirowcount retourne le nombre de lignes affectées par la commande de modification (UPDATE) statement . Cette fonction ne vous indiquera pas le nombre de lignes retournées par un SELECT : il faut que les lignes aient été modifiées.

Exemple avec ocirowcount

```
<?php
   print "<HTML><PRE>";
   $conn = OCILogon("scott","tiger");
```

```
    $stmt = OCIParse($conn,"create table emp2 as select * from emp");
    OCIExecute($stmt);
    print OCIRowCount($stmt) . " rows inserted.<BR>";
    OCIFreeStatement($stmt);
    $stmt = OCIParse($conn,"delete from emp2");
    OCIExecute($stmt);
    print OCIRowCount($stmt) . " rows deleted.<BR>";
    OCICommit($conn);
    OCIFreeStatement($stmt);
    $stmt = OCIParse($conn,"drop table emp2");
    OCIExecute($stmt);
    OCIFreeStatement($stmt);
    OCILogOff($conn);
    print "</PRE></HTML>";
?>
```

## 6.27.50 ocisavelob() : Sauve un LOB Oracle

bool **ocisavelob** ( )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.51 ocisavelobfile() : Sauve un fichier LOB Oracle

bool **ocisavelobfile** ( )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.27.52 ociserverversion() : Retourne une chaîne contenant les informations de version du serveur Oracle

string **ociserverversion** ( resource conn )

ociserverversion retourne une chaîne contenant les informations de version du serveur Oracle, identifié par la connexion conn .

Exemple avec ociserverversion

```
<?php
  $conn = OCILogon("scott","tiger");
  print "Version du serveur Oracle : " . OCIServerVersion($conn);
  OCILogOff($conn);
?>
```

## 6.27.53 ocisetprefetch() : Indique le nombre de lignes qui doivent être pré−lues par Oracle

bool **ocisetprefetch** ( resource stmt , int rows )

ocisetprefetch spécifie le nombre des premières lignes qui doivent être pré−lues à rows pour la connexion Oracle stmt . La valeur par défaut de rows est 1 ligne.

## 6.27.54 ocistatementtype() : Retourne le type de commande OCI

string **ocistatementtype** ( resource stmt )

ocistatementtype retourne une des valeurs suivantes :

1. SELECT
2. UPDATE
3. DELETE
4. INSERT
5. CREATE
6. DROP
7. ALTER
8. BEGIN
9. DECLARE
10. UNKNOWN

Exemple avec ocistatementtype

```
<?php
   print "<HTML><PRE>";
   $conn = OCILogon("scott","tiger");
   $sql  = "delete from emp where deptno = 10";

   $stmt = OCIParse($conn,$sql);
   if ( OCIStatementType($stmt) == "DELETE" ) {
       die "Nous n'êtes pas autorisés à effacer des lignes dans cette table<BR>";
   }

   OCILogoff($conn);
   print "</PRE></HTML>";
?>
```

## 6.27.55 ociwritelobtofile() : Ecrit un LOB Oracle dans un fichier

bool **ociwritelobtofile** ( *string filename* , *int start* , *int length* )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.27.56 ociwritetemporarylob() : Ecrit un blob temporaire

bool **ociwritetemporarylob** ( string var , *int lob_type* )

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

# 6.28 Bufferisation de sortie

## 6.28.1 Introduction

Les fonctions de bufferisation de sortie vous permettent de contrôler quand les données ont été envoyées par le script. Cela peut être utile dans certaines situations, notamment si vous devez envoyer des en−têtes au navigateur après avoir envoyé des données. Ces fonctions n'affectent pas les en−têtes envoyées par la fonction header ou les cookies envoyés par setcookie . Seules les fonctions telles que echo et les données entre blocs PHP sont affectées.

## 6.28.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.28.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.28.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| output_buffering | "0" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| output_handler | NULL | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| implicit_flush | "0" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*output_buffering booléen / entier*
: Vous pouvez activer la bufferisation de sortie pour tous les fichiers avec cette directive, en lui passant la valeur de On . Si vous souhaitez limiter la taille du buffer à une certaine taille, vous pouvez alors indiquer un nombre maximum d'octets à la place de On . Par exemple, output_buffering=4096 ).

*output_handler chaîne de caractères*
: Vous pouvez rediriger le résultat de tous vos scripts à une fonction avant leur envoi au navigateur. Par exemple, si vous configurez output_handler à mb_output_handler , l'encodage des caractères sera adaptée de manière transparente. Configurer une telle fonction active automatiquement la bufferisation de sortie.

| Note |
|---|
| Vous ne pouvez pas utiliser simultanément mb_output_handler avec ob_inconv_handler et non plus ob_gzhandler et zlib.output_compression . |

*implicit_flush booléen*
: FALSE par défaut. En changeant cette valeur pour TRUE vous indiquez à PHP que le buffer de sortie doit être vidé automatiquement après chaque fonction d'affichage. Cela revient à appeler la fonction flush après chaque appel à print ou echo et pour tous les blocs HTML .

Lorsque vous utilisez PHP en environnement web, activer cette option a de sérieuse implications et généralement, cela n'est conseillé que pour les débogage. Cette valeur est par défaut à TRUE lorsque PHP fonctionne en mode CLI SAPI .

Voir aussi ob_implicit_flush .

## 6.28.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.28.6 Constantes prédefinies

Cette extension ne définit aucune constante.

## 6.28.7 Exemples

Exemple de bufferisation de sortie

```
<?php
ob_start();
echo "Bonjour\n";
setcookie ("nom_du_cookie", "valeur_du_cookie");
ob_end_flush();
?>
```

Dans l'exemple ci−dessus, la fonction echo est stockée dans un buffer jusqu'à l'appel de la fonction ob_end_flush . Dans le même temps, l'appel à setcookie a réussi à créer un cookie, sans générer d'erreur. (D'habitude, vous devez envoyer les en−têtes avant les données).

Note

Lorsque vous passez de PHP 4.1 ou 4.2 à 4.3, assurez vous que implict_flush est à OFF dans votre php.ini , sinon la fonction ob_start ne masquera pas les affichages engendrés.

## 6.28.8 Voir aussi

Voir aussi header et setcookie .


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- Voir aussi
- flush
- ob_clean
- ob_end_clean
- ob_end_flush
- ob_flush
- ob_get_clean
- ob_get_contents
- ob_get_flush

- ob_get_length
- ob_get_level
- ob_get_status
- ob_gzhandler
- ob_implicit_flush
- ob_list_handlers
- ob_start
- output_add_rewrite_var
- output_reset_rewrite_vars

## 6.28.10 ob_clean() : Efface le buffer de sortie

void **ob_clean** ( void )

ob_clean vide le buffer de sortie sans l'envoyer au navigateur.

ob_clean ne détruit pas le contenu du buffer de sortie comme peut le faire ob_end_clean .

Voir aussi ob_flush , ob_end_flush et ob_end_clean .

## 6.28.11 ob_end_clean() : Détruit les données du buffer de sortie, et éteint la bufferisation de sortie

bool **ob_end_clean** ( void )

ob_end_clean vide le contenu du premier buffer de sortie, et désactive la bufferisation de sortie. Si vous voulez traiter le contenu du buffer, vous devrez appeler ob_get_contents avant ob_end_clean , car le buffer est détruit par ob_end_flush . La fonction TRUE lorsqu'elle a réussi à détruire correctement le buffer, et FALSE sinon. Les raisons d'un tel échec sont la bufferisation de sortie pouvaient ne pas être activé, ou que pour une raison quelconque, le buffer n'a pu être détruit.

L'exemple suivant montre comment se débarraser de tous les buffers de sortie :

Exemple avec ob_end_clean

```
<?php
while (@ob_end_clean());
?>
```

Note

Si la fonction échoue, elle génère une note E_NOTICE.

La valeur booléenne retournée par la fonction a été ajoutée en PHP 4.2.0.

Voir aussi ob_start , ob_flush et ob_get_contents .

## 6.28.12 ob_end_flush() : Envoie les données du buffer de sortie, et éteint la bufferisation de sortie

bool **ob_end_flush** ( void )

ob_end_flush envoie le contenu du buffer de sortie (s'il existe) et éteint la bufferisation de sortie. Si

vous voulez continuer à manipuler la valeur du buffer, vous pouvez appeler ob_get_contents avant ob_end_flush car le contenu du buffer est détruit après un appel à ob_end_flush . La fonction retourne TRUE lorsqu'elle a réussi à envoyer tout le buffer, et FALSE otherwise. Les raisons d'échec sont que vous pourriez avoir appelé la fonction sans avoir de buffer actif, pour que pour une raison quelconque, le buffer n'a pu être effacé (possible pour un buffer spécial).

L'exemple ci−dessous montre une méthode simple pour vider tous les buffers :

Exemple avec ob_end_flush

```
<?php
while (@ob_end_flush());
?>
```

Note

Si la fonction échoue, elle émet une note E_NOTICE.

Le booléen retourné a été ajouté en PHP 4.2.0.

Voir aussi ob_start , ob_get_contents , ob_flush et ob_end_clean .

## 6.28.13 ob_flush() : Envoie le buffer de sortie

void **ob_flush** ( void )

**ob_flush** envoie le contenu du buffer de sortie (si il y en a un). Si vous voulez contrôler le contenu du buffer, vous devez appeler la fonction ob_get_contents avant ob_flush car le contenu du buffer est effacé après l'appel de ob_flush .

ob_flush ne détruit pas le contenu du buffer de sortie comme peut le faire ob_end_clean .

Voir aussi ob_get_contents , ob_clean , ob_end_flush et ob_end_clean .

## 6.28.14 ob_get_clean() : Lit le contenu courant du buffer de sortie puis l'efface

string **ob_get_clean** ( void )

ob_get_clean va retourner le contenu du buffer de sortie, et terminer la session de bufferisation. Si la bufferisation n'était pas activée, alors ob_get_clean retourne FALSE . ob_get_clean exécute successivement ob_get_contents et ob_end_clean .

Exemple avec ob_get_clean

```
<?php

ob_start();

print "Bonjour le monde";

$out = ob_get_clean();
$out = strtolower($out);

var_dump($out);

/* Notre exemple va afficher :

string(16) "Bonjour le monde"

*/
?>
```

Voir aussi ob_start et ob_get_contents .

## 6.28.15 ob_get_contents() : Retourne le contenu du buffer de sortie

string **ob_get_contents** ( void )

ob_get_contents retourne le contenu du buffer de sortie si la bufferisation est activée, ou FALSE sinon.

Voir aussi ob_start et ob_get_length .

## 6.28.16 ob_get_flush() : Vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon.

string **ob_get_flush** ( void )

ob_get_flush vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon. ob_get_flush retourne FALSE si la mise en tampon n'est pas active.

Note

ob_get_flush est similaire à ob_end_flush , sauf que cette fonction retourne le tampon en tant que chaîne.

Exemple avec ob_get_flush

```
<?php
// avec output_buffering=On
print_r(ob_list_handlers());

// sauvegarde le tampon dans un fichier
$buffer = ob_get_flush();
file_put_contents('buffer.txt', $buffer);

print_r(ob_list_handlers());
?>
```

L'exemple précédent va afficher :

```
Array
(
[0] => default output handler
)
Array
(
)
```

Voir aussi ob_end_clean , ob_end_flush et ob_list_handlers .

## 6.28.17 ob_get_length() : Retourne la longueur du contenu du buffer de sortie

int **ob_get_length** ( void )

ob_get_length retourne la longueur du contenu du buffer de sortie si la bufferisation est activée, et FALSE sinon.

Voir aussi ob_start et ob_get_contents .

## 6.28.18 ob_get_level() : Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie

int **ob_get_level** ( void )

**ob_get_level** retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie.

Voir aussi ob_start et ob_get_contents .

## 6.28.19 ob_get_status() : Lit le statut du buffer de sortie

array **ob_get_status** ( *bool full_status* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

ob_get_status retourne le statut actuel du système de bufferisation de sortie : un tableau qui contient le statut du buffer, ou bien FALSE en cas d'erreur.

Voir aussi ob_get_level .

## 6.28.20 ob_gzhandler() : Fonction de callback pour la compression automatique des buffers

string **ob_gzhandler** ( string buffer , *int mode* )

ob_gzhandler est destinée à être utilisée comme fonction de callback par ob_start pour faciliter l'envoi de données compressées aux navigateurs qui supportent les pages compressées. Avant que ob_gzhandler envoie les données compressées, il détermine les types d'encodage qui sont supportés par le navigateur ("gzip", "deflate" ou aucun) et retourne le contenu des buffers de manière appropriée. Tous les navigateurs sont traités, car c'est aux navigateurs d'envoyer une en−tête indiquant les types de pages supportés.

| Note |
| --- |
| Vous ne pouvez pas utiliser simultanément ob_gzhandler et ini.zlib.output_compression . De plus, notez bien que ini.zlib.output_compression est préférable à ob_gzhandler . |

Exemple d'envoi de page compressée avec ob_gzhandler

```
 <?php
  ob_start("ob_gzhandler");
?>
<HTML>
<BODY>
<P>Ceci devrait être une page compressée.
</HTML>
</BODY>
```

Voir aussi ob_start et ob_end_flush .

## 6.28.21 ob_implicit_flush() : Active/désactive l'envoi implicite

void **ob_implicit_flush** ( *int flag* )

ob_implicit_flush active/désactive l'envoi implicite (si flag est fourni. Par défaut, il est activé). L'envoi implicite signifie que toute fonction qui envoie des données au navigateur verra ses données envoyées immédiatement (la fonction flush est appelée automatiquement).

Une fois que l'envoi implicite est désactivé, le buffer de sortie ne sera envoyé qu'au moment de l'appel de ob_end_flush .

Voir aussi flush , ob_start et ob_end_flush .

## 6.28.22 ob_list_handlers() : Liste les gestionnaires d'affichage utilisés

array **ob_list_handlers** ( void )

Cette fonction retourne un tableau avec le gestionnaire d'affichage en cours d'utilisation (si il existe). Si output_buffering est activé, ob_list_handlers retournera un tableau avec comme valeur d'entrée : "default output handler".

Exemple avec ob_list_handlers

```
<?php

// avec output_buffering=On
print_r(ob_list_handlers());
ob_end_flush();

ob_start("ob_gzhandler");
print_r(ob_list_handlers());
ob_end_flush();
?>
```

L'exemple va afficher :

```
Array
(
[0] => default output handler
)
Array
(
[0] => ob_gzhandler
)
```

Voir aussi ob_end_clean , ob_end_flush , ob_get_flush et ob_start .

## 6.28.23 ob_start() : Enclenche la bufferisation de sortie

bool **ob_start** ( *callback output_callback* )

ob_start démarre la bufferisation de sortie. Tant qu'elle est enclenchée, aucune donnée, hormis les en−têtes, n'est envoyée au navigateur, mais temporairement mise en buffer.

Le contenu de ce buffer peut être copié dans une chaîne avec la fonction ob_get_contents . Pour afficher le contenu de ce buffer, utilisez ob_end_flush . Au contraire, ob_end_clean effacera le contenu de ce buffer.

Une fonction optionnelle de callback peut être spécifiée en troisième argument. ob_start prend une chaîne comme paramètre, et retourne une chaîne. Elle sera appelée par ob_end_flush ou lorsque

le buffer sera envoyé au navigateur à la fin du script et recevra le contenu du buffer de sortie. Lorsque la fonction output_callback est appelée, elle doit retourner un nouveau contenu pour le buffer de sortie : celui−ci sera envoyé au navigateur.

| Note |
| --- |
| En PHP 4.0.4, ob_gzhandler a été introduit pour faciliter l'envoi de fichier compressé avec gz aux navigateurs web qui supportent les pages compressées. ob_gzhandler détermine le type d'encodage accepté par un navigateur, et retourne le contenu le plus adéquat. |

Les buffers de sortie sont gérés par pile, c'est−à−dire que vous pouvez appeler plusieurs fois ob_start simultanément. Assurez−vous que vous appelez ob_end_flush suffisamment souvent. Si plusieurs fonctions de callback sont actives, les contenus seront filtrés séquentiellement, dans l'ordre d'emboîtement.

Exemple de gestion de sortie avec fonction utilisateur

```
 <?php
function callback($buffer) {
  // remplace toutes les pommes par des carottes
  return (ereg_replace("pommes de terre", "carottes", $buffer));
}
ob_start("callback");
?>
<html>
<body>
<p>C'est comme comparer des carottes et des pommes de terre.
</body>
</html>
<?php
 ob_end_flush();
?>
```

va afficher :

```
 <html>
<body>
<p>C'est comme comparer des carottes et des carottes.
</body>
</html>
```

Voir aussi ob_get_contents , ob_end_flush , ob_end_clean , ob_implicit_flush et ob_gzhandler .

## 6.28.24 output_add_rewrite_var() : Ajoute une règle de réécriture d'URL

bool **output_add_rewrite_var** ( string name , string value )

output_add_rewrite_var réécrit les URLs avec la variable donnée.

| Note |
| --- |
| Cette fonction met en tampon la sortie. |

Exemple avec output_add_rewrite_var

```
 <?php
output_add_rewrite_var('var', 'value');

// un lien
echo '<a href="file.php">link</a>';

// un formulaire
echo '<form action="script.php" method="post">
<input type="text" name="var2" />
</form>';

print_r(ob_list_handlers());
?>
```

Cet exemple va afficher :

```
<a href="file.php?var=value">link</a>

<form action="script.php" method="POST">
<input type="hidden" name="var" value="value" />
<input type="text" name="var2" />
</form>

Array
(
[0] => URL-Rewriter
)
```

Voir aussi ob_reset_rewrite_vars , ob_flush et ob_list_handlers .

## 6.28.25 output_reset_rewrite_vars() : Annule la réécriture d'URL

bool **output_reset_rewrite_vars** ( void )

output_reset_rewrite_vars efface la règle de réécriture d'URLs et annule les changements faits par output_add_rewrite_var et/ou par session_start et qui sont encore en tampon.

Exemple avec output_reset_rewrite_vars

```
<?php
session_start();
output_add_rewrite_var('var', 'value');

echo '<a href="file.php">link</a>';
ob_flush();

output_reset_rewrite_vars();
echo '<a href="file.php">link</a>';
?>
```

L'exemple va afficher :

```
<a href="file.php?PHPSESSID=xxxlink</a>
<a href="file.php">link</a>
```

Voir aussi output_add_rewrite_var , ob_flush , ob_list_handlers et session_start .

# 6.29 PDF

## 6.29.1 Introduction

Les fonctions PDF permettent de créer des fichiers PDF en utilisant la librairie PDFlib créée par Thomas Merz .

La documentation de cette section est uniquement destinée à être une introduction aux fonctions disponibles, et ne doit pas être considérée comme une référence exaustive. Reportez vous à la documentation inclue dans la distribution de PDFlib pour plus de détails et d'explications. Elle fournit une excellente référence des capacités de PDFlib, et contient la documentation la plus à jour.

Toutes les fonctions de la PDFlib et du module PHP ont des noms et des arguments identiques. Vous aurez à comprendre quelques concepts basiques de PDF et PostScript pour l'utiliser efficacement. Toutes les longueurs et coordonnées sont mesurées en points PostScript. Il y a généralement 72 points PostScript pour faire un pouche (2.54 cm), mais cela dépend de la résolution de sortie. Reportez vous à la documentation PDFlib inclue dans la distribution pour plus de détails sur le système de coordonnées.

Notez que la plupart des fonctions PDF requièrent une ressource pdf object comme premier paramètre. Voyez les exemples ci−dessous pour plus de détails.

| Note |
| --- |
| Si vous êtes intéressés par des alternatives gratuites pour générer des PDF, sans passer par des librairies PDF, voyez cette entrée de la FAQ . |

## 6.29.2 Pré−requis

PDFlib est disponible en téléchargement à http://www.pdflib.com/products/pdflib/index.html , mais impose l'achat d'une license pour l'utiliser commercialement. Les librairies JPEG et TIFF sont nécessaires pour faire fonctionner cette librairie.

### 6.29.2.1 Problèmes avec les anciennes versions de PDFlib

Toutes les versions des PHP 4 éditées après le 9 Mars 2000 ne suportent pas les versions de PDFlib plus anciennes que PDFlib 3.0.

PDFlib 3.0 est plus récent est supporté par PHP 3.0.19 et plus récent.

## 6.29.3 Installation

Pour inclure le support de la PDFlib dans votre PHP , il faut compiler PHP avec l'option −−with−pdflib[=DIR] . DIR est le dossier d'isntallation de PDFlib, et par défaut, il vaut /usr/local . De plus, vous pouvez aussi spécifier les dossiers des libariries jpeg, tiff, et png que vous voulez utiliser avec la PDFlib. Ceci est optionnel pour PDFlib 4.x. Pour cela, ajoutes les lignes suivantes à votre script de configuration : −−with−jpeg−dir[=DIR] , −−with−png−dir[=DIR] et −−with−tiff−dir[=DIR] .

Lorsque vous utilisez les versions 3.x de PDFlib, vous devez configurer PDFlib avec l'option −−enable−shared−pdflib .

## 6.29.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.29.5 Confusion entre les vieilles versions de PDFLib

Depuis le début du support de PDF sous PHP, (commençant avec la version PDFLib 0.6), il y a eu des milliers de modifications dans les API de PDFLib. La plupart de ces modifications ont été suivies par PHP, et parfois même au prix de modifications des API PHP. Depuis la version 3.x, ces API semblent s'être stabilisées, et PHP 4 a adoptée cette version comme le minimum nécessaire pour supporter PDF. En conséquence de quoi, un grand nombre de fonction vont disparaître, ou être remplacée. Le support de PDFLib 0.6 est complétement abandonné. La liste suivante indique quelles sont les fonctions obsolètes en PHP 4.02, et qui devraient être remplacées par de nouvelles versions.

| Ancienne fonction | Remplacement |
|---|---|
| pdf_put_image | Désormais inutile. |
| pdf_execute_image | Désormais inutile. |
| pdf_get_annotation | pdf_get_bookmark en utilisant les mêmes paramètres. |
| pdf_get_font | pdf_get_value en passant "font" comme second paramètre. |
| pdf_get_fontsize | pdf_get_value en passant "fontsize" comme second paramètre. |
| pdf_get_fontname | pdf_get_parameter en passant "fontname" comme second paramètre. |
| pdf_set_info_creator | pdf_set_info en passant "Creator" comme second paramètre. |
| pdf_set_info_title | pdf_set_info en passant "Title" comme second paramètre. |
| pdf_set_info_subject | pdf_set_info en passant "Subject" comme second paramètre. |
| pdf_set_info_author | pdf_set_info en passant "Author" comme second paramètre. |
| pdf_set_info_keywords | pdf_set_info en passant "Keywords" comme second paramètre. |
| pdf_set_leading | pdf_set_value en passant "leading" comme second paramètre. |
| pdf_set_text_rendering | pdf_set_value en passant "textrendering" comme second paramètre. |
| pdf_set_text_rise | pdf_set_value en passant "textrise" comme second paramètre. |
| pdf_set_horiz_scaling | pdf_set_value en passant "horizscaling" comme second paramètre. |
| pdf_set_text_matrix | Désormais abandonnée. |
| pdf_set_char_spacing | pdf_set_value en passant "charspacing" comme second paramètre. |
| pdf_set_word_spacing | pdf_set_value en passant "wordspacing" comme second paramètre. |
| pdf_set_transition | pdf_set_parameter en passant "transition" comme second paramètre. |
| pdf_open | pdf_new suivi d'un appel à pdf_open_file |
| pdf_set_font | pdf_findfont suivi d'un appel à pdf_setfont |
| pdf_set_duration | pdf_set_value en passant "duration" comme second paramètre. |
| pdf_open_gif | pdf_open_image_file en passant "gif" comme second paramètre. |
| pdf_open_jpeg | pdf_open_image_file en passant "jpeg" comme second paramètre. |
| pdf_open_tiff | pdf_open_image_file en passant "tiff" comme second paramètre. |
| pdf_open_png | pdf_open_image_file en passant "png" comme second paramètre. |
| pdf_get_image_width | pdf_get_value en passant "imagewidth" comme second paramètre et l'image en troisième paramètre. |
| pdf_get_image_height | pdf_get_value en passant "imageheight" comme second paramètre et l'image en troisième paramètre. |

## 6.29.6 Exemples

La plupart des fonctions sont simples d'emploi. Le plus difficile est probablement de créer un fichier PDF simple. L'exemple suivant devrait vous mettre sur les rails. Il crée un fichier test.pdf d'une page. La page contient du texte "Times Roman outlined", de taille de 30pt. Le texte est aussi souligné.

Création d'un document PDF avec PDFlib

```
<?php
$pdf = pdf_new();
pdf_open_file($pdf, "test.pdf");
```

```
pdf_set_info($pdf, "Author", "Uwe Steinmann");
pdf_set_info($pdf, "Title", "Test for PHP wrapper of PDFlib 2.0");
pdf_set_info($pdf, "Creator", "See Author");
pdf_set_info($pdf, "Subject", "Testing");
pdf_begin_page($pdf, 595, 842);
pdf_add_outline($pdf, "Page 1");
pdf_set_font($pdf, "Times-Roman", 30, "host");
pdf_set_value($pdf, "textrendering", 1);
pdf_show_xy($pdf, "Times Roman outlined", 50, 750);
pdf_moveto($pdf, 50, 740);
pdf_lineto($pdf, 330, 740);
pdf_stroke($pdf);
pdf_end_page($pdf);
pdf_close($pdf);
pdf_delete($pdf);
echo "<A HREF=getpdf.php>terminé</A>";
?>
```

Le script getpdf.php ne fait que retourner le document PDF.

```
 <?php
$len = filesize($filename);
header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=foo.pdf");
readfile($filename);
?>
```

La distribution de la PDFlib contient des exemples plus complexes, qui créent un page avec une horloge à aiguille. Ici, nous utilisons la création de pages en mémoire, pour éviter l'utilisation de fichiers temporaires. L'exemple a été adapté à PHP. (Cet exemple est disponible dans la documentation CLibPDF .)

Exemple pdfclock de PDFlib

```
 <?php
$radius = 200;
$margin = 20;
$pagecount = 10;

$pdf = pdf_new();

if (!pdf_open_file($pdf, "")) {
    print error;
    exit;
};

pdf_set_parameter($pdf, "warning", "true");

pdf_set_info($pdf, "Creator", "pdf_clock.php");
pdf_set_info($pdf, "Author", "Uwe Steinmann");
pdf_set_info($pdf, "Title", "Analog Clock");

while($pagecount-- > 0) {
    pdf_begin_page($pdf, 2 * ($radius + $margin), 2 * ($radius + $margin));

    pdf_set_parameter($pdf, "transition", "wipe");
    pdf_set_value($pdf, "duration", 0.5);

    pdf_translate($pdf, $radius + $margin, $radius + $margin);
    pdf_save($pdf);
    pdf_setrgbcolor($pdf, 0.0, 0.0, 1.0);

    /* les marques des minutes */
    pdf_setlinewidth($pdf, 2.0);
    for ($alpha = 0; $alpha < 360; $alpha += 6) {
        pdf_rotate($pdf, 6.0);
        pdf_moveto($pdf, $radius, 0.0);
        pdf_lineto($pdf, $radius-$margin/3, 0.0);
        pdf_stroke($pdf);
    }

    pdf_restore($pdf);
    pdf_save($pdf);

    /* les marques des 5 minutes */
    pdf_setlinewidth($pdf, 3.0);
    for ($alpha = 0; $alpha < 360; $alpha += 30) {
         pdf_rotate($pdf, 30.0);
        pdf_moveto($pdf, $radius, 0.0);
        pdf_lineto($pdf, $radius-$margin, 0.0);
        pdf_stroke($pdf);
    }

    $ltime = getdate();

    /* l'aiguille des heures */
```

```
    pdf_save($pdf);
    pdf_rotate($pdf,-(($ltime['minutes']/60.0)+$ltime['hours']-3.0)*30.0);
    pdf_moveto($pdf, -$radius/10, -$radius/20);
    pdf_lineto($pdf, $radius/2, 0.0);
    pdf_lineto($pdf, -$radius/10, $radius/20);
    pdf_closepath($pdf);
    pdf_fill($pdf);
    pdf_restore($pdf);

    /* l'aiguille des minutes */
    pdf_save($pdf);
    pdf_rotate($pdf,-(($ltime['seconds']/60.0)+$ltime['minutes']-15.0)*6.0);
    pdf_moveto($pdf, -$radius/10, -$radius/20);
    pdf_lineto($pdf, $radius * 0.8, 0.0);
    pdf_lineto($pdf, -$radius/10, $radius/20);
    pdf_closepath($pdf);
    pdf_fill($pdf);
    pdf_restore($pdf);

    /* l'aiguille des secondes */
    pdf_setrgbcolor($pdf, 1.0, 0.0, 0.0);
    pdf_setlinewidth($pdf, 2);
    pdf_save($pdf);
    pdf_rotate($pdf, -(($ltime['seconds'] - 15.0) * 6.0));
    pdf_moveto($pdf, -$radius/5, 0.0);
    pdf_lineto($pdf, $radius, 0.0);
    pdf_stroke($pdf);
    pdf_restore($pdf);

    /* Dessin d'un petit cercle dans le centre */
    pdf_circle($pdf, 0, 0, $radius/30);
    pdf_fill($pdf);

    pdf_restore($pdf);

    pdf_end_page($pdf);

    // pour voir la différence
    sleep(1);
}

pdf_close($pdf);

$buf = pdf_get_buffer($pdf);
$len = strlen($buf);

header("Content-type: application/pdf");
header("Content-Length: $len");
header("Content-Disposition: inline; filename=foo.pdf");
print $buf;

pdf_delete($pdf);
?>
```

## 6.29.7 Voir aussi

| Note |
| --- |
| Une alternative au module module PDFlib est le module de FastIO : ClibPDF. Voyez la section ClibPDF pour plus de détails. Notez que ClibPDF disponse d'une API différente de PDFlib. |

- Introduction
- Pré-requis
- Installation
- Configuration à l'exécution
- Confusion entre les vieilles versions de PDFLib
- Exemples
- Voir aussi
- pdf_add_annotation
- pdf_add_bookmark
- pdf_add_launchlink
- pdf_add_locallink
- pdf_add_note
- pdf_add_outline
- pdf_add_pdflink
- pdf_add_thumbnail

- pdf_add_weblink
- pdf_arc
- pdf_arcn
- pdf_attach_file
- pdf_begin_page
- pdf_begin_pattern
- pdf_begin_template
- pdf_circle
- pdf_clip
- pdf_close_image
- pdf_close_pdi_page
- pdf_close_pdi
- pdf_close
- pdf_closepath_fill_stroke
- pdf_closepath_stroke
- pdf_closepath
- pdf_concat
- pdf_continue_text
- pdf_curveto
- pdf_delete
- pdf_end_page
- pdf_end_pattern
- pdf_end_template
- pdf_endpath
- pdf_fill_stroke
- pdf_fill
- pdf_findfont
- pdf_get_buffer
- pdf_get_font
- pdf_get_fontname
- pdf_get_fontsize
- pdf_get_image_height
- pdf_get_image_width
- pdf_get_majorversion
- pdf_get_minorversion
- pdf_get_parameter
- pdf_get_pdi_parameter
- pdf_get_pdi_value
- pdf_get_value
- pdf_initgraphics
- pdf_lineto
- pdf_makespotcolor
- pdf_moveto
- pdf_new
- pdf_open_ccitt
- pdf_open_file
- pdf_open_gif
- pdf_open_image_file
- pdf_open_image
- pdf_open_jpeg
- pdf_open_memory_image
- pdf_open_pdi_page
- pdf_open_pdi
- pdf_open_png

- pdf_open_tiff
- pdf_open
- pdf_place_image
- pdf_place_pdi_page
- pdf_rect
- pdf_restore
- pdf_rotate
- pdf_save
- pdf_scale
- pdf_set_border_color
- pdf_set_border_dash
- pdf_set_border_style
- pdf_set_char_spacing
- pdf_set_duration
- pdf_set_font
- pdf_set_horiz_scaling
- pdf_set_info_author
- pdf_set_info_creator
- pdf_set_info_keywords
- pdf_set_info_subject
- pdf_set_info_title
- pdf_set_info
- pdf_set_leading
- pdf_set_parameter
- pdf_set_text_matrix
- pdf_set_text_pos
- pdf_set_text_rendering
- pdf_set_text_rise
- pdf_set_value
- pdf_set_word_spacing
- pdf_setcolor
- pdf_setdash
- pdf_setflat
- pdf_setfont
- pdf_setgray_fill
- pdf_setgray_stroke
- pdf_setgray
- pdf_setlinecap
- pdf_setlinejoin
- pdf_setlinewidth
- pdf_setmatrix
- pdf_setmiterlimit
- pdf_setpolydash
- pdf_setrgbcolor_fill
- pdf_setrgbcolor_stroke
- pdf_setrgbcolor
- pdf_show_boxed
- pdf_show_xy
- pdf_show
- pdf_skew
- pdf_stringwidth
- pdf_stroke
- pdf_translate

## 6.29.9 pdf_add_bookmark() : Ajoute un signet dans la page courante

int **pdf_add_bookmark** ( resource pdfdoc , string text , *int parent* , *int open* )

pdf_add_bookmark ajoute un signet imbriqué dans la page courante, sous parent , ou bien dans le signet global si parent = 0. pdf_add_bookmark retourne une ressource de signet, qui peut servir de paramètre parent Si open vaut 1, le signet fils sera visible. Si open vaut 0, le signet fils sera invisible.

Exemple avec pdf_add_bookmark

```
 <?php
// create a new PDF

$pdf = pdf_new();
pdf_open_file($pdf);
pdf_set_info($pdf, "Author","Bob Nijman");

// begin a new page
pdf_begin_page($pdf, 300, 300);

// add a top-level bookmark
$bookmark = pdf_add_bookmark($pdf, "People");

// add a nested bookmark
pdf_add_bookmark($pdf, "Rasmus", $bookmark);

// and some text
pdf_set_font($pdf, "Helvetica", 20, "host");
$text = "This is R's page";
$width = pdf_stringwidth($pdf, $text);
pdf_set_text_pos($pdf, (300-$width)/2, 100);
pdf_show($pdf, $text);

// close the page and the PDF
pdf_end_page($pdf);
 pdf_close($pdf);

?>
```

## 6.29.10 pdf_add_launchlink() : Ajoute une annotation de lancement dans la page PDF courante

bool **pdf_add_launchlink** ( resource pdfdoc , float llx , float lly , float urx , float ury , string filename )

pdf_add_launchlink ajoute une annotation de lancement de fichier (type arbitraire de fichiers). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_add_locallink .

## 6.29.11 pdf_add_locallink() : Ajoute une annotation de lien dans la page PDF courante

bool **pdf_add_locallink** ( resource pdfdoc , float lowerleftx , float lowerlefty , float upperrightx , float upperrighty , int page , string dest )

pdf_add_locallink ajoute une annotation de lien dans la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

dest est le zoom de la page de destination. Il peut valoir retain , fitpage , fitwidth , fitheight ou fitbbox .

Voir aussi pdf_add_launchlink .

## 6.29.12 pdf_add_note() : Ajoute une annotation dans la page PDF courante

bool **pdf_add_note** ( resource pdfdoc , float llx , float lly , float urx , float ury , string contents , string title , string icon , int open )

pdf_add_note ajoute une annotation dans la page courante. Le paramèter icon peut prendre l'une des valeurs suivantes : " comment ", " insert ", " note ", " paragraph ", " newparagraph " " key" et " help ". Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.13 pdf_add_outline() : Obsolète : Ajoute un signet dans la page courante

Obsolète.

Voir pdf_add_bookmark .

## 6.29.14 pdf_add_pdflink() : Ajoute une annotation sur un lien vers un fichier dans la page PDF courante

bool **pdf_add_pdflink** ( resource pdfdoc , float bottom_left_x , float bottom_left_y , float up_right_x , float up_right_y , string filename , int page , string dest )

pdf_add_pdflink ajoute une annotation sur un lien vers un fichier dans la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_add_locallink et pdf_add_weblink .

## 6.29.15 pdf_add_thumbnail() : Ajoute une miniature sur la page PDF courante

bool **pdf_add_thumbnail** ( resource pdfdoc , int image )

pdf_add_thumbnail ajoute une miniature sur la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.16 pdf_add_weblink() : Ajoute un lien web sur la page PDF courante

bool **pdf_add_weblink** ( resource pdfdoc , float lowerleftx , float lowerlefty , float upperrightx , float upperrighty , string url )

pdf_add_weblink ajoute un lien web url sur la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.17 pdf_arc() : Dessine un arc de cercle PDF dans le sens anti−horaire

bool **pdf_arc** ( resource pdfdoc , float x , float y , float r , float alpha , float beta )

pdf_arc dessine un arc de cercle dans le sens anti−horaire, dont le centre est le point de coordonnées ( x , y ) et de rayon r , depuis l'angle alpha jusqu'à beta , exprimés en degrés. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_arcn et pdf_circle .

## 6.29.18 pdf_arcn() : Dessine un arc de cercle dans le sens horaire

bool **pdf_arcn** ( resource pdfdoc , float x , float y , float r , float alpha , float beta )

pdf_arc dessine un arc de cercle dans le sens horaire dont le centre est le point de coordonnées ( x , y ) et de rayon r , depuis l'angle alpha jusqu'à beta , exprimés en degrés. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_arc et pdf_circle .

## 6.29.19 pdf_attach_file() : Ajoute un fichier attaché à la page PDF

bool **pdf_attach_file** ( resource pdfdoc , float llx , float lly , float urx , float ury , string filename , string description , string author , string mimetype , string icon )

**pdf_attach_file** ajoute un fichier attaché à la page courante. icon peut prendre l'une des valeurs suivantes : graph , paperclip , pushpin ou tag . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Seule la version complète d'Acrobat sera capable d'afficher les attachements. Toutes les autres versions d'Acrobat n'afficheront qu'un point d'interrogation. |

## 6.29.20 pdf_begin_page() : Initialise une nouvelle page de document PDF

bool **pdf_begin_page** ( resource pdfdoc , float width , float height )

**pdf_begin_page** ajoute une nouvelle page au document. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Les paramètres width et height sont spécifiés en points, qui représentent 1/72 d'un pouce (2.54 cm).

| nom | taille |
| --- | --- |
| A0 | 2380 par 3368 |
| A1 | 1684 par 2380 |
| A2 | 1190 par 1684 |
| A3 | 842 par 1190 |
| A4 | 595 par 842 |
| A5 | 421 par 595 |
| A6 | 297 par 421 |
| B5 | 501 par 709 |

| | |
|---|---|
| letter (8.5" par 11") | 612 par 792 |
| legal (8.5" par 14") | 612 par 1008 |
| ledger (17" par 11") | 1224 par 792 |
| 11" par 17" | 792 par 1224 |

Voir aussi pdf_end_page .

## 6.29.21 pdf_begin_pattern() : Initialise un nouveau pattern PDF

int **pdf_begin_pattern** ( resource pdfdoc , float width , float height , float xstep , float ystep , int painttype )

**pdf_begin_pattern** initialise un nouveau pattern, et retourne une ressource. width et height définissent le cadre du pattern. xstep et ystep représentent les offsets des répétitions. painttype =1 indique que le pattern dispose de ses propres couleurs, tandis que la valeur 2 indique que la couleur courante est utilisée lorsque le pattern est appliqué.

Voir aussi pdf_end_pattern .

## 6.29.22 pdf_begin_template() : Initialise un nouveau template PDF

int **pdf_begin_template** ( resource pdfdoc , float width , float height )

pdf_begin_template initialise un nouveau template.

## 6.29.23 pdf_circle() : Dessine un cercle dans un document PDF

bool **pdf_circle** ( resource pdfdoc , float x , float y , float r )

pdf_circle dessine un cercle de centre ( x , y ) et de rayon r . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_arc et pdf_arcn .

## 6.29.24 pdf_clip() : Modifie le chemin de clipping PDF

bool **pdf_clip** ( resource pdfdoc )

pdf_clip utilise le chemin courant comme chemin de clipping. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.25 pdf_close_image() : Ferme une image dans un document PDF

void **pdf_close_image** ( resource pdfdoc , int image )

pdf_close_image ferme l'image image , ouverte avec l'une des fonctions images : pdf_open_image , pdf_open_gif , pdf_open_jpg , pdf_open_png , pdf_open_image_file , pdf_open_tiff et pdf_open_memory_image .

## 6.29.26 pdf_close_pdi_page() : Ferme la page PDF

bool **pdf_close_pdi_page** ( resource pdfdoc , int pagehandle )

pdf_close_pdi_page ferme la page, et libère toutes les ressources liées à cette page. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.27 pdf_close_pdi() : Ferme le fichier PDF d'entrée

bool **pdf_close_pdi** ( resource pdfdoc , int dochandle )

pdf_close_pdi ferme toutes les pages ouvertes, et referme le document PDF utilsée en entrée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_open_pdi .

## 6.29.28 pdf_close() : Ferme le fichier PDF

bool **pdf_close** ( resource pdfdoc )

pdf_close ferme le fichier PDF généré, et libère toutes les ressources qui y était rattachées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_new .

## 6.29.29 pdf_closepath_fill_stroke() : Termine le chemin, dessine les bords et remplit la forme

bool **pdf_closepath_fill_stroke** ( resource pdfdoc )

pdf_closepath_fill_stroke termine le chemin, dessine les bords et remplit la forme. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_closepath et pdf_closepath_stroke .

## 6.29.30 pdf_closepath_stroke() : Termine le chemin et dessine les bords

bool **pdf_closepath_stroke** ( resource pdfdoc )

pdf_closepath_stroke termine le chemin et dessine les bords. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_closepath et pdf_closepath_fil_stroke .

## 6.29.31 pdf_closepath() : Termine le chemin PDF courant

bool **pdf_closepath** ( resource pdfdoc )

pdf_closepath termine le chemin courant. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_closepath_stroke et pdf_closepath_fil_stroke .

### 6.29.32 pdf_concat() : Concatène une matrice avec le CTM

bool **pdf_concat** ( resource pdfdoc , float a , float b , float c , float d , float e , float f )

pdf_concat concatène une matrice avec le CTM. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

### 6.29.33 pdf_continue_text() : Affiche du texte à la prochaine ligne PDF

bool **pdf_continue_text** ( resource pdfdoc , string text )

pdf_continue_text affiche le texte text à la prochaine ligne. L'espacement entre les lignes est déterminé avec le paramètre leading . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

### 6.29.34 pdf_curveto() : Dessine une courbe de Bezier

bool **pdf_curveto** ( resource pdfdoc , float x1 , float y1 , float x2 , float y2 , float x3 , float y3 )

pdf_curveto dessine une courbe de Bezier à partir du point courant, et en utilisant trois points de contrôle. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

### 6.29.35 pdf_delete() : Efface un objet PDF

bool **pdf_delete** ( resource pdfdoc )

pdf_delete efface l'objet PDF > pdf_object , et libère toutes les ressources. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_new .

### 6.29.36 pdf_end_page() : Termine la page PDF courante

bool **pdf_end_page** ( resource pdfdoc )

pdf_end_page termine la page courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_begin_page .

## 6.29.37 pdf_end_pattern() : Termine le pattern PDF

bool **pdf_end_pattern** ( resource pdfdoc )

pdf_end_pattern termine la définition du pattern. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_begin_pattern .

## 6.29.38 pdf_end_template() : Termine le template PDF

bool **pdf_end_template** ( resource pdfdoc )

pdf_end_template termine la définition du template. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.39 pdf_endpath() : Obsolète : Termine le chemin courant

Obsolète. Utilsiez une des fonctions de pinceau, remplissage ou de clip.

## 6.29.40 pdf_fill_stroke() : Remplit et passe le pinceau sur le chemin PDF courant

bool **pdf_fill_stroke** ( resource pdfdoc )

pdf_fill_stroke remplit le chemin courant avec la couleur courant, en applicant le pinceau courant (stroke). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_setcolor .

## 6.29.41 pdf_fill() : Remplit le chemin PDF courant avec la couleur courante

bool **pdf_fill** ( resource pdfdoc )

pdf_fill_stroke remplit le chemin courant avec la couleur courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_setcolor .

## 6.29.42 pdf_findfont() : Prépare une police pour utilisation ultérieure avec pdf_setfont

int **pdf_findfont** ( resource pdfdoc , string fontname , string encoding , *int embed* )

**pdf_findfont** prépare la police fontname pour utilisation ultérieure avec pdf_setfont . Les dimensions seront chargées, et si embed est non nul, le fichier de police sera vérifié mais pas

utilisé. encoding peut prendre une des valeurs suivantes : builtin , macroman , winansi , host , ou bien un encodage définit par l'utilisateur, ou encore le nom d'une CMap.

pdf_findfont retourne une ressource de police, ou bien FALSE en cas d'erreur.

Exemple avec pdf_findfont

```
 <?php

$font = pdf_findfont($pdf, "Times New Roman", "winansi", 1);
if ($font) {
    pdf_setfont($pdf, $font, 10);
}

?>
```

## 6.29.43 pdf_get_buffer() : Lit le tampon contenant le fichier PDF généré

string **pdf_get_buffer** ( resource pdfdoc )

pdf_get_buffer lit le contenu du tampon PDF. Le résultat doit etre utilisé par le client avant d'appeler toute autre fonction PDFlib.

## 6.29.44 pdf_get_font() : Obsolète: charge une police

Obsolète.

Voir pdf_get_value .

## 6.29.45 pdf_get_fontname() : Obsolète : lit le nom de la police

Obsolète.

Voir pdf_get_parameter .

## 6.29.46 pdf_get_fontsize() : Obsolète : gère les polices

Obsolète.

Voir pdf_get_value .

## 6.29.47 pdf_get_image_height() : Retourne la hauteur d'une image

string **pdf_get_image_height** ( int pdf_object , int image )

pdf_get_image_height est obsolète. Utilisez plutôt pdf_get_value .

## 6.29.48 pdf_get_image_width() : Retourne la largeur d'une image

string **pdf_get_image_width** ( int pdf_object , int image )

pdf_get_image_width est obsolète. Utilisez plutôt pdf_get_value .

### 6.29.49 pdf_get_majorversion() : Retourne le numéro de version majeur de PDFlib

int **pdf_get_majorversion** ( void )

pdf_get_majorversion retourne le numéro de version majeur de PDFlib.

Voir aussi pdf_get_minorversion .

### 6.29.50 pdf_get_minorversion() : Retourne le numéro de version mineure de PDFlib

int **pdf_get_minorversion** ( void )

pdf_get_majorversion retourne le numéro de version mineure de PDFlib.

Voir aussi pdf_get_majorversion .

### 6.29.51 pdf_get_parameter() : Lit certains paramètres

string **pdf_get_parameter** ( resource pdfdoc , string key , *float* *modifier* )

pdf_get_parameter lit certains paramètres au format textuel.

### 6.29.52 pdf_get_pdi_parameter() : Lit des paramètres textuels dans le document PDI

string **pdf_get_pdi_parameter** ( resource pdfdoc , string key , int document , int page , int index )

pdf_get_pdi_parameter lit des paramètres textuels dans le document PDI.

### 6.29.53 pdf_get_pdi_value() : Lit des paramètres numériques dans le document PDF d'entrée

string **pdf_get_pdi_value** ( resource pdfdoc , string key , int doc , int page , int index )

pdf_get_pdi_value lit des paramètres numériques dans le document PDF d'entrée. Les paramètres lus sont de type numérique.

### 6.29.54 pdf_get_value() : Lit des paramètres numériques

float **pdf_get_value** ( resource pdfdoc , string key , *float* *modifier* )

**pdf_get_value** lit des paramètres numériques du document pdf_object , identifié par key .

## 6.29.55 pdf_initgraphics() : Remet à zéro l'environnement graphique PDF

bool **pdf_initgraphics** ( resource pdfdoc )

pdf_initgraphics remet à leur valeur intiale toutes les couleurs et les paramètres grahpiques. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.56 pdf_lineto() : Dessine une ligne PDF

bool **pdf_lineto** ( resource pdfdoc , float x , float y )

pdf_lineto dessine une ligne entre le point courant le point de coordonnées ( x , y ). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.57 pdf_makespotcolor() : Place un point de couleur PDF

bool **pdf_makespotcolor** ( resource pdfdoc , string spotname )

pdf_makespotcolor place un point de couleur, ayant le nom de spotname et la couleur courante. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_setcolor .

## 6.29.58 pdf_moveto() : Place le point courant PDF

bool **pdf_moveto** ( resource pdfdoc , float x , float y )

**pdf_moveto** place le point courant aux coordonnées ( x , y ). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Le point courant graphique et le point courant de texte sont différents. Voyez pdf_set_text_pos pour déplacer le point courant de texte. |

## 6.29.59 pdf_new() : Crée un nouvel objet PDF

resource **pdf_new** ( )

pdf_new crée un nouvel objet PDF, en utilisant le système de gestion d'erreur et celui de gestion de la mémoire.

Voir aussi pdf_close .

## 6.29.60 pdf_open_ccitt() : Ouvre une image contenant des données brutes CCITT

int **pdf_open_ccitt** ( resource pdfdoc , string filename , int width , int height , int bitreverse , int k , int blackls1 )

pdf_open_ccitt ouvre une image brute CCITT. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.61 pdf_open_file() : Ouvre un nouveau fichier PDF

bool **pdf_open_file** ( resource pdfdoc , *string filename* )

pdf_open_file crée un nouveau fichier PDF dans le fichier filename . Si filename est vide, le document PDF sera envoyé en mémoire. Le résultat pourra alors être lu avec la fonction pdf_get_buffer function. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

L'exemple suivant montre comment créer un document PDF en mémoire et comment l'afficher correctement.

Création d'un document PDF en mémoire

```
<?php

$pdf = pdf_new();

pdf_open_file($pdf);
pdf_begin_page($pdf, 595, 842);
pdf_set_font($pdf, "Times-Roman", 30, "host");
pdf_set_value($pdf, "textrendering", 1);
pdf_show_xy($pdf, "A PDF document created in memory!", 50, 750);
pdf_end_page($pdf);
pdf_close($pdf);

$data = pdf_get_buffer($pdf);

header("Content-type: application/pdf");
header("Content-disposition: inline; filename=test.pdf");
header("Content-length: " . strlen($data));

echo $data;

?>
```

## 6.29.62 pdf_open_gif() : Obsolète : Ouvre une image GIF

Obsolète.

Voir pdf_open_image ,

## 6.29.63 pdf_open_image_file() : Lit une image depuis un fichier

int **pdf_open_image_file** ( resource pdfdoc , string imagetype , string filename , *string stringparam* , *string intparam* )

pdf_open_image_file ouvre une image depuis le fichier filename . Les formats supportés sont jpeg , tiff , gif , et png . stringparam vaut soit "", mask , masked ou page . intparam vaut soit 0, soit la ressource d'image appliquée au masque, soit la ressource de page.

## 6.29.64 pdf_open_image() : Ouvre une image

int **pdf_open_image** ( resource pdf−document , string imagetype , string source , string data , long length , int width , int height , int components , int bpc , string params )

**pdf_open_image** permet d'accéder à des images de différents formats. Les formats supportés sont : jpeg , ccitt , raw . Les sources supportées sont memory , fileref , url . length est uniquement utilisé si imagetype vaut raw . params est uniquement utilisé si imagetype vaut ccitt .

## 6.29.65 pdf_open_jpeg() : Obsolète : Ouvre une image JPEG

Cette fonction est obsolète, utilisez plutôt pdf_open_image .

## 6.29.66 pdf_open_memory_image() : Ouvre une image créée en mémoire par PHP

int **pdf_open_memory_image** ( resource pdfdoc , resource image )

pdf_open_memory_image prend l'image image créée par PHP avec les fonctions d'images, et l'intègre dans l'objet PDF pdf_object . Cette fonction retourne une ressource d'image PDF.

Inclure une image PHP en PDF

```
<?php
$im = ImageCreate(100, 100);
$col = ImageColorAllocate($im, 80, 45, 190);
ImageFill($im, 10, 10, $col);
$pim = pdf_open_memory_image($pdf, $im);
ImageDestroy($im);
pdf_place_image($pdf, $pim, 100, 100, 1);
pdf_close_image($pdf, $pim);
?>
```

Voir aussi pdf_close_image et pdf_place_image .

## 6.29.67 pdf_open_pdi_page() : Prépare une page

int **pdf_open_pdi_page** ( resource pdfdoc , int dochandle , int pagenumber , string pagelabel )

pdf_open_pdi_page prépare une page pour utilisation ultérieure avec pdf_place_pdi_page

## 6.29.68 pdf_open_pdi() : Ouvre un fichier PDF

int **pdf_open_pdi** ( resource pdfdoc , string filename , string stringparam , int intparam )

pdf_open_pdi ouvre un fichier PDF existant, pour utilisation ultérieure.

Voir aussi pdf_close_pdi .

## 6.29.69 pdf_open_png() :  Obsolète : ouvre une image PNG

Obsolète.

Reportez−vous à pdf_open_image .

## 6.29.70 pdf_open_tiff() : Obsolète : Ouvre une image TIFF

Cette fonction est obsolète, utilisez plutôt pdf_open_image .

## 6.29.71 pdf_open() : Obsolète : Ouvre un nouvel objet PDF

Cette fonction est obsolète, utilisez de préférence pdf_new avec pdf_open_file .

## 6.29.72 pdf_place_image() : Place une image dans la page

bool **pdf_place_image** ( resource pdfdoc , int image , float x , float y , float scale )

pdf_place_image place l'image image , dans la page courante, au point de coordonnées ( x , y ), et la met à l'échelle scale . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.73 pdf_place_pdi_page() : Place une page dans le document

bool **pdf_place_pdi_page** ( resource pdfdoc , int page , float x , float y , float sx , float sy )

**pdf_place_pdi_page** place une page PDI, dans la page courante, aux coordonnées ( x , y ), et la met à l'échelle. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.74 pdf_rect() : Dessine un rectangle

bool **pdf_rect** ( resource pdfdoc , float x , float y , float width , float height )

**pdf_rect** dessine un rectangle de coin inférieur gauche aux coordonnées ( x , y ), et avec la largeur width et la hauteur height . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.75 pdf_restore() : Rétablit l'ancien environnement graphique PDF

bool **pdf_restore** ( resource pdfdoc )

pdf_restore rétablit l'ancien environnement graphique, sauvé avec pdf_save . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.76 pdf_rotate() : Configure la rotation

bool **pdf_rotate** ( resource pdfdoc , float phi )

pdf_rotate fait pivoter le système de coordonnées de phi degrés. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.77 pdf_save() : Sauve l'environnement graphique courant

bool **pdf_save** ( resource pdfdoc )

pdf_save sauve l'environnement graphique courant. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.78 pdf_scale() : Configure l'échelle du document

bool **pdf_scale** ( resource pdfdoc , float x−scale , float y−scale )

pdf_scale configure l'échelle du document en coordonnées horizontales et verticales. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.79 pdf_set_border_color() : Configure la couleur des bords autour des liens et annotations

bool **pdf_set_border_color** ( resource pdfdoc , float red , float green , float blue )

pdf_set_border_color configure la couleur des bords autour des liens et annotations; Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.80 pdf_set_border_dash() : Configure le style des lignes autour des liens at annotations

bool **pdf_set_border_dash** ( resource pdfdoc , float black , float white )

pdf_set_border_dash configure le style des lignes autour des liens. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_setdash .

## 6.29.81 pdf_set_border_style() : Choisit le style de bord autour des liens et annotations

bool **pdf_set_border_style** ( resource pdfdoc , string style , float width )

pdf_set_border_style configure le style des bords de tous les liens et annotations. style peut être solid ou dashed . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.82 pdf_set_char_spacing() : Obsolète : Configure l'espacement de caractères character spacing

Cette fonction est obsolète. Utilisez plutôt pdf_set_value .

## 6.29.83 pdf_set_duration() : Obsolète : Configure la durée entre deux pages

Obsolète.

Voir pdf_set_value .

## 6.29.84 pdf_set_font() : Obsolète : Sélectionne la taille et le nom d'une police

Obsolète. Utilisez plutôt pdf_findfont suivie de pdf_setfont .

Voir pdf_findfont et pdf_setfont .

## 6.29.85 pdf_set_horiz_scaling() : Obsolète : Configure l'échelle horizontale du texte

void **pdf_set_horiz_scaling** ( int pdf_object , float scale )

Cette fonction est bsolète. Utilisez plutôt pdf_set_value .

## 6.29.86 pdf_set_info_author() : Obsolète : remplit le champ d'auteur du document

bool **pdf_set_info_author** ( int pdfdoc , string author )

Obsolète. Utilisez pdf_set_info .

## 6.29.87 pdf_set_info_creator() : Obsolète : remplit le champ de créateur du document

bool **pdf_set_info_creator** ( int pdfdoc , string creator )

Obsolète. Utilisez pdf_set_info .

## 6.29.88 pdf_set_info_keywords() : Obsolète : remplit le champ de mots clés du document

bool **pdf_set_info_keywords** ( int pdfdoc , string keywords )

Obsolète. Utilisez pdf_set_info .

### 6.29.89 pdf_set_info_subject() : Obsolète : remplit le champ de sujet du document

bool **pdf_set_info_subject** ( int pdfdoc , string subject )

Obsolète. Utilisez pdf_set_info .

### 6.29.90 pdf_set_info_title() : Obsolète : remplit le champ de titre du document

bool **pdf_set_info_title** ( int pdfdoc , string title )

Obsolète. Utilisez pdf_set_info .

### 6.29.91 pdf_set_info() : Remplit un champ de l'en−tête de document PDF

bool **pdf_set_info** ( resource pdfdoc , string key , string value )

pdf_set_info remplit l'un des champs d'en−tête du document. Le paramètre key peut prendre l'une des valeurs suivantes : Subject , Title , Creator , Author , Keywords , ou une chaîne personnalisée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

### 6.29.92 pdf_set_leading() : Obsolète : Configure la distance entre deux lignes de texte

Cette fonction est obsolète, utilisez de préférence pdf_set_value .

### 6.29.93 pdf_set_parameter() : Modifie certains paramètres

bool **pdf_set_parameter** ( resource pdfdoc , string key , string value )

pdf_set_parameter modifie les paramètres PDFlib de type texte. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

### 6.29.94 pdf_set_text_matrix() : Obsolète : configure la matrice de texte

Voir pdf_set_paramter .

### 6.29.95 pdf_set_text_pos() : Modifie la position du texte

bool **pdf_set_text_pos** ( resource pdfdoc , float x , float y )

pdf_set_text_pos modifie la position du texte. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.96 pdf_set_text_rendering() : Obsolète : Détermine le rendu du texte

Obsolète.

Voir pdf_set_value .

## 6.29.97 pdf_set_text_rise() : Obsolète : configure l'élévation de texte

Obsolète.

Reportez vous à pdf_set_value .

## 6.29.98 pdf_set_value() : Modifie les paramètres numériques

bool **pdf_set_value** ( resource pdfdoc , string key , float value )

pdf_set_value modifie la valeur du paramètre de la PDFlib key par la valeur de value . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.99 pdf_set_word_spacing() : Obsolète : configure l'espace entre deux mots

Obsolète.

Reportez vous à pdf_set_value .

## 6.29.100 pdf_setcolor() : Configure la couleur de dessin et de remplissage

bool **pdf_setcolor** ( resource pdfdoc , string type , string colorspace , float c1 , *float c2* , *float c3* , *float c4* )

pdf_setcolor configure la couleur de dessin et de remplissage. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Le paramètre type peut être fill , stroke , ou both pour spécifier que la couleur est utilisée pour le remplissage, le dessin ou pour les deux. Le paramètre colorspace peut être gray , rgb , cmyk , spot ou pattern . Le paramètre c1 , c2 , c3 et c4 représente le composant de couleur pour l'espace de couleur colorspace . Excepté si précisé, le composant de couleur est un nombre à virgule flottante, de valeur comprise entre 0 et 1.

Pour les gris, seul c1 est utilisé.

Pour les couleurs rgb , les paramètres c1 , c2 et c3 spécifie le rouge, le vert, et le bleu, respectivement.

```
Exemple avec pdf_setcolor : couleurs RGB
 // Configure le blanc pour le dessin et le remplissage
pdf_setcolor($pdf, "both", "rgb", 1, 1, 1);
```

Pour les couleurs cmyk , les paramètres c1 , c2 , c3 et c4 représentent le cyan, le magenta, le jaune et le noir, repsectivement

```
Exemple avec pdf_setcolor : couleurs CMYK
 // Configure le noir pour le dessin et le remplissage
pdf_setcolor($pdf, "both", "cmyk", 0, 0, 0, 1);
```

Pour spot , c1 doit être une ressource de point de couleur, retournée par pdf_makespotcolor et c2 est une valeur de teinte, comprise entre 0 et 1 inclus.

Pour pattern , c1 doit être une ressource de pattern, retournée par pdf_begin_pattern .

## 6.29.101 pdf_setdash() : Configure le mode de pointillé

bool **pdf_setdash** ( resource pdfdoc , float b , float w )

**pdf_setdash** configure le mode de pointillé à la taille de b pour les traits noirs, et w pour les traits blancs. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.102 pdf_setflat() : Configure la position à plat (flatness)

bool **pdf_setflat** ( resource pdfdoc , float flatness )

pdf_setflat configure la position à plat, à une valeur comprise entre 0 et 100 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.103 pdf_setfont() : Configure la police courante

bool **pdf_setfont** ( resource pdfdoc , int font , float size )

pdf_setfont configure la police courante avec la police font et la taille size . font est une ressource retournée par pdf_findfont . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pdf_findfont .

## 6.29.104 pdf_setgray_fill() : Configure la couleur de remplissage à un niveau de gris

bool **pdf_setgray_fill** ( resource pdfdoc , float gray )

pdf_setgray_fill configure la couleur de remplissage à un niveau de gris, entre 0 et 1 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| PDFlib V4.0: Obsolète, utilisez pdf_setcolor . |

## 6.29.105 pdf_setgray_stroke() : Configure la couleur de dessin à un niveau de gris

bool **pdf_setgray_stroke** ( resource pdfdoc , float gray )

pdf_setgray_stroke configure la couleur de dessin à un niveau de gris, entre 0 et 1 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| PDFlib V4.0: Obsolète, utilisez pdf_setcolor . |

## 6.29.106 pdf_setgray() : Configure la couleur de dessin et de remplissage à un niveau de gris

bool **pdf_setgray** ( resource pdfdoc , float gray )

**pdf_setgray** configure la couleur de dessin et de remplissage au niveau de gris gray . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| PDFlib V4.0: Obsolète , utilisez pdf_setcolor . |

## 6.29.107 pdf_setlinecap() : Configure le paramètre de linecap

void **pdf_setlinecap** ( resource pdfdoc , int linecap )

**pdf_setlinecap** configure le paramètre linecap à une valeur entre 0 et 2 inclusivement.

## 6.29.108 pdf_setlinejoin() : Configure le paramètre de linejoin

bool **pdf_setlinejoin** ( resource pdfdoc , int value )

pdf_setlinejoin configure le paramètre de linejoin à une valeur entre 0 et 2 inclus. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.109 pdf_setlinewidth() : Configure la largeur de ligne

void **pdf_setlinewidth** ( resource pdfdoc , float width )

pdf_setlinewidth configure la largeur de ligne à width . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.110 pdf_setmatrix() : Configure la matrice de transformation

bool **pdf_setmatrix** ( resource pdfdoc , float a , float b , float c , float d , float e , float f )

pdf_setmatrix configure explicitement la matrice de transformation. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.111 pdf_setmiterlimit() : Configure la "miter limit"

bool **pdf_setmiterlimit** ( resource pdfdoc , float miter )

pdf_setmiterlimit configure la "miter limit" à une valeur supérieure ou égale à 1. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.112 pdf_setpolydash() : Configure des pointillés complexes

void **pdf_setpolydash** ( int pdf_object , float * dasharray )

pdf_setpolydash configure des modèles de pointillés complexes.

## 6.29.113 pdf_setrgbcolor_fill() : Choisit la couleur utilisée pour le remplissage

bool **pdf_setrgbcolor_fill** ( resource pdfdoc , float red_value , float green_value , float blue_value )

pdf_setrgbcolor_fill choisit la couleur utilisée pour le remplissage. Cette couleur est une couleur RGB. red_value représente le rouge, green_value représente le vert et blue_value représente le bleu. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| PDFlib V4.0: Obsolète : utilisez pdf_setcolor . |

## 6.29.114 pdf_setrgbcolor_stroke() : Choisit la couleur utilisée pour le dessin

bool **pdf_setrgbcolor_stroke** ( resource pdfdoc , float red_value , float green_value , float blue_value )

pdf_setrgbcolor_stroke choisit la couleur utilisée pour le dessin. Cette couleur est une couleur RGB. red_value représente le rouge, green_value représente le vert et blue_value représente le bleu. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| PDFlib V4.0: Obsolète : utilisez pdf_setcolor . |

## 6.29.115 pdf_setrgbcolor() : Choisit la couleur rgb de remplissage et de dessin

bool **pdf_setrgbcolor** ( resource pdfdoc , float red_value , float green_value , float blue_value )

pdf_setrgbcolor choisit la couleur utilisée pour le dessin et le remplissage. Cette couleur est une couleur RGB. red_value représente le rouge, green_value représente le vert et blue_value représente le bleu. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| PDFlib V4.0: Obsolète : utilisez pdf_setcolor . |

## 6.29.116 pdf_show_boxed() : Affiche le texte dans un cadre

int **pdf_show_boxed** ( resource pdfdoc , string text , float left , float top , float width , float height , string mode , *string feature* )

pdf_show_boxed formate le texte text avec la police et la taille courante, pour qu'il soit contenu dans le rectangle de coin supérieur gauche ( left , top ), de largeur width et de hauteur height . Le mode de formattage peut être left , right , center , justify , ou fulljustify . Si la largeur et la hauteur sont nulles, une simple ligne sera dessinée.

pdf_show_boxed retourne le nombre de caractères qui n'ont pas pu être dessiné dans le rectangle définit. pdf_show_boxed retourne 0 si tous les caractères ont pu être dessnié, ou si width et height sont nuls.

## 6.29.117 pdf_show_xy() : Affiche un text à une position donnée

bool **pdf_show_xy** ( resource pdfdoc , string text , float x , float y )

**pdf_show_xy** qffiche le texte text aux coordonnées ( x , y ). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.118 pdf_show() : Affiche le texte à la position courante

bool **pdf_show** ( resource pdfdoc , string text )

pdf_show affiche le texte text à la position courante, et avec la police et sa taille courante.

## 6.29.119 pdf_skew() : Incline le système de coordonnées

bool **pdf_skew** ( resource pdfdoc , float alpha , float beta )

pdf_skew incline le système de coordonnées de alpha degrés dans le sens des absisses, et de beta degrés dans le sens des coordonnées. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.120 pdf_stringwidth() : Retourne la largeur d'un texte avec la police courante

float **pdf_stringwidth** ( resource pdfdoc , string text , *int font* , *float size* )

pdf_stringwidth retourne la largeur du texte text en utilisant la police courante, configurée avec pdf_setfont . Si les paramètres optionnels de font et size sont spécifiés, la largeur sera calculée en utilisant cette police et cette taille. Notez que font est une ressource de police, retournée par pdf_findfont .

| Note |
| --- |
| Les paramètres font et size doivent être utilisé ensemble. |

Voir aussi pdf_setfont et pdf_findfont .

## 6.29.121 pdf_stroke() : Dessine la ligne le long du chemin

bool **pdf_stroke** ( resource pdfdoc )

pdf_stroke dessine la ligne le long du chemin courant, avec la couleur courante et la largeur de ligne courante, puis clot le chemin. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

## 6.29.122 pdf_translate() : Translate l'origine du système de coordonnées

bool **pdf_translate** ( resource pdfdoc , float tx , float ty )

pdf_translate translate l'origine du système de coordonnées d'un vecteur de ( tx , ty ).

# 6.30 Options PHP et informations

## 6.30.1 Introduction

Ces fonctions vous donnent accès à de nombreuses informations sur PHP lui−même, comme par exemple les configurations d'exécution, les extensions chargées, les versions, etc... Vous trouverez aussi des fonctions pour modifier des options. Et vous trouverez aussi la star des fonctions PHP phpinfo .

## 6.30.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.30.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.30.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| assert.active | "1" | PHP_INI_ALL |
| assert.bail | "0" | PHP_INI_ALL |
| assert.warning | "1" | PHP_INI_ALL |
| assert.callback | NULL | PHP_INI_ALL |
| assert.quiet_eval | "0" | PHP_INI_ALL |
| enable_dl | "1" | PHP_INI_SYSTEM |
| max_execution_time | "30" | PHP_INI_ALL |
| magic_quotes_gpc | "1" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| magic_quotes_runtime | "0" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*assert.active boolean*
: Active les évaluations de type assert .

*assert.bail boolean*
: Termine le script si une assertion échoue.

*assert.warning boolean*
: Emet une alerte PHP pour chaque assertion échouée.

*assert.callback string*
: Fonction définie par le programmeur, à appeler pour chaque assertion échouée.

*assert.quiet_eval boolean*
: Utilise la configuration courante de error_reporting durant les évaluations d'assertions. Si activée, aucune erreur n'est affichée (error_reporting(0) implicite) durant l'évaluation. Si désactivée, les erreurs sont affichées en fonction de la configuration de error_reporting

*enable_dl boolean*
: Cette directive est réellement utile lorsque PHP est compilé comme module PHP.

Vous pouvez activer le chargement dynamique d'extension (avec la fonction dl ) au cas par cas, pour chaque serveur virtuel.

La raison principale pour désactiver ce système est la sécurité. Avec le chargement dynamique, il est possible de passer outre les configurations de safe mode et open_basedir .

Par défaut, le chargement dynamique est autorisé, sauf avec le safe mode . En safe mode , il est toujours impossible d'utiliser la fonction dl .

*max_execution_time integer*

Fixe le temps maximal d'exécution d'un script, en secondes. Cela permet d'éviter que des scripts en boucles infinies saturent le serveur. La configuration par défaut est de 30 secondes.

Le temps d'exécution maximum n,est pas affectés par des appels systèmes tels que sleep . Reportez−vous à la fonction set_time_limit pour plus de détails.

Vous ne pouvez pas modifier la valeur de cette directive avec ini_set lorsque PHP est configuré en safe mode . Le seul moyen de le faire est de désactiver le safe mode ou de changer la valeur dans php.ini .

*magic_quotes_gpc boolean*

Fixe le mode magic_quotes pour les opérations GPC (Get/Post/Cookie). Lorsque magic_quotes est activé, tous les caractères ' (guillemets simples), " (guillemets doubles), \ (antislash) et NUL sont échappés avec un antislash. Si magic_quotes_sybase fonctionne aussi, les guillemets simples seront échappés avec un autre guillemet simple, plutôt qu'un antislash.

Voir aussi get_magic_quotes_gpc .

*magic_quotes_runtime boolean*

Si magic_quotes_runtime est activé, toutes les fonctions qui obtiennent des données auprès d'une source externe, y compris les bases de données et les fichiers texte, verront leur guillemets échappés avec un antislash. Si magic_quotes_sybase est aussi activé, les guillemets simples seront échappés avec un autre guillemet simple, plutôt qu'un antislash.

## 6.30.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.30.6 Constantes prédefinies

Les constantes listées ici sont toujours disponibles dans PHP.

| Constant | Description |
|---|---|
| CREDITS_ALL | Tous les crédits. C'est l'équivalent de : CREDITS_DOCS + CREDITS_GENERAL + CREDITS_GROUP + CREDITS_MODULES + CREDITS_FULLPAGE. Elle génère une page HTML complète et autonome. |
| CREDITS_DOCS | Les crédits de l'équipe de documentation |
| CREDITS_FULLPAGE | Généralement utilisé combiné avec d'autres options. Cette option indique qu'une page HTML complète doit être générée. |
| CREDITS_GENERAL | Crédits généraux. Design du langage, concepts, auteurs de PHP 4.0 et module SAPI. |
| CREDITS_GROUP | Une liste des développeurs principaux |
| CREDITS_MODULES | Une liste des extensions de PHP, et leurs auteurs |
| CREDITS_SAPI | Une liste des API de serveurs, et leurs auteurs. |

| Constante | Valeur | Description |
|---|---|---|

| | | |
|---|---|---|
| INFO_GENERAL | 1 | La ligne de configuration, le chemin du php.ini , la date de compilation, le système et plus encore. |
| INFO_CREDITS | 2 | Crédits de PHP 4. Voir aussi phpcredits . |
| INFO_CONFIGURATION | 4 | Valeurs locales et serveurs des directives PHP. Voyez aussi ini_get . |
| INFO_MODULES | 8 | Les modules chargés et leur configuration respective. |
| INFO_ENVIRONMENT | 16 | Les varibles d'environnement, qui sont aussi disponibles dans $_ENV . |
| INFO_VARIABLES | 32 | Toutes les variables prédéfinies : EGPCS (Environnement, GET, POST, Cookie, Server). |
| INFO_LICENSE | 64 | La licence PHP. Voir aussi la FAQ de la license. |
| INFO_ALL | -1 | Affiche toutes les valeurs citées ci-dessus. C'est la valeur par défaut. |

*ASSERT_ACTIVE (entier)*
*ASSERT_CALLBACK (entier)*
*ASSERT_BAIL (entier)*
*ASSERT_WARNING (entier)*
*ASSERT_QUIET_EVAL (entier)*


- Introduction
- Pré-requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- assert_options
- assert
- dl
- extension_loaded
- get_cfg_var
- get_current_user
- get_defined_constants
- get_extension_funcs
- get_include_path
- get_included_files
- get_loaded_extensions
- get_magic_quotes_gpc
- get_magic_quotes_runtime
- get_required_files
- getenv
- getlastmod
- getmygid
- getmyinode
- getmypid
- getmyuid
- getopt
- getrusage
- ini_alter
- ini_get_all
- ini_get
- ini_restore
- ini_set
- main
- memory_get_usage
- php_ini_scanned_files
- php_logo_guid
- php_sapi_name
- php_uname
- phpcredits
- phpinfo

- phpversion
- putenv
- restore_include_path
- set_include_path
- set_magic_quotes_runtime
- set_time_limit
- version_compare
- zend_logo_guid
- zend_version

## 6.30.8 assert() : Vérifie si une assertion est fausse

int **assert** ( mixed assertion )

assert va vérifier l'assertion assertion et prendre la mesure appropriée si le résultat est FALSE .

Si assertion est donnée sous la forme d'une chaîne, elle sera évaluée comme un code PHP par la fonction assert . Les avantages de ce type d'assertion sont d'être moins lourd si la vérification d'assertion est désactivée, et le contenu des messages lorsque l'assertion échoue.

Il est recommandé de n'utiliser les assertions que comme outil de débuggage. Vous pouvez les utiliser pour les vérifications d'usage : ces conditions doivent normalement être vraies, et indiquer une erreur de programmation si ce n'est pas le cas. Vous pouvez aussi vérifier la présence de certaines extensions ou limitations du système.

Les assertions ne doivent pas être utilisées pour faire des opérations de vérifications en production, comme par exemple des vérifications de valeur d'argument. En conditions normales, votre code doit être en état de fonctionner si la vérification d'assertion est désactivée.

Le comportement de assert peut être configuré par assert_options ou par .ini−settings.

La fonction assert_options et la directive ASSERT_CALLBACK permettent de configurer une fonction qui sera appelée lorsque l'assertion échoue.

Les fonctions de callback pour assert sont particuliérement utiles pour bâtir des suites de tests automatiques, car elles vous permettent de capturer facileent le code passé à l'assertion, ainsi que des informations sur le lieu et le moment de l'assertion. Même si ces informations peuvent être appelés par d'autres méthodes, les assertions sont plus rapides et plus faciles.

La fonction de callback doit accepter trois arguments. Le premier contient le nom du fichier qui a vu l'assertion échouer. Le second contient le numéro de ligne dans le fichier précédent. Le troisième argument contient l'expression qui a échoué (si il y en a : les valeurs litérales comme 1 ou "deux" ne seront pas passées par cet argument).

Gestion des assertions avec un gestionnaire personnalisé

```
 <?php
// Activation des assertions et mise en mode discret.
assert_options (ASSERT_ACTIVE, 1);
assert_options (ASSERT_WARNING, 0);
assert_options (ASSERT_QUIET_EVAL, 1);

// Cération d'un gestionnaire d'assertions
function mon_assert_handler ($file, $line, $code) {
    echo "<hr>Assertion Failed:
        File '$file'<br>
        Line '$line'<br>
        Code '$code'<br><hr>";
}

// Configuration de la méthode de callback
```

```
assert_options (ASSERT_CALLBACK, 'mon_assert_handler');

// Utilisation d'une assertion qui va échouer
assert ('mysql_query ("")');
?>
```

## 6.30.9 dl() : Charge une extension PHP à la volée

int **dl** ( string library )

dl charge l'extension PHP library à la volée. Le paramètre library est **seulement** le nom de fichier de l'extension, qui dépend de votre plate−forme. Par exemple l'extension sockets (si compilée comme module partagé, et non par défaut), sera appelée sockets.so sous Unix, et php_sockets.dll sous Windows.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.. Si la fonctionnalité de chargement de module n'est pas disponible (voir Note), ou a été désactivé (soit en inactivant la directive enable_dl ou en activant le safe mode dans le php.ini ) une E_ERROR sera émise et l'exécution du script sera stoppée. Si la fonction dl échoue parce que la librairie n'a pu être trouvée, dl retournera FALSE et émettra un message d'alerte.

Utilisez la fonction extension_loaded pour vérifier qu'une extension est chargée ou non. Cette fonction travaille aussi bien avec les extensions intégrées qu'avec les extensions dynamiquement chargées (via le php.ini , ou via dl ).

Exemple avec dl

```
<?php
 if (!extension_loaded('gd')) {
   if (!dl('gd.so')) {
       exit;
   }
 }
```

Le dossier où sont rangées les extensions dépend de votre plate forme :

Windows − Si il n'est pas explicitement indiqué dans le fichier php.ini , le dossier des extensions est c:\php4\extensions\ .

Unix − Si il n'est pas explicitement indiqué dans le fichier php.ini , le dossier des extensions dépend de

- Si PHP a été compilé avec l'option −−enable−debug
- Si PHP a été compilé avec le support (expérimental) de ZTS (Zend Thread Safety)
- de la constante interne ZEND_MODULE_API_NO (version interne de module d'API Zend, qui est en réalité la date à laquelle une modification importante de l'API a été faite, comme par exemple 20010901 )

En prenant ces paramètres en considération, le dossier des extensions vaut alors <php−install−directory>/lib/php/extension/<debug−or−not>−<zts−or−not>−ZEND_MODULE_API_NO , i.e. /usr/local/php/lib/php/extensions/debug−non−zts−20010901 ou /usr/local/php/lib/php/extensions/no−debug−zts−20010901 .

Note

dl **n'est pas** supporté sur les serveur web multi−threadés. Utilisez la directive extensions dans votre fichier php.ini lorsque vous vous trouvez dans un environnement de ce type. Cependant, les versions CGI et CLI **ne sont pas** !

Note

dl est sensible à la casse sur les plate−formes Unix.

Voir aussi directives de chargement d'extensions et extension_loaded .

## 6.30.10 extension_loaded() : Détermine si une extension est chargée ou non

bool **extension_loaded** ( string name )

extension_loaded retourne TRUE si l'extension name a été chargée.

Exemple avec extension_loaded

```
<?php
if (!extension_loaded('gd')) {
    if (!dl('gd.so')) {
        exit;
    }
}
?>
```

Vous pouvez connaître le nom des différentes extensions php en utilisant la fonction phpinfo ou bien si vous utilisez la version CGI ou CLI de PHP , vous pouvez utiliser l'option de ligne de commande −m pour lister toutes les extensions disponibles :
$ php −m
[PHP Modules]
xml
tokenizer
standard
sockets
session
posix
pcre
overload
mysql
mbstring
ctype

[Zend Modules]

Note

extension_loaded utilisent le nom interne de l'extension pour vérifier si une extension est disponible ou pas. La plupart des extensions ont des noms internes écrits en minuscules, mais il peut arriver que des extensions aient des noms en majuscules. Soyez donc prévenus que la comparaison effectuée par cette fonction est **sensible à la casse** !

Voir aussi get_loaded_extensions , get_extension_funcs , phpinfo et dl .

## 6.30.11 get_cfg_var() : Retourne la valeur d'une option de PHP

string **get_cfg_var** ( string varname )

get_cfg_var retourne la valeur courante de l'option PHP varname , ou bien FALSE en cas d'erreur.

get_cfg_var ne retourne pas les options qui ont été choisies lors de la compilation de PHP, ni ne lit dans le fichier de configuration d'Apache.

Pour vérifier si le système utilise le fichier de configuration , essayez de lire la valeur de cfg_file_path. Si cette valeur est disponible, alors le fichier de configuration est utilisé.

## 6.30.12 get_current_user() : Retourne le nom du possesseur du script courant.

string **get_current_user** ( void )

get_current_user retourne le nom du possesseur du script courant.

Voir aussi getmyuid , getmypid , getmyinode et getlastmod .

## 6.30.13 get_defined_constants() : Retourne la liste des constantes et leur valeur

array **get_defined_constants** ( void )

get_defined_constants retourne les noms et valeurs des constantes déjà définies. Cela inclut les constantes créées par les extensions, et celles créées avec la fonction define .

Exemple avec get_defined_constants

```
<?php
 print_r(get_defined_constants());
?>
```

affichera

```
Array
(
    [E_ERROR] => 1
    [E_WARNING] => 2
    [E_PARSE] => 4
    [E_NOTICE] => 8
    [E_CORE_ERROR] => 16
    [E_CORE_WARNING] => 32
    [E_COMPILE_ERROR] => 64
    [E_COMPILE_WARNING] => 128
    [E_USER_ERROR] => 256
    [E_USER_WARNING] => 512
    [E_USER_NOTICE] => 1024
    [E_ALL] => 2047
    [TRUE] => 1
)
```

Voir aussi get_loaded_extensions , get_defined_functions et get_defined_vars .

## 6.30.14 get_extension_funcs() : Liste les fonctions d'une extension

array **get_extension_funcs** ( string module_name )

get_extension_funcs retourne le nom des fonctions définies dans le module module_name .

Par exemple, les lignes suivantes :

Exemple avec get_extension_funcs

```
<?php
 print_r(get_extension_funcs("xml"));
 print_r(get_extension_funcs("gd"));
?>
```

vont afficher la liste des fonctions disponibles avec les modules xml et gd .

Voir aussi get_loaded_extensions .

## 6.30.15 get_include_path() : Lit la valeur de la directive de configuration include_path

string **get_include_path** ( )

get_include_path lit la valeur de la directive de configuration include_path .

Exemple avec get_include_path

```
<?php
// Fonctionne depuis PHP 4.3.0
print get_include_path();

// Fonctionne sur toutes les versions
print ini_get('include_path');
?>
```

Voir aussi ini_get , restore_include_path , set_include_path et include .

## 6.30.16 get_included_files() : Retourne un tableau avec les noms des fichiers qui sont inclus dans un script

array **get_included_files** ( void )

get_included_files retourne un tableau contenant les noms de tous les fichiers qui ont été ajoutés au script avec les fonctions require_once ou include_once .

L'exemple ci−dessous

Affichage des fichiers inclus et requis

```
<?php
 require_once("local.php");
 require_once("../inc/global.php");
 for ($i=1; $i<5; $i++)
   include "util".$i.".php";
   echo "Fichiers appelés avec required_once/included_once\n";
   print_r(get_required_files());
?>
```

va afficher :

```
 Fichiers appelés avec required_once/included_once
Array
(
  [0] => local.php
  [1] => /full/path/to/inc/global.php
  [2] => util1.php
  [3] => util2.php
  [4] => util3.php
  [5] => util4.php
)
```

Note

En PHP 4.0.1pl2, cette fonction supposait que required_once utilisait l'extension ".php" : les autres extensions ne fonctionnaient pas. Par ailleurs, dans cette version, le tableau retourné était un tableau associatif, et cette fonction n'était pas un alias de get_included_files

Voir aussi include , include_once , require , require_once , et get_required_files .

## 6.30.17 get_loaded_extensions() : Retourne la liste de tous les modules compilés et chargés

array **get_loaded_extensions** ( void )

get_loaded_extensions retourne un tableau contenant les noms de tous les modules compilés et chargés sur l'interpreteur PHP courant.

Par exemple, la ligne ci−dessous

Exemple avec get_loaded_extensions

```
<?php
print_r(get_loaded_extensions());
?>
```

affichera la liste suivante :

```
Array
(
    [0] => xml
    [1] => wddx
    [2] => standard
      [3] => session
      [4] => posix
      [5] => pgsql
      [6] => pcre
      [7] => gd
      [8] => ftp
      [9] => db
      [10] => Calendar
      [11] => bcmath
)
```

Voir aussi get_extension_funcs , extension_loaded , dl et phpinfo .

## 6.30.18 get_magic_quotes_gpc() : Retourne la configuration actuelle de l'option magic_quotes_gpc

int **get_magic_quotes_gpc** ( void )

get_magic_quotes_gpc retourne la configuration actuelle de l'option magic_quotes_gpc (0 pour l'option désactivée, 1 pour l'option activée).

Note

Si la directive magic_quotes_sybase est activée, elle remplacera complètement magic_quotes_gpc . Ce qui fait que même si get_magic_quotes retourne TRUE les guillemets doubles, les anti−slashes ou les caractères NULL ne seront pas protégés. Seul les guillemets simples le seront. Dans ce cas, ils ressembleront à **''** .

Gardez en tête que magic_quotes_gpc ne peut être configuré durant l'exécution du script.

Exemple avec get_magic_quotes_gpc

```
<?php
echo get_magic_quotes_gpc();         // 1
echo $_POST['lastname'];             // O\'reilly
echo addslashes($_POST['lastname']); // O\\\'reilly

if (!get_magic_quotes_gpc()) {
    $lastname = addslashes($_POST['lastname']);
} else {
    $lastname = $_POST['lastname'];
}

echo $lastname; // O\'reilly
$sql = "INSERT INTO lastnames (lastname) VALUES ('$lastname')";
?>
```

Voir aussi addslashes , stripslashes , get_magic_quotes_runtime et ini_get .

## 6.30.19 get_magic_quotes_runtime() : Retourne la configuration actuelle de l'option magic_quotes_runtime

int **get_magic_quotes_runtime** ( void )

get_magic_quotes_runtime retourne la configuration actuelle de l'option magic_quotes_runtime . (0 pour option désactivée, 1 pour option activée).

Voir aussi get_magic_quotes_gpc et set_magic_quotes_runtime .

## 6.30.20 get_required_files() : Alias de get_included_files

get_required_files est un alias de get_included_files .

## 6.30.21 getenv() : Retourne la valeur d'une variable d'environnement

string **getenv** ( string varname )

getenv retourne la valeur de la variable d'environnement varname , ou FALSE en cas d'erreur.

Exemple avec getenv

```
<?php
 $ip = getenv("REMOTE_ADDR");
  // retourne l'adresse IP de l'utilisateur
?>
```

Vous pouvez voir une liste complète des variables d'environnement en utilisant la fonction phpinfo . Vous pouvez trouver la signification de chacune d'entre elles en consultant le site concernant CGI specification (en anglais>, et particulièment la page concernant les variables d'environnement. .

Voir aussi putenv .

## 6.30.22 getlastmod() : Retourne la date de dernière modification de la page

int **getlastmod** ( void )

getlastmod retourne la date de dernière modification de la page. La valeur retournée est un marqueur de temps UNIX, utilisable comme paramètre avec la fonction date . getlastmod retourne FALSE en cas d'erreur.

Exemple avec getlastmod

```
<?php
// affiche 'Dernière modification: March 04 1998 20:43:59.'
  echo "Dernière modification : ".date( "F d Y H:i:s.", getlastmod() );
?>
```

Voir aussi date , getmyuid , getmygid , get_current_user , getmyinode , getmypid et filemtime .

## 6.30.23 getmygid() : Retourne le GID du propriétaire du script

int **getmygid** ( void )

getmygid retourne le GID du propriétaire du script, ou FALSE en cas d'erreur.

Voir aussi getmyuid , getmypid , get_current_user , getmyinode et getlastmod .

## 6.30.24 getmyinode() : Retourne l'inode du script

int **getmyinode** ( void )

getmyinode retourne l'inode du script, ou FALSE en cas d'erreur.

Voir aussi getmygid , getmyuid , get_current_user , getmypid et getlastmod .

| Note |
| --- |
| Cette fonction n'est pas implémentée sous Windows. |

## 6.30.25 getmypid() : Retourne le numéro de processus courant

int **getmypid** ( void )

getmypid retourne le numéro de processus actuel ou FALSE en cas d'erreur.

| Attention |
| --- |
| Les identifiants de processus ne sont pas uniques, et forment une source d'entropie faible. Nous recommandons de ne pas utiliser les pid pour assurer la sécurité d'un système. |

Voir aussi getmygid , getmyuid , get_current_user , getmyinode et getlastmod .

## 6.30.26 getmyuid() : Retourne l'UID du propriétaire du script actuel

int **getmyuid** ( void )

getmyuid retourne l'UID du propriétaire du script actuel ou FALSE en cas d'erreur.

Voir aussi getmypid , get_current_user , getmyinode et getlastmod .

## 6.30.27 getopt() : Lit des options passés dans la ligne de commande

array **getopt** ( string options )

getopt retourne un tableau associatif avec la liste des paire option/valeur, lues dans la ligne de commande, et basée sur le format d'option options ou bien FALSE en cas d'erreur.

Exemple avec getopt

```
$options = getopt("f:hp:");
// Analyse la ligne de commande (i.e. $GLOBALS['argv'])
```

Le paramètre options peut contenir les éléments suivants : caractères, ou caractères suivis d'un signe deux−points pour indiquer un élément optionnel. Par exemple, une option appelée x correspond à une option −x , et une chaîne x: correspond à une option de type −x argument . Cela importe peut si un argument a des espaces auparavant.

getopt retourne un tableau avec les options et leurs valeurs. Si une option n'a pas d'argument, sa valeur sera FALSE .

| Note |
| --- |
| getopt requiert la directive register_argc_argv activée. |

| Note |
| --- |
| Cette fonction n'est pas implémentée sous Windows. |

## 6.30.28 getrusage() : Retourne le niveau d'utilisation des ressources

array **getrusage** ( *int* *who* )

getrusage est une interface à la fonction system getrusage(2). Elle retourne un tableau associatif contenant les informations renvoyées par cet appel système. Si "who is 1", getrusage sera appelé avec le paramètre RUSAGE_CHILDREN.

Toutes les valeurs du tableau sont accessibles en utilisant leur nom dans le tableau.

Exemple getrusage

```
<?php
 $dat = getrusage();
 echo $dat["ru_nswap"];        // Taille de la mémoire swap
 echo $dat["ru_majflt"];       // Nombre de page mémoires utilisées
 echo $dat["ru_utime.tv_sec"]; // Temps utilisateur (en secondes)
 echo $dat["ru_utime.tv_usec"];// Temps utilisateur (en microsecondes)
?>
```

Consultez le manuel "man" pour plus de détails.

| Note |
| --- |
| Cette fonction n'est pas implémentée sous Windows. |

## 6.30.29 ini_alter() : Alias de ini_set

| Note |
| --- |
| ini_alter est un alias de ini_set . |

## 6.30.30 ini_get_all() : Lit toutes les valeurs de configuration

array **ini_get_all** ( *string* *extension* )

ini_get_all retourne toutes les valeurs de configuration sous la forme d'un tableau associatif. Si le paramètre optionnel extension est fourni, ini_get_all retourne uniquement les configurations concernant cette extension.

Le tableau retourné utilise les noms des directives comme index, et les valeurs associées sont global_value (configuré dans php.ini ), local_value (configuré via ini_set ou .htaccess ), et access (le niveau d'accès). Voyez le manuel sur les changement de configuration pour plus d'informations sur les niveaux d'accès.

Note

Il est possible pour une directive d'avoir plusieurs niveaux d'accès, ce qui est indiqué dans le champs de bits access .

Exemple avec ini_get_all

```
 <?php
$inis = ini_get_all();

print_r($inis);

/* L'affichage de ce script peut être :

 Array
(
    [allow_call_time_pass_reference] => Array
    (
        [global_value] => 1
        [local_value] => 1
        [access] => 6
    )
    [allow_url_fopen] => Array
    (
        [global_value] => 1
        [local_value] => 1
        [access] => 7
    )

    ...

)
*/
?>
```

Voir aussi get_loaded_extensions , phpinfo , ini_restore , ini_get et ini_set

## 6.30.31 ini_get() : Lit la valeur d'une option de configuration

string **ini_get** ( string varname )

ini_get retourne la valeur de l'option de configuration varname en cas de succès, et FALSE .

Note

Lecture de valeurs booléennes

Une directive de configuration ayant la valeur de off sera retournée sous la forme d'une chaîne vide, tandis qu'une directive de configuration ayant une valeur de on sera retournée sous la forme de la chaîne "1" .

Note

Lors de la lecture des tailles de mémoire

Plusieurs directives traitant de taille mémoire, comme upload_max_filesize sont stockées dans le fichier php.ini avec une notation courte. ini_get retourne la chaîne exacte stockée dans le fichier php.ini et **NON PAS** son équivalent entier. Appliquer des opérations arithmétiques classiques sur ces valeurs ne conduira à rien de bon.

```
 <?php
/*
Notre fichier php.ini contient les directives suivantes :

display_errors = On
register_globals = Off
post_max_size = 8M
*/

print 'display_errors = ' . ini_get('display_errors') . "\n";
print 'register_globals = ' . ini_get('register_globals') . "\n";
print 'post_max_size = ' . ini_get('post_max_size') . "\n";
print 'post_max_size+1 = ' . (ini_get('post_max_size')+1) . "\n";

 ?>
```

Ce script va afficher :

```
display_errors = 1
register_globals = 0
post_max_size = 8M
post_max_size+1 = 9
```

Voir aussi get_cfg_var , ini_get_all , ini_alter , ini_restore et ini_set .

## 6.30.32 ini_restore() : Restaure la valeur de l'option de configuration

string **ini_restore** ( string varname )

ini_restore restaure la valeur originale de l'option de configuration varname .

Voir aussi ini_get , ini_get_all et ini_set .

## 6.30.33 ini_set() : Modifie la valeur d'une option de configuration

string **ini_set** ( string varname , string newvalue )

ini_set change la valeur de l'option de configuration varname et lui donne la valeur de newvalue . ini_set retourne FALSE en cas d'échec, et la valeur précédente en cas de succès. La valeur de l'option de configuration sera modifiée durant toute l'exécution du script et pour ce script spécifiquement. Elle reprendra sa valeur par défaut dès la fin du script.

Toutes les options disponibles ne peuvent pas être toutes modifiées avec ini_set . Ci−dessous, vous trouverez une liste de toutes les options (disponibles en PHP 4.2.0), et si elles peuvent être modifiées.

| Nom | Par défaut | Modifiable |
|---|---|---|
| com.allow_dcom | "0" | PHP_INI_SYSTEM |
| com.autoregister_typelib | "0" | PHP_INI_SYSTEM |
| com.autoregister_verbose | "0" | PHP_INI_SYSTEM |
| com.autoregister_casesensitive | "1" | PHP_INI_SYSTEM |
| com.typelib_file | "" | PHP_INI_SYSTEM |
| crack.default_dictionary | NULL | PHP_INI_SYSTEM |
| exif.encode_unicode | "ISO−8859−15" | PHP_INI_ALL |
| exif.decode_unicode_motorola | "UCS−2BE" | PHP_INI_ALL |
| exif.decode_unicode_intel | "UCS−2LE" | PHP_INI_ALL |
| exif.encode_jis | "" | PHP_INI_ALL |
| exif.decode_jis_motorola | "JIS" | PHP_INI_ALL |
| exif.decode_jis_intel | "JIS" | PHP_INI_ALL |
| fbsql.allow_persistent | "1" | PHP_INI_SYSTEM |
| fbsql.generate_warnings | "0" | PHP_INI_SYSTEM |
| fbsql.autocommit | "1" | PHP_INI_SYSTEM |
| fbsql.max_persistent | "−1" | PHP_INI_SYSTEM |
| fbsql.max_links | "128" | PHP_INI_SYSTEM |
| fbsql.max_connections | "128" | PHP_INI_SYSTEM |
| fbsql.max_results | "128" | PHP_INI_SYSTEM |
| fbsql.batchSize | "1000" | PHP_INI_SYSTEM |
| fbsql.default_host | NULL | PHP_INI_SYSTEM |
| fbsql.default_user | "_SYSTEM" | PHP_INI_SYSTEM |
| fbsql.default_password | "" | PHP_INI_SYSTEM |
| fbsql.default_database | "" | PHP_INI_SYSTEM |
| fbsql.default_database_password | "" | PHP_INI_SYSTEM |
| hwapi.allow_persistent | "0" | PHP_INI_SYSTEM |
| hyperwave.allow_persistent | "0" | PHP_INI_SYSTEM |
| hyperwave.default_port | "418" | PHP_INI_ALL |

| | | |
|---|---|---|
| iconv.input_encoding | ICONV_INPUT_ENCODING | PHP_INI_ALL |
| iconv.output_encoding | ICONV_OUTPUT_ENCODING | PHP_INI_ALL |
| iconv.internal_encoding | ICONV_INTERNAL_ENCODING | PHP_INI_ALL |
| ifx.allow_persistent | "1" | PHP_INI_SYSTEM |
| ifx.max_persistent | "-1" | PHP_INI_SYSTEM |
| ifx.max_links | "-1" | PHP_INI_SYSTEM |
| ifx.default_host | NULL | PHP_INI_SYSTEM |
| ifx.default_user | NULL | PHP_INI_SYSTEM |
| ifx.default_password | NULL | PHP_INI_SYSTEM |
| ifx.blobinfile | "1" | PHP_INI_ALL |
| ifx.textasvarchar | "0" | PHP_INI_ALL |
| ifx.byteasvarchar | "0" | PHP_INI_ALL |
| ifx.charasvarchar | "0" | PHP_INI_ALL |
| ifx.nullformat | "0" | PHP_INI_ALL |
| ingres.allow_persistent | "1" | PHP_INI_SYSTEM |
| ingres.max_persistent | "-1" | PHP_INI_SYSTEM |
| ingres.max_links | "-1" | PHP_INI_SYSTEM |
| ingres.default_database | NULL | PHP_INI_ALL |
| ingres.default_user | NULL | PHP_INI_ALL |
| ingres.default_password | NULL | PHP_INI_ALL |
| ibase.allow_persistent | "1" | PHP_INI_SYSTEM |
| ibase.max_persistent | "-1" | PHP_INI_SYSTEM |
| ibase.max_links | "-1" | PHP_INI_SYSTEM |
| ibase.default_user | NULL | PHP_INI_ALL |
| ibase.default_password | NULL | PHP_INI_ALL |
| ibase.timestampformat | "%m/%d/%Y%H:%M:%S" | PHP_INI_ALL |
| ibase.dateformat | "%m/%d/%Y" | PHP_INI_ALL |
| ibase.timeformat | "%H:%M:%S" | PHP_INI_ALL |
| java.class.path | NULL | PHP_INI_ALL |
| java.home | NULL | PHP_INI_ALL |
| java.library.path | NULL | PHP_INI_ALL |
| java.library | JAVALIB | PHP_INI_ALL |
| java.library | NULL | PHP_INI_ALL |
| ldap.max_links | "-1" | PHP_INI_SYSTEM |
| mbstring.detect_order | NULL | PHP_INI_ALL |
| mbstring.http_input | NULL | PHP_INI_ALL |
| mbstring.http_output | NULL | PHP_INI_ALL |
| mbstring.internal_encoding | NULL | PHP_INI_ALL |
| mbstring.substitute_character | NULL | PHP_INI_ALL |
| mbstring.func_overload | "0" | PHP_INI_SYSTEM |
| mcrypt.algorithms_dir | NULL | PHP_INI_ALL |
| mcrypt.modes_dir | NULL | PHP_INI_ALL |
| mime_magic.magicfile | "/usr/share/misc/magic.mime" | PHP_INI_SYSTEM |
| mssql.allow_persistent | "1" | PHP_INI_SYSTEM |
| mssql.max_persistent | "-1" | PHP_INI_SYSTEM |
| mssql.max_links | "-1" | PHP_INI_SYSTEM |
| mssql.max_procs | "25" | PHP_INI_ALL |
| mssql.min_error_severity | "10" | PHP_INI_ALL |
| mssql.min_message_severity | "10" | PHP_INI_ALL |
| mssql.compatability_mode | "0" | PHP_INI_ALL |
| mssql.connect_timeout | "5" | PHP_INI_ALL |
| mssql.timeout | "60" | PHP_INI_ALL |
| mssql.textsize | "-1" | PHP_INI_ALL |
| mssql.textlimit | "-1" | PHP_INI_ALL |
| mssql.batchsize | "0" | PHP_INI_ALL |
| mssql.datetimeconvert | "1" | PHP_INI_ALL |
| mysql.allow_persistent | "1" | PHP_INI_SYSTEM |
| mysql.max_persistent | "-1" | PHP_INI_SYSTEM |
| mysql.max_links | "-1" | PHP_INI_SYSTEM |
| mysql.default_host | NULL | PHP_INI_ALL |
| mysql.default_user | NULL | PHP_INI_ALL |
| mysql.default_password | NULL | PHP_INI_ALL |
| mysql.default_port | NULL | PHP_INI_ALL |
| mysql.default_socket | NULL | PHP_INI_ALL |
| ncurses.value | "42" | PHP_INI_ALL |
| ncurses.string | "foobar" | PHP_INI_ALL |
| odbc.allow_persistent | "1" | PHP_INI_SYSTEM |
| odbc.max_persistent | "-1" | PHP_INI_SYSTEM |
| odbc.max_links | "-1" | PHP_INI_SYSTEM |
| odbc.default_db | NULL | PHP_INI_ALL |
| odbc.default_user | NULL | PHP_INI_ALL |

| | | |
|---|---|---|
| odbc.default_pw | NULL | PHP_INI_ALL |
| odbc.defaultlrl | "4096" | PHP_INI_ALL |
| odbc.defaultbinmode | "1" | PHP_INI_ALL |
| odbc.check_persistent | "1" | PHP_INI_SYSTEM |
| pfpro.defaulthost | "test.signio.com" | |
| pfpro.defaulthost | "test-payflow.verisign.com" | |
| pfpro.defaultport | "443" | PHP_INI_ALL |
| pfpro.defaulttimeout | "30" | PHP_INI_ALL |
| pfpro.proxyaddress | "" | PHP_INI_ALL |
| pfpro.proxyport | "" | PHP_INI_ALL |
| pfpro.proxylogon | "" | PHP_INI_ALL |
| pfpro.proxypassword | "" | PHP_INI_ALL |
| pgsql.allow_persistent | "1" | PHP_INI_SYSTEM |
| pgsql.max_persistent | "-1" | PHP_INI_SYSTEM |
| pgsql.max_links | "-1" | PHP_INI_SYSTEM |
| pgsql.auto_reset_persistent | "0" | PHP_INI_SYSTEM |
| pgsql.ignore_notice | "0" | PHP_INI_ALL |
| pgsql.log_notice | "0" | PHP_INI_ALL |
| session.save_path | "/tmp" | PHP_INI_ALL |
| session.name | "PHPSESSID" | PHP_INI_ALL |
| session.save_handler | "files" | PHP_INI_ALL |
| session.auto_start | "0" | PHP_INI_ALL |
| session.gc_probability | "1" | PHP_INI_ALL |
| session.gc_maxlifetime | "1440" | PHP_INI_ALL |
| session.serialize_handler | "php" | PHP_INI_ALL |
| session.cookie_lifetime | "0" | PHP_INI_ALL |
| session.cookie_path | "/" | PHP_INI_ALL |
| session.cookie_domain | "" | PHP_INI_ALL |
| session.cookie_secure | "" | PHP_INI_ALL |
| session.use_cookies | "1" | PHP_INI_ALL |
| session.use_only_cookies | "0" | PHP_INI_ALL |
| session.referer_check | "" | PHP_INI_ALL |
| session.entropy_file | "" | PHP_INI_ALL |
| session.entropy_length | "0" | PHP_INI_ALL |
| session.cache_limiter | "nocache" | PHP_INI_ALL |
| session.cache_expire | "180" | PHP_INI_ALL |
| session.use_trans_sid | "1" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| session.encode_sources | "globals,track" | PHP_INI_ALL |
| extname.global_value | "42" | PHP_INI_ALL |
| extname.global_string | "foobar" | PHP_INI_ALL |
| assert.active | "1" | PHP_INI_ALL |
| assert.bail | "0" | PHP_INI_ALL |
| assert.warning | "1" | PHP_INI_ALL |
| assert.callback | NULL | PHP_INI_ALL |
| assert.quiet_eval | "0" | PHP_INI_ALL |
| safe_mode_protected_env_vars | SAFE_MODE_PROTECTED_ENV_VARS | PHP_INI_SYSTEM |
| safe_mode_allowed_env_vars | SAFE_MODE_ALLOWED_ENV_VARS | PHP_INI_SYSTEM |
| url_rewriter.tags | "a=href,area=href,frame=src,form=fakeentry" | PHP_INI_ALL |
| sybct.allow_persistent | "1" | PHP_INI_SYSTEM |
| sybct.max_persistent | "-1" | PHP_INI_SYSTEM |
| sybct.max_links | "-1" | PHP_INI_SYSTEM |
| sybct.min_server_severity | "10" | PHP_INI_ALL |
| sybct.min_client_severity | "10" | PHP_INI_ALL |
| sybct.hostname | NULL | PHP_INI_ALL |
| tokenizer.global_value | "42" | PHP_INI_ALL |
| tokenizer.global_string | "foobar" | PHP_INI_ALL |
| vpopmail.directory | "" | PHP_INI_ALL |
| zlib.output_compression | "0" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| zlib.output_compression_level | "-1" | PHP_INI_ALL |
| define_syslog_variables | "0" | PHP_INI_ALL |
| highlight.bg | HL_BG_COLOR | PHP_INI_ALL |
| highlight.comment | HL_COMMENT_COLOR | PHP_INI_ALL |
| highlight.default | HL_DEFAULT_COLOR | PHP_INI_ALL |
| highlight.html | HL_HTML_COLOR | PHP_INI_ALL |
| highlight.keyword | HL_KEYWORD_COLOR | PHP_INI_ALL |
| highlight.string | HL_STRING_COLOR | PHP_INI_ALL |
| allow_call_time_pass_reference | "1" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| asp_tags | "0" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| display_errors | "1" | PHP_INI_ALL |
| display_startup_errors | "0" | PHP_INI_ALL |
| enable_dl | "1" | PHP_INI_SYSTEM |

| | | |
|---|---|---|
| expose_php | "1" | PHP_INI_SYSTEM |
| html_errors | "1" | PHP_INI_SYSTEM |
| xmlrpc_errors | "0" | PHP_INI_SYSTEM |
| xmlrpc_error_number | "0" | PHP_INI_ALL |
| ignore_user_abort | "0" | PHP_INI_ALL |
| implicit_flush | "0" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| log_errors | "0" | PHP_INI_ALL |
| log_errors_max_len | "1024" | PHP_INI_ALL |
| ignore_repeated_errors | "0" | PHP_INI_ALL |
| ignore_repeated_source | "0" | PHP_INI_ALL |
| magic_quotes_gpc | "1" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| magic_quotes_runtime | "0" | PHP_INI_ALL |
| magic_quotes_sybase | "0" | PHP_INI_ALL |
| output_buffering | "0" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| output_handler | NULL | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| register_argc_argv | "1" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| register_globals | "0" | PHP_INI_PERDIR\|PHP_INI_SYSTEM |
| safe_mode | "1" | PHP_INI_SYSTEM |
| safe_mode | "0" | PHP_INI_SYSTEM |
| safe_mode_include_dir | NULL | PHP_INI_SYSTEM |
| safe_mode_gid | "0" | PHP_INI_SYSTEM |
| short_open_tag | DEFAULT_SHORT_OPEN_TAG | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| sql.safe_mode | "0" | PHP_INI_SYSTEM |
| track_errors | "0" | PHP_INI_ALL |
| y2k_compliance | "0" | PHP_INI_ALL |
| unserialize_callback_func | NULL | PHP_INI_ALL |
| arg_separator.output | "&" | PHP_INI_ALL |
| arg_separator.input | "&" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| auto_append_file | NULL | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| auto_prepend_file | NULL | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| doc_root | NULL | PHP_INI_SYSTEM |
| default_charset | SAPI_DEFAULT_CHARSET | PHP_INI_ALL |
| default_mimetype | SAPI_DEFAULT_MIMETYPE | PHP_INI_ALL |
| error_log | NULL | PHP_INI_ALL |
| extension_dir | PHP_EXTENSION_DIR | PHP_INI_SYSTEM |
| gpc_order | "GPC" | PHP_INI_ALL |
| include_path | PHP_INCLUDE_PATH | PHP_INI_ALL |
| max_execution_time | "30" | PHP_INI_ALL |
| open_basedir | NULL | PHP_INI_SYSTEM |
| safe_mode_exec_dir | "1" | PHP_INI_SYSTEM |
| upload_max_filesize | "2M" | PHP_INI_SYSTEM |
| file_uploads | "1" | PHP_INI_SYSTEM |
| post_max_size | "8M" | PHP_INI_SYSTEM |
| upload_tmp_dir | NULL | PHP_INI_SYSTEM |
| user_dir | NULL | PHP_INI_SYSTEM |
| variables_order | NULL | PHP_INI_ALL |
| error_append_string | NULL | PHP_INI_ALL |
| error_prepend_string | NULL | PHP_INI_ALL |
| SMTP | "localhost" | PHP_INI_ALL |
| smtp_port | 25 | PHP_INI_ALL |
| browscap | NULL | PHP_INI_SYSTEM |
| error_reporting | NULL | PHP_INI_ALL |
| memory_limit | "8M" | PHP_INI_ALL |
| precision | "14" | PHP_INI_ALL |
| sendmail_from | NULL | PHP_INI_ALL |
| sendmail_path | DEFAULT_SENDMAIL_PATH | PHP_INI_SYSTEM |
| disable_functions | "" | PHP_INI_SYSTEM |
| allow_url_fopen | "1" | PHP_INI_ALL |
| always_populate_raw_post_data | "0" | PHP_INI_ALL |
| xbithack | "0" | PHP_INI_ALL |
| engine | "1" | PHP_INI_ALL |
| last_modified | "0" | PHP_INI_ALL |
| child_terminate | "0" | PHP_INI_ALL |
| async_send | "0" | PHP_INI_ALL |

| Constante | Valeur | Signification |
|---|---|---|
| PHP_INI_USER | 1 | La valeur peut être modifiée dans un script |
| PHP_INI_PERDIR | 2 | La valeur peut être modifiée dans le fichier .htaccess et les directives d'un vhost |
| PHP_INI_SYSTEM | 4 | La valeur peut être modifiée dans php.ini ou httpd.conf (mais pas dans les blocs vhost de httpd.conf |
| PHP_INI_ALL | 7 | La valeur peut être modifiée n'importe où |

## 6.30.32 ini_restore() : Restaure la valeur de l'option de configuration

Voir aussi get_cfg_var , ini_get_all , ini_get et ini_restore

## 6.30.34 main() : Fausse documentation pour main

Il n'y a pas de fonction appelée main , hormis dans les sources PHP . En PHP 4.3.0, un nouveau type de gestion d'erreur a été introduit dans les sources de PHP : php_error_docref. Une des fonctionnalité était de fournir un lien vers la page de manuel de PHP lorsque la directive html_errors était activée (par défaut).

Certains messages d'erreur faisaient référence à une page de manuel pour la fonction main alors que cette page n'existe pas.

| Nom de la fonction | Ne pointe plus sur cette page depuis la version |
|---|---|
| include | 4.3.1 |
| include_once | 4.3.1 |
| require | 4.3.1 |
| require_once | 4.3.1 |

Voir aussi html_errors et display_errors .

## 6.30.35 memory_get_usage() : Indique la quantité de mémoire utilisée par PHP

int **memory_get_usage** ( )

memory_get_usage retourne la quantité de mémoire allouée à PHP à cet instant.

memory_get_usage n'est disponible que si PHP est compilé avec l'option −−enable−memory−limit .

Exemple avec memory_get_usage

```
<?php
// Ceci n'est qu'un exemple. Les chiffres ci-dessous
// différeront suivant les systèmes et les configurations

echo memory_get_usage() . "\n"; // 36640

$a = str_repeat("Salut", 4242);

echo memory_get_usage() . "\n"; // 57960

unset($a);

echo memory_get_usage() . "\n"; // 36744

?>
```

Voir aussi memory−limit .

## 6.30.36 php_ini_scanned_files() : Retourne la liste des fichiers .ini analysés dans les dossiers de configuration supplémentaires

string **php_ini_scanned_files** ( void )

php_ini_scanned_files retourne une liste de nom de fichiers de configuration analysés, en plus de php.ini . Cette liste est au format CSV. Ces fichiers sont situés dans un dossier défini par l'option −−with−config−file−scan−dir , définie au moment de la compilation.

php_ini_scanned_files retourne une chaîne de caractères, et les noms de fichiers sont séparés par des virgules. Si −−with−config−files−scan−dir n'était pas configuré, FALSE est retourné. Si cette option était configurée, mais que le dossier était vide, une chaîne vide est retournée. Si un fichier n'est pas lisible, le nom du fichier sera inclus dans la liste, mais une erreur sera générée. Cette erreur sera visible au moment de la compilation, et lorsque vous appelerez la fonction php_ini_scanned_files .

Les fichiers retournés incluent le chemin, comme déclaré dans la directive −−with−config−file−scan−dir . De plus, chaque virgule est suivi d'une nouvelle ligne.

Un exemple simple de liste retournée par php_ini_scanned_files

```
<?php

if ($filelist = php_ini_scanned_files()) {

    if (strlen($filelist) > 0) {

        $files = explode(',', $filelist);

        foreach ($files as $file) {
            echo "<li>" . trim($file) . "</li>\n";
        }
    }
}
?>
```

Voir aussi ini_set et phpinfo .

## 6.30.37 php_logo_guid() : Retourne l'identifiant du logo PHP

string **php_logo_guid** ( void )

Cette fonction retourne l'identifiant qui peut être utilisé pour afficher le logo PHP en utilisant une image incorporée dans les sources.

Note

Cette fonctionnalité a été ajoutée dans PHP 4 Beta 4.

Exemple avec php_logo_guid

```
<?php

 echo '<img src="' . $_SERVER['PHP_SELF'] .
 '?=' . php_logo_guid() . '" alt="Logo PHP !" />';

?>
```

Voir aussi phpinfo , phpversion , phpcredits et zend_logo_guid .

## 6.30.38 php_sapi_name() : Retourne le type d'interface utilisé entre le serveur web et PHP

string **php_sapi_name** ( void )

php_sapi_name retourne une chaîne en minuscule qui décrit le type d'interface utilisé en le serveur web et PHP (Server API, SAPI). En CGI PHP, cette chaîne est "CGI", en mod_php pour Apache, cette chaîne est "apache", etc...

Exemple avec php_sapi_name

```
<?php
$inter_type = php_sapi_name();
if ($inter_type == "cgi")
    print "Vous utilisez CGI PHP\n";
```

```
else
    print "Vous n'utilisez pas CGI PHP\n";
?>
```

## 6.30.39 php_uname() : Retourne les informations sur le système d'exploitation

string **php_uname** ( void )

php_uname retourne une chaîne de caractères avec des informations sur le système d'exploitation sur lequel tourne PHP. Si vous voulez juste savoir le nom du système d'exploitation, utilisez plutôt la constante PHP_OS .

Exemples avec php_uname

```
<?php
echo php_uname();
echo PHP_OS;

/* Affichages possibles :
Linux localhost 2.4.21-0.13mdk #1 Fri Mar 14 15:08:06 EST 2003 i686
Linux

FreeBSD localhost 3.2-RELEASE #15: Mon Dec 17 08:46:02 GMT 2001
FreeBSD

Windows NT XN1 5.1 build 2600
WINNT
*/

if (strtoupper(substr(PHP_OS, 0, 3)) === 'WIN') {
    echo 'Le serveur tourne sous Windows !';
} else {
    echo 'Le serveur ne tourne pas sous Windows !';
}

?>
```

Il existe aussi des constantes PHP prédéfinies liées, qui peuvent s'avérer utiles, par exemple :

Exemples avec quelques constantes liées au système

```
<?php
// *nix
echo DIRECTORY_SEPARATOR; // :
echo PHP_SHLIB_SUFFIX;    // so

// Win*
echo DIRECTORY_SEPARATOR; // ;
echo PHP_SHLIB_SUFFIX;    // dll
?>
```

Voir aussi phpversion , php_sapi_name et phpinfo .

## 6.30.40 phpcredits() : Imprime les crédits de PHP

void **phpcredits** ( *int flag* )

phpcredits affiche la liste des développeurs PHP , des modules, etc... Elle génère le code HTML approprié pour insérer les informations dans une page. Le paramètre flag indique les informations qui doivent être affichées. Par exemple, pour afficher les crédits généraux, vous pouvez utiliser le code suivant :

Afficher tous les crédits de PHP

```
<?php
 phpcredits(CREDITS_GENERAL);
?>
```

Et pour afficher la liste des développeurs et du groupe de documentation dans une page séparée, vous utiliserez

Afficher les développeurs et le groupe de documentation PHP

```
<?php
 phpcredits(CREDITS_GROUP + CREDITS_DOCS + CREDITS_FULLPAGE);
?>
```

Si vous vous sentez l'envie de placer tous les crédits dans votre page, vous pouvez utiliser ceci :

Personnalisation de la page de crédits PHP

```
<html>
<head>
 <title>Ma page de crédits</title>
</head>
<body>
<?php
 // Un peu de votre code
 phpcredits(CREDITS_ALL);
 // Un autre peu de votre code
?>
</body>
</html>
```

| Nom | Description |
|---|---|
| CREDITS_ALL | Tous les crédits, équivalent à : CREDITS_DOCS + CREDITS_GENERAL + CREDITS_GROUP + CREDITS_MODULES + CREDITS_FULLPAGE . La fonction génère alors une page HTML complète. |
| CREDITS_DOCS | Les crédits du groupe de documentation |
| CREDITS_FULLPAGE | En général, ce paramètre est utilisé avec d'autres constantes. Il indique que la page ainsi générée doit être une page HTML complète, avec toutes les balises nécessaires. |
| CREDITS_GENERAL | Crédits Généraux : conception et design du langage, auteurs de PHP 4.0, module SAPI. |
| CREDITS_GROUP | Une liste des développeurs principaux |
| CREDITS_MODULES | Une liste des extensions de PHP , et leurs auteurs |
| CREDITS_SAPI | Cette constante est définie, mais elle n'est toujours pas utilisée sous PHP 4.0.1pl2. |

Voir aussi phpinfo , phpversion et php_logo_guid .

## 6.30.41 phpinfo() : Affiche de nombreuses informations sur le PHP

int **phpinfo** ( *int what* )

phpinfo affiche de nombreuses informations sur le PHP, concernant sa configuration courante : options de compilation, extensions, version, informations sur le serveur, et environnement (lorsque compilé comme module), environnement PHP, chemins, utilisateur, en−têtes HTTP, et licence GNU Public License.

Comme tous les systèmes sont configurés différemment, phpinfo sert généralement à vérifier la configuration ainsi que les variables prédéfinies , pour une plate−forme donnée. De plus, phpinfo est un outil de débogage, car il affiche le contenu de toutes les variables EGPCS (Environnement, GET, POST, Cookie, Serveur).

L'affichage peut être personnalisé en utilisant une ou plusieurs des constantes suivantes. Elles sont combinables avec l'opérateur OR , et doivent être passées dans le paramètre what . Vous pouvez aussi les additionner.

| Nom (constant) | Valeur | Description |
|---|---|---|
| INFO_GENERAL | 1 | La ligne de configuration, le chemin du php.ini , la date de compilation, le serveur web, le système, etc... |
| INFO_CREDITS | 2 | Les crédits de PHP 4. Voir aussi phpcredits . |
| INFO_CONFIGURATION | 4 | Valeurs courantes locales et générales des directives PHP. Voyez aussi la fonction ini_get . |
| INFO_MODULES | 8 | Modules chargées et leur configuration spécifique. |
| INFO_ENVIRONMENT | 16 | Informations sur les environnements de variables, qui sont disponibles dans la variable $_ENV . |
| INFO_VARIABLES | 32 | Affiche toutes les variables prédéfinies, issues de l'environnement, la méthode GET, la méthode POST, les cookies et le serveur. |
| INFO_LICENSE | 64 | La licence PHP. Voir aussi la FAQ de a licence. |
| INFO_ALL | −1 | Affiche toutes les informations sus−citées. C'est la valeur par défaut. |

Exemple avec phpinfo

```
<?php

// Affiche toutes les information, comme le ferait INFO_ALL
phpinfo();

// Affiche uniquement le module d'information.
// phpinfo(8) fournirait les mêmes informations.
phpinfo(INFO_MODULES);

?>
```

Note

Une partie des informations affichées sont désactivées si la directive expose_php est configurée avec la valeur off . Cela inclus les logos PHP et Zend, ainsi que les crédits.

Voir aussi get_loaded_modules , phpversion , phpcredits , php_logo_guid , ini_get , ini_set et la section sur les variables prédéfinies .

## 6.30.42 phpversion() : Retourne le numéro de la version courante de PHP

string **phpversion** ( void )

phpversion retourne le numéro de la version courante de PHP.

Note

Cette information est aussi disponible via la constante prédéfinie PHP_VERSION .

Exemple avec phpversion

```
<?php
// affiche le numéro de version courante du PHP.
  echo "PHP Version : ".phpversion();
?>
```

Voir aussi version_compare , phpinfo , phpcredits , php_logo_guid et zend_version .

## 6.30.43 putenv() : Fixe la valeur d'une variable d'environnement

void **putenv** ( string setting )

putenv fixe la valeur d'une variable d'environnement. Cette valeur n'existera que durant la vie du script courant, et l'environnement initial sera restauré lorsque le script sera terminé.

Modifier la valeur de certaines variables système peut être un trou de sécurité considérable. La directive de configuration safe_mode_allowed_env_vars contient une liste de préfixes, séparés par des virgules. Lorsque le Safe Mode est actif, l'utilisateur ne peut que modifier les variables qui dont le nom commence par les préfixes fournis par cette directive. Par défaut, les utilisateurs ne peuvent modifier que les variables qui commencent par PHP_ (i.e. PHP_FOO=BAR ). Note: si cette directive est vide, PHP autorisera la modification de TOUTES les variables d'environnement!.

La directive de configuration safe_mode_protected_env_vars contient une liste de variables d'environnement, séparées par des virgules. Les utilisateurs ne pourront pas modifier ces variables avec la fonction putenv . Ces variables seront protégées même si safe_mode_allowed_env_vars permet leur modification.

Modification d'une variable d'environnement

```
<?php
  putenv("UNIQID=$uniqid");
?>
```

Voir aussi getenv .

## 6.30.44 restore_include_path() : Restaure la valeur de la directive de configuration include_path

void **restore_include_path** ( )

**restore_include_path** restaure la valeur de la directive de configuration include_path à sa valeur originale de début de script, telle qu'inscrite dans le php.ini .

Exemple restore_include_path

```
<?php

print get_include_path();  // .:/usr/local/lib/php

set_include_path('/inc');

print get_include_path();  // /inc


// Fonctionne depuis PHP 4.3.0
restore_include_path();

// Fonctionne sur toutes les versions
ini_restore('include_path');

print get_include_path();  // .:/usr/local/lib/php

?>
```

Voir aussi ini_restore , set_include_path , get_include_path et include .

## 6.30.45 set_include_path() : Modifie la valeur de la directive de configuration include_path

string **set_include_path** ( string new_include_path )

set_include_path modifie la valeur de la directive de configuration include_path , pour la durée du script en cours. set_include_path retourne l'ancienne valeur de include_path en cas de succès, et FALSE sinon.

Exemple avec set_include_path

```
<?php
// Fonctionne depuis PHP 4.3.0
set_include_path('/inc');

// Fonctionne sur toutes les versions
ini_set('include_path', '/inc');
?>
```

Voir aussi ini_set , get_include_path , restore_include_path et include .

## 6.30.46 set_magic_quotes_runtime() : Active/désactive l'option magic_quotes_runtime.

bool **set_magic_quotes_runtime** ( int new_setting )

set_magic_quotes_runtime active/désactive l'option magic_quotes_runtime . (0 l'option est désactivée, 1 l'option est activée).

Voir aussi get_magic_quotes_gpc et get_magic_quotes_runtime .

## 6.30.47 set_time_limit() : Fixe le temps maximum d'exécution d'un script

void **set_time_limit** ( int seconds )

set_time_limit fixe le délai d'expiration d'un script, en secondes. Si cette limite est atteinte, le script s'interrompt, et renvoie une erreur fatale. La valeur par défaut est 30 secondes ou, si c'est le cas, la valeur de la directive max_execution_time définie dans le php.ini . Si la valeur est zéro, il n'y a alors aucune limite imposée.

Lorsqu'elle est appelée, la fonction set_time_limit remet le compteur de zéro. En d'autres termes, si la limite par défaut est à 30 secondes, et qu'après 25 secondes d'exécution du script l'appel set_time_limit(20) est fait, alors le script tournera pendant un total de 45 secondes avant de finir.

Notez que set_time_limit n'a pas d'effet lorsque PHP fonctionne en mode safe mode . Il n'y a pas d'autre solution que de changer de mode, ou de modifier la durée maximale d'exécution dans le php.ini .

## 6.30.48 version_compare() : Compare deux versions de PHP

int **version_compare** ( string version1 , string version2 , *string operator* )

version_compare compare les deux versions de PHP standardisée. Cette fonction est pratique pour les programmes qui doivent vérifier la version de PHP qui les fait tourner.

version_compare retourne −1 si version1 est inférieure à version2 , 0 si elles sont égales, et 1 dans le reste des cas.

Si vous spécifiez le troisième argument optionnel operator , vous pouvez tester une relation particulière. Les opérateurs possibles sont < , lt , <= , le , > , gt , >= , ge , == , = , eq , != , <> , ne . En utilisant cet argument, version_compare retournera 1 si la relation est vérifiée et 0 sinon.

Exemple avec version_compare

```
<?php
 echo version_compare("4.0.4", "4.0.6");
// affiche -1
 echo version_compare("4.0.4", "4.0.6", "<");
 echo version_compare("4.0.6", "4.0.6", "eq");
// affichent tous 1
?>
```

## 6.30.49 zend_logo_guid() : Retourne le logo de Zend

string **zend_logo_guid** ( void )

Note

Cette fonctionnalité a été ajoutée en PHP 4 Beta 4.

## 6.30.50 zend_version() : Lit la version courante du moteur Zend

string **zend_version** ( void )

zend_version retourne une chaîne contenant le numéro de version du moteur d'analyse Zend, pour l'exécutable PHP courant.

Exemple avec zend_version

```
 <?php
// affiche e.g. 'Version du moteur Zend: 1.0.4'
// ou bien quelque chose d'approchant si votre version de PHP date un peu
  echo "Version du moteur Zend: ".zend_version();
?>
```

Voir aussi phpinfo , phpcredits , php_logo_guid et phpversion .

# 6.31 PostgreSQL

## 6.31.1 Introduction

La base de données PostgreSQL est un produit Open Source, disponible sans frais. Postgres, développé au département de Science informatique, à UC Berkeley, mis en place la majorité des concepts des bases relationnelles, actuellement disponibles sur le marché. PostgreSQL accepte le langage SQL92/SQL3, assure l'intégrité transactionnelle, et l'extension de type. PostgreSQL est une évolution du code originale de Berkeley.

## 6.31.2 Pré−requis

Pour accéder au support PostgreSQL, vous avez besoin de PostgreSQL 6.5 ou plus récent, PostgreSQL 7.0 ou plus récent pour activer toutes les fonctionnalités du moduel PostgreSQL. PostgreSQL supporte de nombreux jeux de caractères, y compris les jeux multi−octets asiatiques. La version courante et plus de détails sur PostgreSQL sont accessibles sur le site http://www.postgresql.org/ et http://techdocs.postgresql.org/ .

## 6.31.3 Installation

Afin d'activer le support PostgreSQL, l'option −−with−pgsql[=DIR] est nécessaire lore de la compilation de PHP. DIR est le path du dossier d'installation de PostgreSQL, et par défaut il vaut /usr/local/pgsql . Si le module shared object est disponible, le module PostgreSQL est chargeable avec la directive du fichier php.ini ou via la fonction dl .

## 6.31.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| pgsql.allow_persistent | "1" | PHP_INI_SYSTEM |
| pgsql.max_persistent | "−1" | PHP_INI_SYSTEM |
| pgsql.max_links | "−1" | PHP_INI_SYSTEM |
| pgsql.auto_reset_persistent | "0" | PHP_INI_SYSTEM |
| pgsql.ignore_notice | "0" | PHP_INI_ALL |
| pgsql.log_notice | "0" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*pgsql.allow_persistent bool*
: Whether to allow persistent Postgres connections.

*pgsql.max_persistent integer*
: Le nombre maximum de connexions persistantes Postgre par processus.

*pgsql.max_links integer*
: Le nombre maximum de connexions Postgre par processus, y compris les connexions persistantes.

## 6.31.5 Trucs et astuces d'utilisation

| Attention |
| --- |
| Utiliser le module PostgreSQL avec PHP 4.0.6 n'est pas recommandé à cause d'un bug dans le gestionnaire d'alerte. Utilisez plutôt la version 4.1.0 ou plus récent. |

Attention

Le nom des fonctions PostgreSQL va changer dans la version 4.2.0 pour prendre en compte les standards actuels de programmation. La plupart des nouveaux noms recevront des soulignés, comme par exemple pg_lo_open(). Certaines fonctions sont renommées différemment, comme par exemple pg_exec() en pg_query(). Les anciens noms sont toujours utilisés pour encore quelques versions, mais ils seront bientôt supprimés définitivement.

| Ancien nom | Nouveau nom |
| --- | --- |
| pg_cmdtuples | pg_affected_rows |
| pg_errormessage | pg_last_error |
| pg_exec | pg_query |
| pg_fieldname | pg_field_name |
| pg_fieldsize | pg_field_size |
| pg_fieldnum | pg_field_num |
| pg_fieldprtlen | pg_field_prtlen |
| pg_fieldisnull | pg_field_is_null |
| pg_freeresult | pg_free_result |
| pg_getlastoid | pg_last_oid |
| pg_loreadall | pg_lo_read_all |
| pg_locreate | pg_lo_create |
| pg_lounlink | pg_lo_unlink |
| pg_loopen | pg_lo_open |
| pg_loclose | pg_lo_close |
| pg_loread | pg_lo_read |
| pg_lowrite | pg_lo_write |
| pg_loimport | pg_lo_import |
| pg_loexport | pg_lo_export |
| pg_numrows | pg_num_rows |
| pg_numfields | pg_num_fields |
| pg_result | pg_fetch_result |

Les anciennes syntaxes de pg_connect / pg_pconnect seront rendues obsolètes pour supporter les nouvelles connexions asynchrones. Utilisez la chaîne de connexion avec pg_connect et pg_pconnect .

Toutes les fonctions ne sont pas supportées par toutes les compilations. Cela dépend de votre librairie libpq (la librairie C client de PostgreSQL), et comment libq est compilé. Si il y a des fonctions qui manquent, libpq ne supporte par la fonctionnalité sur laquelle reposait la fonction n'est pas disponible.

Il est aussi important que vous utilisiez une version de libpq qui soit plus récente que le serveur sur lequel vous vous connectez. Si vous utilisez une version de libq plus ancienne que le serveur, vous aurez des problèmes.

Depuis la version 6.3 (03/02/1998) PostgreSQL utilise les sockets UNIX, et une table est dédiée à ces nouvelles capacités. La socket est située dans le dossier /tmp/.s.PGSQL.5432 . Cette option peut être activée avec '−i' passé au postmaster et cela s'interprète: "écoute sur les sockets TCP/IP et sur les sockets Unix".

| Postmaster | PHP | Statut |
| --- | --- | --- |
| postmaster & | pg_connect("dbname=MonDbName"); | OK |
| postmaster −i & | pg_connect("dbname=MonDbName"); | OK |
| postmaster & | pg_connect("host=localhost dbname=MonDbName"); | Unable to connect to PostgreSQL server: connectDB() failed: Impossible de se connecter au serveur PostgreSQL: connectDB() a échoué. Est ce que le postmaster |

| | | |
|---|---|---|
| | | fonctionne, et accepte les TCP/IP (option −i) sur le port '5432'? |
| postmaster −i & | pg_connect("host=localhost dbname=MonDbName"); | OK |

Il est possible de se connecter avec la commande suivante : $conn = pg_Connect("host=monHote port=monPort tty=monTTY options=myOptions dbname=myDB user=myUser password=myPassword");

L'ancienne syntaxe : $conn = pg_connect("host", "port", "options", "tty", "dbname") est obsolète.

Les variables environnementales affecent le comportement de PostgreSQL. Par exemple, le module PostgreSQL va rechercher PGHOST dans les variables d'environnement, si le nom du serveur hôte est omis dans la chaîne de connexion. Les variables d'environnement supportées sont différentes suivant les versions. Reportez vous au manuel du programmeur PostgreSQL (libpq − Environment Variables) pour plus de détails.

Assurez vous que vous avez bien configuré vos variables d'environnement pour le bon utilisateur. Utilisez $_ENV ou getenv pour vérifier quelles variables d'environnement sont disponibles pour le processus courant.

```
Configuration par défaut des paramètres
 PGHOST=pgsql.example.com
PGPORT=7890
PGDATABASE=web−system
PGUSER=web−user
PGPASSWORD=secret
PGDATESTYLE=ISO
PGTZ=JST
PGCLIENTENCODING=EUC−JP

export PGHOST PGPORT PGDATABASE PGUSER PGPASSWORD PGDATESTYLE PGTZ PGCLIENTENCODING
```

## 6.31.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*PGSQL_ASSOC (entier)*
*PGSQL_NUM (entier)*
*PGSQL_BOTH (entier)*
*PGSQL_CONNECTION_BAD (entier)*
*PGSQL_CONNECTION_OK (entier)*
*PGSQL_SEEK_SET (entier)*
*PGSQL_SEEK_CUR (entier)*
*PGSQL_SEEK_END (entier)*
*PGSQL_ESCAPE_STRING (entier)*
*PGSQL_ESCAPE_BYTEA (entier)*
*PGSQL_EMPTY_QUERY (entier)*
*PGSQL_COMMAND_OK (entier)*
*PGSQL_TUPLES_OK (entier)*
*PGSQL_COPY_OUT (entier)*
*PGSQL_COPY_IN (entier)*
*PGSQL_BAD_RESPONSE (entier)*
*PGSQL_NONFATAL_ERROR (entier)*
*PGSQL_FATAL_ERROR (entier)*

## 6.31.7 Exemples

Depuis PostgreSQL 7.1.0, vous pouvez stocker jusqu'à 1Go dans un champ de type text. Dans les anciennes versions, vous étiez limité à la taille maximale d'un bloc (qui, par défaut, valait 8 ko, et au mieux, 32 ko, suivant le choix au moment de la compilation).

Pour utiliser l'interface des grands objets (large object (lo) interface), il est nécessaire de les placer dans un bloc de transaction. Un bloc de transaction commence avec begin et si la transaction se termine avec un commit et end . Si la transaction échoue, elle doit être conclue par un abort et rollback .

Utilisation des objets de grande taille (Large Objects)

```
<?php
   $database = pg_connect ("dbname=jacarta");
   pg_query ($database, "begin");
   $oid = pg_lo_create ($database);
   echo "$oid\n";
   $handle = pg_lo_open ($database, $oid, "w");
   echo "$handle\n";
   pg_lo_write ($handle, "large object data");
   pg_lo_close ($handle);
   pg_query ($database, "commit");
?>
```

Vous ne devez pas fermer la connexion au serveur PostgreSQL avant de fermer l'objet de grande taille.


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Trucs et astuces d'utilisation
- Constantes prédefinies
- Exemples
- pg_affected_rows
- pg_cancel_query
- pg_client_encoding
- pg_close
- pg_connect
- pg_connection_busy
- pg_connection_reset
- pg_connection_status
- pg_convert
- pg_copy_from
- pg_copy_to
- pg_dbname
- pg_delete
- pg_end_copy
- pg_escape_bytea
- pg_escape_string
- pg_fetch_all
- pg_fetch_array
- pg_fetch_assoc
- pg_fetch_object
- pg_fetch_result
- pg_fetch_row
- pg_field_is_null
- pg_field_name
- pg_field_num

- pg_field_prtlen
- pg_field_size
- pg_field_type
- pg_free_result
- pg_get_notify
- pg_get_pid
- pg_get_result
- pg_host
- pg_insert
- pg_last_error
- pg_last_notice
- pg_last_oid
- pg_lo_close
- pg_lo_create
- pg_lo_export
- pg_lo_import
- pg_lo_open
- pg_lo_read_all
- pg_lo_read
- pg_lo_seek
- pg_lo_tell
- pg_lo_unlink
- pg_lo_write
- pg_meta_data
- pg_num_fields
- pg_num_rows
- pg_options
- pg_pconnect
- pg_ping
- pg_port
- pg_put_line
- pg_query
- pg_result_error
- pg_result_seek
- pg_result_status
- pg_select
- pg_send_query
- pg_set_client_encoding
- pg_trace
- pg_tty
- pg_unescape_bytea
- pg_untrace
- pg_update

## 6.31.9 pg_cancel_query() : Annule une requête asynchrone

bool **pg_cancel_query** ( resource connection )

pg_cancel_query annule la requête asynchrone, démarrée avec pg_send_query , sur la connexion connection . Vous ne pouvez pas annuler une requête démarrée avec pg_query .

Voir aussi pg_send_query et pg_connection_busy .

## 6.31.10 pg_client_encoding() : Lit l'encodage du client

string **pg_client_encoding** ( *resource connection* )

pg_client_encoding retourne l'encodage du client. Elle retourne une des valeurs suivantes : SQL_ASCII, EUC_JP, EUC_CN, EUC_KR, EUC_TW, UNICODE, MULE_INTERNAL, LATINX (X=1...9), KOI8, WIN, ALT, SJIS, BIG5, WIN1250.

| Note |
| --- |
| Cette fonction requiert PHP−4.0.2 ou plus récent et PostgreSQL−7.0 ou plus récent.<br><br>Jadis, pg_client_encoding s'appelait pg_clientencoding(). |

Voir aussi pg_set_client_encoding .

## 6.31.11 pg_close() : Termine une connexion PostgreSQL

bool **pg_close** ( resource connection )

pg_close ferme la connexion au serveur PostgreSQL associé à connection . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Il n'est généralement pas nécessaire de fermer une connexion non persistante, car elles sont automatiquement fermées à la fin d'un script. |

pg_close ne ferme pas les connexions persistantes ouvertes avec pg_pconnect .

Exemple avec pg_close

```
<?php
   $dbconn = pg_connect("host=localhost port=5432 dbname=mary")
       or die("Impossible de se connecter");
   echo "Connexion à PostGreSQL";
   pg_close($dbconn);
?>
```

Si des objets de grande taille ont été ouverts avec cette connexion, ne fermez pas la connexion avant d'avoir refermé les objets.

## 6.31.12 pg_connect() : Ouvre une connexion PostgreSQL

resource **pg_connect** ( string connection_string )

pg_connect retourne un index de connexion en cas de succès, et FALSE sinon.

pg_connect ouvre une connexion à un serveur PostgreSQL, grâce à la chaîne de connexion connection_string . Les arguments doivent être placé entre guillemets.

Exemples avec pg_connect

```
<?php
 $dbconn = pg_connect("dbname=marie");
 //connexion à une base de données nommée "marie"
 $dbconn2 = pg_connect("host=localhost port=5432 dbname=marie");
 //connexion à une base de données nommée "marie" sur l'hôte "localhost" sur le port "5432"
 $dbconn3 = pg_Connect ("host=sheep port=5432 dbname=marie user=mouton password=baaaa");
 //connexion à une base de données nommée "marie" sur le serveur "mouton" avec
 // un nom d'utilisateur et le mot de passe associé
 $conn_string = "host=sheep port=5432 dbname=test user=lamb password=bar";
```

```
 $dbconn4 = pg_connect ($conn_string);
 //connexionà une base de données appelée "test" sur l'hôte "sheep" avec le nom d'utilisateur
?>
```

Les arguments disponibles comprennent notamment connection_string includes host , port , tty , options , dbname , user et password .

pg_connect retourne un index de connexion qui sera nécessaire aux autres fonctions PostgreSQL. Vous pouvez ouvrir plusieurs connexions simultanées.

Si un deuxième appel à pg_connect est fait avec les mêmes arguments, aucune nouvelle connexion ne sera établie, mais la connexion précédente sera retournée.

L'ancienne syntaxe $conn = pg_connect("host", "port", "options", "tty", "dbname") est obsolète.

Voir aussi pg_pconnect , pg_close , pg_host , pg_port , pg_tty , pg_options et pg_dbname .

## 6.31.13 pg_connection_busy() : Vérifie si la connexion PostGreSQL est occupée

bool **pg_connection_busy** ( resource connection )

pg_connection_busy retourne TRUE si la connexion est occupe. Si elle est occupée, une requête a déjà été lancée, et est en cours. Si pg_get_result est appelé, elle sera alors bloquée.

Exemple avec pg_connection_busy

```
<?php
   $dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");
   $bs = pg_connection_busy($dbconn);
   if ($bs) {
       echo 'La connexion est occupée';
   }
   else {
      echo 'La connexion est libre';
   }
?>
```

Voir aussi pg_connection_status et pg_get_result .

## 6.31.14 pg_connection_reset() : Relance la connexion au serveur PostGresSQL

bool **pg_connection_reset** ( resource connection )

pg_connection_reset effectue une reconnexion au serveur, avec les mêmes paramètres que lors de la connexion précédente avec connection . Cette fonction est pratique pour le traitement des erreurs. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec pg_connection_reset

```
<?php
   $dbconn = pg_connect("dbname=publisher") or die ("Impossible de se connecter");
   $dbconn2 = pg_connection_reset($dbconn);
   if ($dbconn2) {
       echo 'Redémarrage réussi';
   }
   else {
       echo 'Redémarrage échoué';
   }
?>
```

Voir aussi pg_connect , pg_pconnect et pg_connection_status .

## 6.31.15 pg_connection_status() : Lit le statut de la connexion PostGreSQL

int **pg_connection_status** ( resource connection )

pg_connection_status retourne le statut de la connexion connection . Les valeurs possibles sont PGSQL_CONNECTION_OK (connexion correcte) et PGSQL_CONNECTION_BAD (connexion abandonnée). La valeur de retour entière 0 indique une connexion valide.

Exemple avec pg_connection_status

```
<?php
   $dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");
   $stat = pg_connection_status($dbconn);
   if ($stat === 0) {
       echo 'Connexion ok';
   } else {
       echo 'Connexion ko';
   }
?>
```

Voir aussi pg_connection_busy .

## 6.31.16 pg_convert() : Convertit un tableau associatif en une commande PostGreSQL

array **pg_convert** ( resource connection , string table_name , array assoc_array , *int options* )

pg_convert vérifie et convertit le tableau associatif assoc_array en une requête SQL valide. Pour que pg_convert fonctionne, il faut que la table table_name existe, et contienne au moins autant de colonnes que le tableau assoc_array a d'éléments. Les noms des champs de table_name doit correspondre aux index du tableau assoc_array . pg_convert retourne un tableau avec les valeurs converties, en cas de succès, et sinon, FALSE .

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Voir aussi pg_meta_data .

## 6.31.17 pg_copy_from() : Insère des lignes dans une table à partir d'un tableau

bool **pg_copy_from** ( resource connection , string table_name , array rows , *string delimiter* , *string null_as* )

pg_copy_from insère les éléments du tableau rows dans la tableau table_name . Cette fonction utilise la commande SQL interne COPY FROM . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi pg_copy_to .

## 6.31.18 pg_copy_to() : Copie une table dans un tableau

array **pg_copy_to** ( resource connection , string table_name , *string delimiter* , *string null_as* )

pg_copy_to copie la table table_name dans un tableau, et le retourne. Cette fonction utilise la commande interne SQL COPY TO pour insérer les tableaux. En cas d'erreur, la fonction retourne FALSE .

Voir aussi pg_copy_from

## 6.31.19 pg_dbname() : Nom de la base de données

string **pg_dbname** ( resource connection )

pg_dbname retourne le nom de la base de données PostgreSQL associée à l'index de connexion connection, ou FALSE si connection n'est pas valide.

Exemple avec pg_dbname

```
<?php
   error_reporting(E_ALL);

   pg_connect ("host=localhost port=5432 dbname=mary");
   echo pg_dbname();
?>
```

## 6.31.20 pg_delete() : Efface des lignes PostGreSQL

mixed **pg_delete** ( resource connection , string table_name , array assoc_array , *int options* )

pg_delete efface les lignes de la table table_name , spécfiées par le tableau associatif assoc_array , qui est bâti sous la forme champ=>valeur . Si option est spécifié, pg_convert est appliqué à assoc_array avec cette option.

Exemple avec pg_delete

```
<?php
    $db = pg_connect ('dbname=foo');
   // C'est sans proble, car $_POST est convertit automatiquement
   $res = pg_delete($db, 'post_log', $_POST);
   if ($res) {
       echo "Les données POSTées ont été effacées : $res\n";
   }
   else {
       echo "Les données d'entrées sont erronées.\n";
   }
?>
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Voir aussi pg_convert .

## 6.31.21 pg_end_copy() : Synchronise avec le serveur PostgreSQL

bool **pg_end_copy** ( *resource connection* )

pg_end_copy synchronise le client PostgreSQL (ici PHP) avec le serveur, après une opération de copie. Il faut utiliser cette fonction, sous peine de recevoir une erreur "out of sync" (désynchronisé). Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Pour plus de détails et un exemple voyez : pg_put_line .

## 6.31.22 pg_escape_bytea() : Protège le caractères d'une chaîne binaire en mode bytea

string **pg_escape_bytea** ( string data )

pg_escape_bytea protège les caractères de la chaîne data avec le mode bytea. La chaîne protégée est retournée.

| Note |
| --- |
| Lorsque vous utilisez une commande SELECT avec des données de type bytea, PostgreSQL retourne des valeurs octales, préfixées avec des anti−slash \ (e.g. \032). Les utilisateurs doivent effectuer la conversion en format binaire eux−même.<br><br>pg_escape_bytea requiert PostgreSQL 7.2 ou plus récent. Avec PostgreSQL 7.2.0 et 7.2.1, les données de type bytea doivent être transtypée lorsque vous activez le support des chaînes de caractères multi−octets. i.e. INSERT INTO test_table (image) VALUES ('$image_escaped'::bytea); . PostgreSQL 7.2.2 ou plus récent ne requiert pas cette manipulation. Toutefois, si le client et le serveur n'utilisent pas le même jeux de caractères, il peut arriver des erreurs. Il faut alors forcer le transtypage manuellement. |

Voir aussi pg_unescape_bytea et pg_escape_string .

## 6.31.23 pg_escape_string() : Protège une chaîne de caractères

string **pg_escape_string** ( string data )

pg_escape_string protège la chaîne de caractères data . Elle retourne la chaîne ainsi protégée. L'utilisation de cette fonction est recommandée, à la place de addslashes .

| Note |
| --- |
| Cette fonction requiert PostgreSQL 7.2 ou plus récent. |

Voir aussi pg_escape_bytea .

## 6.31.24 pg_fetch_all() : Lit toutes les lignes d'un résultat

array **pg_fetch_all** ( resource result )

pg_fetch_all retourne un tableau qui contient toutes les lignes du résultat result . Elle retourne FALSE s'il n'y a plus de lignes disponibles.

Exemple avec pg_fetch_all

```
<?php
$conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
```

```
}

$result = pg_query ($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Une erreur est survenue.\n";
    exit;
}

$arr = pg_fetch_all ($result, 0, PGSQL_NUM);

var_dump($arr);

?>
```

Voir aussi pg_fetch_row , pg_fetch_array , pg_fetch_object et pg_fetch_result .

## 6.31.25 pg_fetch_array() : Lit une ligne de résultat PGSQL dans un tableau

array **pg_fetch_array** ( resource result , *int row* , *int result_type* )

pg_fetch_array retourne un tableau qui contient à la ligne demandée, dans le résultat identifiée par result , et FALSE , s'il ne reste plus de lignes.

pg_fetch_array est une version évoluée de pg_fetch_row . En plus de proposer un tableau à indice numérique, elle peut aussi enregistrer les données dans un tableau associatif, en utilisant les noms des champs comme clés.

L'argument optionnel result_type de pg_fetch_array est une constante, qui peut prendre les valeurs suivantes : PGSQL_ASSOC , PGSQL_NUM et PGSQL_BOTH . pg_fetch_array retourne un tableau associatif dont les clés sont les noms de colonnes, avec la constante PGSQL_ASSOC ; les index de colonnes avec PGSQL_NUM , et les deux avec la constante PGSQL_BOTH La valeur par défaut est PGSQL_BOTH .

| Note |
| --- |
| result_type a été ajoutée en PHP 4.0. |

Il est important de noter que pg_fetch_array n'est pas significativement plus lent que pg_fetch_row , tandis qu'elle fournit un confort d'utilisation notable.

Pour plus de détails, reportez−vous à pg_fetch_row .

Exemple avec pg_fetch_array

```
 <?php
 $conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Erreur de connexion.\n";
    exit;
}

$result = pg_query ($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Erreur durant la requête.\n";
    exit;
}

$arr = pg_fetch_array ($result, 0, PGSQL_NUM);
echo $arr[0] . " <− array\n";

$arr = pg_fetch_array ($result, 1, PGSQL_ASSOC);
echo $arr["author"] . " <− array\n";
?>
```

Voir aussi pg_fetch_row , pg_fetch_object et pg_fetch_result .

| Note |
| --- |

Depuis 4.1.0, row est devenu optionnel. Appeler pg_fetch_array avec incrémenter le pointeur interne de 1.

## 6.31.26 pg_fetch_assoc() : Lit une ligne de résultat PGSQL sous forme de tableau numérique

array **pg_fetch_assoc** ( resource result , *int* *row* )

pg_fetch_assoc retourne un tableau associatif qui contient la ligne en cours dans le résultat result . pg_fetch_assoc retourne FALSE si il n'y a plus de lignes.

pg_fetch_assoc est une version améliorée de pg_fetch_row . En plus de stocker les valeurs dans un tableau a index numérique, elle stocke aussi les données dans un tableau associatif, en utilisant les noms de colonnes comme index.

row est le numéro de ligne à lire. Par défaut, il vaut 0.

pg_fetch_assoc n'est pas significativement plus lente que pg_fetch_row , et elle apporte un confort d'utilisation apréciable.

Exemple avec pg_fetch_assoc

```
<?php
 $conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

$result = pg_query ($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Une erreur est survenue.\n";
    exit;
}

$arr = pg_fetch_assoc ($result, 1, PGSQL_ASSOC);
echo $arr["author"] . " <- array\n";
?>
```

Voir aussi pg_fetch_row , pg_fetch_array , pg_fetch_object et pg_fetch_result .

## 6.31.27 pg_fetch_object() : Lit une ligne de résultat PGSQL dans un objet

object **pg_fetch_object** ( resource result , *int* *row* , *int* *result_type* )

pg_fetch_object retourne un objet dont les membres sont les champs de la ligne demandée, ou FALSE , si il n'y a plus de lignes.

pg_fetch_object est similaire à pg_fetch_array , avec une différence majeure : c'est un objet qui est retourné, au lieu d'un tableau. Par conséquent, cela signifie que vous ne pouvez accéder aux membres qu'avec leur nom, et non plus leur offset (les nombres ne sont pas autorisés comme nom de membre).

L'argument optionnel result_type de result_type est une constante qui peut prendre les valeurs suivantes : PGSQL_ASSOC , PGSQL_NUM et PGSQL_BOTH .

Note

result_type a été ajouté dans PHP 4.0.

Au niveau vitesse, pg_fetch_object est aussi rapide que pg_fetch_row et presque aussi rapide que pg_fetch_row (la différence est non significative).

Lecture d'un objet Postgres

```
 <?php
 $database = "verlag";
$db_conn = pg_connect ("host=localhost port=5432 dbname=$database");
if (!$db_conn): ?>
    <H1>Impossible de se connecter à la base PostgreSQL<?php echo $database ?></H1> <?php
    exit;
endif;

$qu = pg_query ($db_conn, "SELECT * FROM verlag ORDER BY autor");
$row = 0; // PostgreSQL requiert un compteur de ligne que d'autres n'ont pas

while ($data = pg_fetch_object ($qu, $row)) {
    echo $data->autor." (";
    echo $data->jahr .")): ";
    echo $data->titel."<BR>";
    $row++;
}
?>
<PRE>
<?php
$fields[] = Array ("autor", "Author");
$fields[] = Array ("jahr",  "  Year");
$fields[] = Array ("titel", " Title");

$row= 0; // PostgreSQL requiert un compteur de ligne que d'autres n'ont pas
while ($data = pg_fetch_object ($qu, $row)) {
    echo "----------\n";
    reset ($fields);
    while (list (,$item) = each ($fields)):
        echo $item[1].": ".$data->$item[0]."\n";
    endwhile;
    $row++;
}
echo "----------\n";
 ?>
</PRE>
 <?php
pg_free_result ($qu);
pg_close ($db_conn);
?>
```

Note

Depuis 4.1.0, row est devenu optionnel. Appeler pg_fetch_array avec incrémenter le pointeur interne de 1.

Voir aussi pg_query , pg_fetch_array , pg_fetch_assoc , pg_fetch_row et pg_fetch_result .

## 6.31.28 pg_fetch_result() : Retourne les valeurs d'un résultat

mixed **pg_fetch_result** ( resource result , int row , mixed field )

pg_fetch_result retourne les valeurs stockées dans la ressource de résultat result , retournée par pg_query . row est un entier. field est le nom du champ (sous forme de chaîne) ou un index de colonne (sous forme d'entier). row et field spécifie quelles cellules de la table de résultat doivent être lu. Le lignes sont numérotées à partir de 0. Au lieu de nommer le champ, vous pouvez utiliser le numéro de colonne. L'indexation des colonnes commence à 0.

PostgreSQL dispose de nombreux types, et seuls les types basiques sont supportés ici. Tous les types integer sont retournées sous forme de integer . Tous les types de float, et nombres réels sont retournées sous forme de float . Les booléens sont retournées comme des "t" ou "f". Tous les autres types, y compris les tableaux, sont retournées sous forme de chaînes formatées, de la même manière que PostgreSQL vous les afficherait dans le client psql .

## 6.31.29 pg_fetch_row() : Lit une ligne dans un tableau

array **pg_fetch_row** ( resource result , int row )

pg_fetch_row retourne un tableau qui contient les données de la ligne demandée, ou FALSE , si il ne reste plus de lignes.

pg_fetch_row lit une ligne dans le résultat associé à l'index result . La ligne est retournée sous la forme d'un tableau. La ligne est retournée sous la forme d'un tableau, qui commence à l'index 0.

Les appels ultérieurs à pg_fetch_row retourneront la ligne d'après, ou bien FALSE , lorsqu'il n'y aura plus de lignes.

Postgres retourne une ligne

```
 <?php
$conn = pg_pconnect("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}
$result = pg_exec($conn, "SELECT * FROM authors");
if (!$result) {
    echo "Une erreur est survenue.\n";
    exit;
}
$num = pg_numrows($result);
for ($i=0; $i<$num; $i++) {
  $r = pg_fetch_row($result, $i);
  for ($j=0; $j<count($r); $j++) {
    echo "$r[$j] ";
  }
  echo "<BR>";
}
?>
```

Voir aussi pg_query , pg_fetch_array , pg_fetch_object et pg_fetch_result .

Note

Depuis PHP 4.1.0, row est devenu un paramètre optionnel. Appeler pg_fetch_row en l'omettant va incrémenter le pointeur interne de résultat de 1.

## 6.31.30 pg_field_is_null() : Teste si un champ PostGreSQL est à NULL

int **pg_field_is_null** ( resource result , int row , mixed field )

pg_field_is_null teste si un champ est à NULL . pg_field_is_null retourne 0 si le champ n'est pas NULL . pg_field_is_null retourne 1 si le champ est à NULL . Le champ peut être identifié avec son nom ou son index numérique (commençant à 0).

Exemple avec pg_field_is_null

```
 <?php
    $dbconn = pg_connect("dbname=publisher") or die ("Impossible de se connecter");
    $res = pg_query($dbconn, "select * from authors where author = 'Orwell'");
    if ($res) {
        if (pg_field_is_null($res, 0, "annee") == 1) {
            echo "La valeur du champ \"annee\" est null.\n";
        }
        if (pg_field_is_null($res, 0, "year") == 0) {
            echo "La valeur du champ \"annee\" n'est pas null.\n";
      }
   }
?>
```

Note

Auparavant, cette fonction s'appelait pg_fieldisnull() .

## 6.31.31 pg_field_name() : Retourne le nom d'un champ PostGreSQL

string **pg_field_name** ( resource result , int field_number )

pg_field_name va retourne le nom du champ qui occupe la colonne numéro field_number dans le résultat result . La numérotation des champs commence à 0.

Lire des informations sur les champs PostGreSQL

```php
<?php
   $dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");

   $res = pg_query($dbconn, "select * from authors where author = 'Orwell'");
   $i = pg_num_fields($res);
   for ($j = 0; $j < $i; $j++) {
       echo "column $j\n";
       $fieldname = pg_field_name($res, $j);
       echo "Champ : $fieldname\n";
       echo "Taille affichée : ".pg_field_prtlen($res, $fieldname)." characters\n";
       echo "Taille de stockage : ".pg_field_size($res, $j)." bytes\n";
       echo "Type de champ : ".pg_field_type($res, $j)." \n\n";
}
?>
```

L'exemple va afficher :

```
column 0
Champ : author
Taille affichée : 6 characters
Taille de stockage : −1 bytes
Type de champ : varchar

column 1
Champ : year
Taille affichée : 4 characters
Taille de stockage : 2 bytes
Type de champ : int2

column 2
Champ : title
Taille affichée : 24 characters
Taille de stockage : −1 bytes
Type de champ : varchar
```

Note

Auparavant, cette fonction s'appelait pg_fieldname() .

Voir aussi pg_field_num .

## 6.31.32 pg_field_num() : Retourne le numéro d'une colonne

int **pg_field_num** ( resource result , string field_name )

pg_field_num retourne le numéro de la colonne, dont le nom est field_name , dans le résultat result . La numérotation des champs commence à 0. pg_field_num retournera −1 en cas d'erreur.

Note

Auparavant, cette fonction s'appelait pg_fieldnum() .

Voir aussi pg_field_name .

## 6.31.33 pg_field_prtlen() : Retourne la taille imprimée

int **pg_field_prtlen** ( resource result , int row_number , string field_name )

pg_field_prtlen retourne la taille imprimée (nombre de caractères) d'une valeur donnée dans un résultat PostgreSQL. La numérotation des lignes commence à 0. pg_field_prtlen retourne −1 en cas d'erreur.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_fieldprtlen() . |

Voir aussi pg_field_size .

## 6.31.34 pg_field_size() : Retourne la taille interne de stockage d'un champ donné.

int **pg_field_size** ( resource result , int field_number )

pg_field_size retourne la taille interne de stockage d'un champ donné, en octets. pg_field_size retourne −1 si la taille est variable. pg_field_size retourne FALSE en cas d'erreur. La numérotation des colonnes commence à 0.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_fieldsize() . |

Voir aussi pg_field_prtlen et pg_field_type .

## 6.31.35 pg_field_type() : Retourne le type d'un champ PostGreSQL donné par index

string **pg_field_type** ( resource result , int field_number )

pg_field_type retourne une chaîne contenant le type du champ donné par son index field_number . La numérotation des champ commence à 0.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_fieldtype() . |

Voir aussi pg_field_prtlen et pg_field_name .

## 6.31.36 pg_free_result() : Libère la mémoire

bool **pg_free_result** ( resource result )

pg_free_result n'est vraiment utile que si vous risquez d'utiliser trop de mémoire durant votre script. La mémoire occupée par les résultats est automatiquement libérée à la fin du script. Mais, si vous êtes sûr de ne pas avoir besoin du résultat ultérieurement, vous pouvez appeler pg_free_result avec l'index de résultat comme argument, et la mémoire sera libérée. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_freeresult() . |

Voir aussi pg_query .

## 6.31.37 pg_get_notify() : Lit le message de NOTIFY

array **pg_get_notify** ( resource connection , *int result_type* )

pg_get_notify reçoit le message de NOTIFY envoyé par une commande SQL NOTIFY . Pour lire le message associé, utilisez la commande LISTEN . Si il y a un message de notification avec cette connexion, un tableau sera retourné avec le message et le PID du serveur. Si il n'y a pas de message, pg_get_notify retourne FALSE .

Exemple avec pg_get_notify

```
<?php
 $conn = pg_pconnect("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

// ecoute le message 'author_updated' des autres processus
pg_query($conn, 'LISTEN author_updated;');
$notify = pg_get_notify($conn);
if (!$notify)
    echo "Aucun message\n";
else
    print_r($notify);
?>
```

Voir aussi pg_get_pid .

## 6.31.38 pg_get_pid() : Lit l'identifiant de processus du serveur

int **pg_get_pid** ( resource connection )

pg_get_pid lit l'identifiant de processus du serveur PostGreSQL. Le PID est pratique pour vérifier si un message de NOTIFY a été envoyé par un autre processus ou pas.

Exemple avec pg_get_pid

```
<?php
 $conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

// PID du serveur. Utilisez alors le PID avec pg_get_notify()
$pid = pg_get_pid($conn);
?>
```

Voir aussi pg_get_notify .

## 6.31.39 pg_get_result() : Lit un résultat asynchrone

resource **pg_get_result** ( *resource connection* )

pg_get_result lit le résultat asynchrone d'une requête exécutée par pg_send_query . pg_send_query peut envoyer plusieurs requêtes à PostgreSQLet pg_get_result sert à lire les résultats un par un. pg_get_result retourne une ressource de résultat. Si il n'y a pas de résultat, pg_get_result retourne FALSE .

## 6.31.40 pg_host() : Retourne le nom d'hôte

string **pg_host** ( resource connection )

pg_host retourne le nom d'hôte associé à l'index de connexion PostgreSQL.

Voir aussi pg_connect et pg_pconnect .

## 6.31.41 pg_insert() : Insère un tableau dans une table

bool **pg_insert** ( resource connection , string table_name , array assoc_array , *int options* )

pg_insert insère le tableau assoc_array (bâtit sur le modèle champ=>valeur dans la table table_name . Si options est spécifié, pg_convert est appliqué à assoc_array avec cette option. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Si options est spécifiée, pg_insert s'applique à assoc_array avec l'option spécifiée.

Exemple avec pg_insert

```
<?php
    $db = pg_connect ('dbname=foo');
   // C'est correct, car $_POST est convertit automatiquement
   $res = pg_insert($db, 'post_log', $_POST);
   if ($res) {
       echo "Les données POSTées ont pu être enregistrée avec succès.\n";
   }
   else {
       echo "Il y a un problème avec les données.\n";
   }
?>
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Voir aussi pg_convert .

## 6.31.42 pg_last_error() : Lit le dernier message d'erreur sur la connexion

string **pg_last_error** ( *resource connection* )

pg_last_error retourne le dernier message d'erreur qui a été enregistré sur la connexion connection .

Les messages d'erreur peuvent être écrasé par des appels internes à l'extension PostgreSQL(libpq) : il se peut que le message retourné ne soit pas approprié, notamment si plusieurs erreurs ont eu lieu dans le module.

Utilisez pg_result_error , pg_result_status et pg_connection_status pour améliorer la gestion des erreurs.

Note

pg_last_error s'appelait pg_errormessage() .

Voir aussi pg_result_error .

## 6.31.43 pg_last_notice() : Retourne la dernière note du serveur PostgreSQL

string **pg_last_notice** ( resource connection )

pg_last_notice retourne la dernière note du serveur PostgreSQL, spécifiée par connection . Le serveur PostgreSQL envoie des notes dans plusieurs cas, comme par exemple si la transaction ne peut être continuée. Avec pg_last_notice , vous pouvez éviter des requêtes inutiles en vérifiant si des notes liées à votre transaction ont été émises.

| Attention |
| --- |
| Cette fonction est EXPERIMENTALE et elle n'est pas complètement implémentée. pg_last_notice a été ajoutée en PHP 4.0.6. Cependant, PHP 4.0.6 avait un problème avec la gestion des notes. L'utilisation du module PostgreSQL avec PHP 4.0.6 n'est pas recommandé, même si vous n'utilisez pas pg_last_notice .<br><br>Cette fonction est totaltement implémentée en PHP 4.3.0. Les versions plus ancienne de PHP ignore les données de connexion au serveur. |

Le suivi des notes peut être rendu optionnel en mettant à 1 la directive de configuration pgsql.ignore_notice du php.ini à partir de PHP 4.3.0.

L'enregistrement des notes peut être rendu optionnel en mettant la directive de configuration pgsql.log_notice du php.ini à 0, à partir de PHP 4.3.0. A moins que pgsql.ignore_notice ne soit à 0, les notes ne seront pas enregistrées.

Voir aussi pg_query et pg_last_error .

## 6.31.44 pg_last_oid() : Retourne l'identifiant du dernier objet

int **pg_last_oid** ( resource result )

pg_last_oid sert à lire le dernier oid assigné à un ligne, qui est stocké dans la ressource result , obtenu lors de la dernière commande pg_query . Il faut que cette commande ait été un INSERT. pg_last_oid retourne un entier positif si il y a eu un oid valide. pg_last_oid retourne FALSE si une erreur est survenue,, ou si la commande utilisée avec pg_query n'était pas un INSERT, ou que cette commande a échoué.

Le champ OID est devenu optionnel depuis PostgreSQL 7.2. Lorsque le champ OID n'est pas défini dans la table, le programmeur doit utiliser pg_result_status pour vérifier si la ligne a été correctement insérée.

| Note |
| --- |
| pg_last_oid s'appelait pg_getlastoid() . |

Voir aussi pg_query et pg_result_status .

## 6.31.45 pg_lo_close() : Ferme un objet de grande taille PostgreSQL

bool **pg_lo_close** ( resource large_object )

pg_lo_close ferme un objet de type Inversion Large Object. fd est un descripteur de fichier, obtenu avec pg_lo_open .

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_loclose() . |

Voir aussi pg_lo_open , pg_lo_create et pg_lo_import .

## 6.31.46 pg_lo_create() : Crée un objet de grande taille PostgreSQL

int **pg_lo_create** ( resource connection )

pg_lo_create crée un objet de type Inversion Large Object et retourne son Oid. conn doit être une connexion valide avec une base de données PostgreSQL. Les modes d'accès PostgreSQL INV_READ, INV_WRITE, et INV_ARCHIVE ne sont pas supportés : l'objet peut toujours être créé, avec des droits d'accès en lecture et écriture. Le mode INV_ARCHIVE a été supprimé des bases PostgreSQL (version 6.3 et ultérieur).

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_locreate() . |

## 6.31.47 pg_lo_export() : Exporte un objet de grande taille vers un fichier

bool **pg_lo_export** ( int oid , string pathname , *resource connection* )

pg_lo_export exporte un objet de grande taille dans un fichier. oid est un identifiant d'objet de grande taille qui sera exporté dans le fichier filename , qui spécifie son chemin. RetourneFALSEsi une erreur survient, et TRUE en cas de succès. N'oubliez pas que la manipulation d'un objet de grande taille dans PostgreSQL doit intervenir dans une transaction.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_loexport() . |

Voir aussi pg_lo_import .

## 6.31.48 pg_lo_import() : Importe un objet de grande taille depuis un fichier

int **pg_lo_import** ( *resource connection* , string pathname )

Dans les versions antérieures à PHP 4.2.0, la syntaxe de cette fonction était différente. Voici la définition :

int **pg_lo_import** ( string pathname , *resource connection* )

pathname est le chemin jusqu'à un fichier qui servira de source pour créer un objet de grande taille. pg_lo_import retourne FALSE en cas d'erreur, et sinon un identifiant d'objet, créé directement à la bonne taille.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_loimport() . |

| Note |
| --- |
| Lorsque le safe−mode est activé, PHP vérifie si le fichier/dossier que vous allez utiliser a le même UID que le script qui est actuellement exécuté. |

Voir aussi pg_lo_export et pg_lo_open .

## 6.31.49 pg_lo_open() : Ouvre un objet de grande taille PostgreSQL

resource **pg_lo_open** ( resource connection , int oid , string mode )

pg_lo_open ouvre un objet de type Inversion Large Object et retourne un descripteur de fichier pour cet objet. Le descripteur de fichier contient les informations de connexion. Ne refermez pas la connexion avant d'avoir fermé l'objet. objoid est un Oid valide de Large Object, et mode peut prendre es valeurs suivantes : "r", "w", ou "rw".

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_loopen() . |

Voir aussi pg_lo_close et pg_lo_create .

## 6.31.50 pg_lo_read_all() : Lit un objet de grande taille en totalité

int **pg_lo_read_all** ( resource large_object )

pg_lo_read_all lit un objet de grande taille en totalité et le passe directement au client, après les en−têtes adéquates. Cette fonction est prévue pour transmettre des sons ou des images.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_loreadall() . |

Voir aussi pg_lo_read .

## 6.31.51 pg_lo_read() : Lit un objet de grande taille

string **pg_lo_read** ( resource large_object , int len )

pg_lo_read lit au plus len octets d'un objet de grande taille, et retourne les données sous la forme d'une chaîne. large_object est un identifiant valide d'objet de grande taille, et len indique la taille

maximale de mémoire alloué à l'objet de grande taille.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_loread() . |

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

Voir aussi pg_lo_read_all .

## 6.31.52 pg_lo_seek() : Modifie la position dans un objet de grande taille

bool **pg_lo_seek** ( resource large_object , int offset , *int whence* )

pg_lo_seek modifie la position du pointeur dans l'objet de grande taille large_object . Le paramètre whence peut prendre les valeurs de PGSQL_SEEK_SET , PGSQL_SEEK_CUR ou PGSQL_SEEK_END .

Voir aussi pg_lo_tell .

## 6.31.53 pg_lo_tell() : Retourne la position courante dans un objet de grande taille

int **pg_lo_tell** ( resource large_object )

pg_lo_tell retourne la position courante du pointeur de lecture sur l'objet de grande taille large_object . C'est le nombre d'octets depuis le dédut du fichier.

Voir aussi pg_lo_seek .

## 6.31.54 pg_lo_unlink() : Efface un objet de grande taille

bool **pg_lo_unlink** ( resource connection , int oid )

pg_lo_unlink efface l'objet de grande taille dont l'identifiant est oid , pour la connexion connection . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Pour utiliser un objet de grande taille (lo), il est nécessaire de le faire dans une transaction.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_lounlink() . |

Voir aussi pg_lo_create et pg_lo_import .

## 6.31.55 pg_lo_write() : Ecrit un objet de grande taille PGSQL

int **pg_lo_write** ( resource large_object , string data )

pg_lo_write écrit dans l'objet de grande taille autant de données possible, issues de la variable buf et retourne le nombre d'octets réellement écrits, ou FALSE en cas d'erreur. fd est un descripteut

d'objet de grande taille, obtenu avec pg_lo_open .

Pour manipuler un objet de grande taille (lo), il est néessaire de placer les opérations dans un bloc de transaction.

Voir aussi pg_lo_create et pg_lo_open .

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_lowrite() . |

## 6.31.56 pg_meta_data() : Lit les données méta de la table PostGreSQL

array **pg_meta_data** ( resource connection , string table_name )

pg_meta_data retourne la définition de la table table_name sous forme de tableau. Si une erreur survient, pg_meta_data retourne FALSE .

Lire les méta−données d'une table PostGreSQL

```php
<?php
   $dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");

   $meta = pg_meta_data($dbconn,'authors');
   if (is_array ($meta)) {
       echo '<pre>';
       var_dump ($meta);
       echo '</pre>';
   }
?>
```

L'exemple ci−dessus pourrait afficher :

```
array(3) {
["author"]=>
array(5) {
["num"]=>
int(1)
["type"]=>
string(7) "varchar"
["len"]=>
int(−1)
["not null"]=>
bool(false)
["has default"]=>
bool(false)
}
["year"]=>
array(5) {
["num"]=>
int(2)
["type"]=>
string(4) "int2"
["len"]=>
int(2)
["not null"]=>
bool(false)
["has default"]=>
bool(false)
}
["title"]=>
array(5) {
["num"]=>
int(3)
["type"]=>
string(7) "varchar"
["len"]=>
int(−1)
["not null"]=>
bool(false)
["has default"]=>
bool(false)
}
}
```

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son |

nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Voir aussi pg_convert .

## 6.31.57 pg_num_fields() : Retourne le nombre de champ

int **pg_num_fields** ( resource result )

pg_num_fields retourne le nombre de champs (ou colonnes) d'un résultat PostgreSQL. L'argument doit être un identifiant de résultat valide retourné par pg_exec . Cette fonction retournera −1 en cas d'erreur.

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_numfields() . |

Voir aussi pg_num_rows et pg_affected_rows .

## 6.31.58 pg_num_rows() : Retourne le nombre de lignes PostgreSQL

int **pg_num_rows** ( resource result )

pg_num_rows retourne le nombre de lignes d'un résultat PostgreSQL. L'argument doit être un identifiant de résultat valide retourné par pg_exec . Cette fonction retournera −1 en cas d'erreur.

| Note |
| --- |
| Utilisez la fonction pg_affected_rows pour connaître le nombre de lignes affectées par les commandes SQL INSERT, UPDATE et DELETE. |

| Note |
| --- |
| Auparavant, cette fonction s'appelait pg_numrows() . |

Voir aussi pg_num_fields et pg_affected_rows .

## 6.31.59 pg_options() : Retourne les options PostgreSQL

string **pg_options** ( resource connection )

pg_options retourne une chaîne contenant les options de la connexion PostgreSQL connection .

## 6.31.60 pg_pconnect() : Etablit une connexion PostgreSQL persistante

resource **pg_pconnect** ( string connection_string )

pg_pconnect retourne une ressource de connexion persistante en cas de succès, ou FALSE en cas d'erreur.

Pour une description détaillée de la chaîne connection_string , voyez pg_connect .

Pour activer les connexions persistantes, la directive de configuration pgsql.allow_persistent du php.ini doit être mise à "On" (ce qui est sa valeur par défaut). Le nombre maximal de connexions peut être limité grâce à la directive de configuration pgsql.max_persistent dans le fichier php.ini (par défaut, elle vaut −1, c'est à dire pas de limite). Le nombre total de connexions peut être configuré avec la directive pgsql.max_links du fichier php.ini .

pg_close ne referemera pas les connexions persistantes générées par pg_pconnect .

Voir aussi pg_connect et la section sur les connexions persistante pour plus d'informations.

## 6.31.61 pg_ping() : Pingue la connexion à la base

bool **pg_ping** ( resource connection )

pg_ping pingue la connexion à la base de données, et essaie de se reconnecter si la connexion est perdue. pg_ping retourne TRUE si la connexion est active, et FALSE sinon.

Exemple avec pg_ping

```
<?php
 $conn = pg_pconnect ("dbname=publisher");
if (!$conn) {
    echo "Une erreur est survenue.\n";
    exit;
}

if (!pg_ping($conn))
    die("La connexion est morte\n");
?>
```

Voir aussi pg_connection_status et pg_connection_reset .

## 6.31.62 pg_port() : Retourne le numéro de port

int **pg_port** ( resource connection )

pg_port retourne le numéro de port de la connexion identifiée connection .

## 6.31.63 pg_put_line() : Envoie une chaîne au serveur PostgreSQL

bool **pg_put_line** ( *resource connection* , string data )

pg_put_line envoie une chaîne (terminée par NULL ) au serveur PostgreSQL. Ceci est pratique pour effectuer des insertions très rapides dans une table, initiée par une opération de copie PostgreSQL copy−operation. Le caractère final NULL est automatiquement ajouté. Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec..

Note

Notez que l'application doit explicitement ajouter les deux caractères "\." à la fin de la chaîne pour indiquer au serveur qu'elle a finit d'envoyer des données.

Insertion à grande vitesse dans une table

```
<?php
   $conn = pg_pconnect("dbname=foo");
   pg_exec($conn, "create table bar (a int4, b char(16), d float8)");
   pg_exec($conn, "copy bar from stdin");
   pg_put_line($conn, "3\tBonjour le monde\t4.5\n");
   pg_put_line($conn, "4\tAu revoir le monde\t7.11\n");
   pg_put_line($conn, "\\.\n");
```

```
    pg_end_copy($conn);
?>
```

Voir aussi pg_end_copy .

## 6.31.64 pg_query() : Exécute une requête

resource **pg_query** ( resource connection , string query )

pg_query retourne une ressource de résultat si la requête a pu être exécutée. Elle retourne FALSE en cas d'échec, ou si la connexion n'est pas valide. Le détails sur les erreurs rencontrées est accessible via la fonction pg_last_error , si la connexion est valide. pg_last_error émet une requête SQL au serveur PostgreSQL, spécifé avec le paramètre connection . connection doit être une connexion valide, retournée par la fonction pg_connect ou pg_pconnect . La valeur retournée par cette fonction est une ressource de résultat. Pour accéder aux lignes lues, il faudra utiliser les autres fonctions PostgreSQL, telles que pg_fetch_array .

| Note |
| --- |
| connection est un paramètre optionnel de pg_query . Si connection est omis, la connexion par défaut sera utilisée. La connexion par défaut est la dernière connexion ouverte avec pg_connect ou pg_pconnect .<br><br>Bien que connection puisse être omis, il n'est pas recommandé de le faire, car il peut se révéler difficile de retrouver la connexion dans un script. |

| Note |
| --- |
| Cette fonction s'appelait pg_exec() . pg_exec() est toujours disponible pour des raisons de compatibilité, mais les utilisateurs sont encouragés à utiliser le nouveau nom. |

Voir aussi pg_connect , pg_pconnect , pg_fetch_array , pg_fetch_object , pg_num_rows et pg_affected_rows .

## 6.31.65 pg_result_error() : Lit le message d'erreur associé à un résultat

string **pg_result_error** ( resource result )

pg_result_error retourne le message d'erreur associé au résultat result . Par conséquent, l'utilisateur a des chances d'obtenir un message d'erreur plus approprié que via pg_last_error .

Voir aussi pg_query , pg_send_query , pg_get_result , pg_last_error et pg_last_notice

## 6.31.66 pg_result_seek() : Modifie la ligne courant dans un résultat

array **pg_result_seek** ( resource result , int offset )

pg_result_seek choisit la ligne offset comme ligne courante dans le résultat result . pg_result_seek retourne FALSE , en cas d'erreur.

Voir aussi pg_fetch_row , pg_fetch_assoc , pg_fetch_array , pg_fetch_object et pg_fetch_result .

## 6.31.67 pg_result_status() : Lit le statut du résultat

int **pg_result_status** ( resource result )

pg_result_status lit le statut du résultat result . Les valeurs retournée possibles sont PGSQL_EMPTY_QUERY , PGSQL_COMMAND_OK , PGSQL_TUPLES_OK , PGSQL_COPY_TO , PGSQL_COPY_FROM , PGSQL_BAD_RESPONSE , PGSQL_NONFATAL_ERROR et PGSQL_FATAL_ERROR .

Voir aussi pg_connection_status .

## 6.31.68 pg_select() : Effectue une selection

array **pg_select** ( resource connection , string table_name , array assoc_array , *int options* )

pg_select effectue une requête de SELECT avec les lignes spécifiées par le tableau assoc_array , au format champ=>valeur . Lorsque la requête réussit, pg_select retouren un tableau contenant toutes les lignes et champs qui vérifient la condition spécifiée par assoc_array . Si options est spécifiée, pg_convert est appliquée à assoc_array avec l'option spécifiée.

Exemple avec pg_select

```
<?php
    $db = pg_connect ('dbname=foo');
   // C'est correct, car $_POST est convertit automatiquement
   $rec = pg_select($db, 'post_log', $_POST);
   if ($rec) {
       echo "Lignes lues\n";
       var_dump($rec);
   }
   else {
       echo "Problème dans les données utilisateur\n";
   }
?>
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Voir aussi pg_convert .

## 6.31.69 pg_send_query() : Exécute une requête PostGreSQL asynchrone

bool **pg_send_query** ( resource connection , string query ) bool **pg_send_query** ( string query )

pg_send_query exécute la requête query sur la connexion connection , de manière asynchrone. Contrairement à pg_query , cette fonction permet d'envoyer plusieurs requêtes à un serveur PostgreSQL et de lire les résultats un par un, avec la fonction pg_get_result . L'exécution du script n'est pas bloquée durant l'exécution de la requête. Utilisez pg_connection_busy pour vérifier si la connexion est encore occupée. Les requêtes peuvent être annulées avec pg_cancel_query .

Bien que l'on puiss envoyer plusieurs requêtes en même temps, il n'est pas possible d'envoyer plusieurs requêtes sur une connexion occupée. Si une requête est envoyée alors que la connexion est occupée, elle attendra que la requête précédente soit finie.

Exécute une requête PostGreSQL asynchrone

```php
<?php
   $dbconn = pg_connect("dbname=publisher") or die("Impossible de se connecter");

   if (!pg_connection_busy($dbconn)) {
       pg_send_query($dbconn,"select * from authors; select count(*) from authors;");
   }

   $res1 = pg_get_result($dbconn);
   echo "Premier appel à pg_get_result() : $res1\n";
   $rows1 = pg_num_rows($res1);
   echo "$res1 has $rows1 records\n\n";

   $res2 = pg_get_result($dbconn);
   echo "Second appel à pg_get_result() : $res2\n";
   $rows2 = pg_num_rows($res2);
   echo "$res2 has $rows2 records\n";
?>
```

L'exemple ci−dessus pourrait afficher :

```
Premier appel à pg_get_result() : Resource id #3
Resource id #3 has 3 records

Second appel à pg_get_result() : Resource id #4
Resource id #4 has 1 records
```

Voir aussi pg_query , pg_cancel_query , pg_get_result et pg_connection_busy .

## 6.31.70 pg_set_client_encoding() : Choisit l'encodage du client

int **pg_set_client_encoding** ( *resource connection* , string encoding )

pg_set_client_encoding fixe l'encodage du client. Elle retourne 0 en cas de succès, et −1 sinon.

encoding est l'encodage du client, et peut être SQL_ASCII, EUC_JP, EUC_CN, EUC_KR, EUC_TW, UNICODE, MULE_INTERNAL, LATINX (X=1...9), KOI8, WIN, ALT, SJIS, BIG5, WIN1250.

Note

Cette fonction requiert PHP−4.0.2 ou plus récent et PostgreSQL−7.0 ou plus récent.

Jadis, pg_set_client_encoding s'appelait pg_setclientencoding().

Voir aussi pg_client_encoding .

## 6.31.71 pg_trace() : Active le suivi d'une connexion PostgreSQL

bool **pg_trace** ( string pathname , *string mode* , *resource connection* )

pg_trace active le suivi des communications entre PHP et le serveur PostgreSQL. Cet historique sera enregistré dans un fichier. Pour comprendre ces lignes, il faut être familier avec le protocole de communication interne à PostgreSQL. Pour ceux qui le ne sont pas, elles peuvent être utiles pour suivre les requêtes et les erreurs : avec la commande grep '^To backend' trace.log , vous pourrez voir les requêtes réellement envoyées au serveur PostgreSQL.

pathname et mode sont les mêmes arguments que pour la fonction fopen ( mode par défaut à 'w'), connection indique la connexion à suivre. Par défaut, c'est la dernière ouverte.

pg_trace retourne TRUE si pathname a pu être ouvert en écriture, etFALSEsinon.

Voir aussi fopen et pg_untrace .

## 6.31.72 pg_tty() : Retourne le nom de tty

string **pg_tty** ( resource connection )

pg_tty retourne le nom de tty de la connexion associée à connection .

## 6.31.73 pg_unescape_bytea() : Supprime le protection d'une chaîne de type bytea

string **pg_unescape_bytea** ( string data )

pg_unescape_bytea supprime la proctection des caractères de type bytea, mis dans la chaîne data par la fonction pg_unescape_bytea . La chaîne originale est ainsi retournée.

| Note |
| --- |
| Lorsque vous utilisez une commande SELECT avec des données de type bytea, PostgreSQL retourne des valeurs octales, préfixées avec des anti−slash \ (e.g. \032). Les utilisateurs doivent effectuer la conversion en format binaire eux−même.<br><br>pg_escape_bytea requiert PostgreSQL 7.2 ou plus récent. Avec PostgreSQL 7.2.0 et 7.2.1, les données de type bytea doivent être transtypée lorsque vous activez le support des chaînes de caractères multi−octets. i.e. INSERT INTO test_table (image) VALUES ('$image_escaped'::bytea); . PostgreSQL 7.2.2 ou plus récent ne requiert pas cette manipulation. Toutefois, si le client et le serveur n'utilisent pas le même jeux de caractères, il peut arriver des erreurs. Il faut alors forcer le transtypage manuellement. |

Voir aussi pg_escape_bytea et pg_escape_string

## 6.31.74 pg_untrace() : Termine le suivi d'une connexion PostgreSQL

bool **pg_untrace** ( *resource connection* )

pg_untrace termine le suivi d'une connexion PostgreSQL, initiée avec pg_trace . connection indique la connexion à suivre. Par défaut, c'est la dernière ouverte.

pg_untrace retourne toujours TRUE .

Voir aussi pg_trace .

## 6.31.75 pg_update() : Modifie les lignes d'une table

mixed **pg_update** ( resource connection , string table_name , array data , array condition , *int options* )

pg_update modifie les lignes de la table table_name , qui vérifient la condition condition , et leur donne la valeur de data . Si options est spécifié, pg_convert est appliqué à data avec les options spécifiées.

Exemple avec pg_update

```
<?php
    $db = pg_connect ('dbname=foo');
   $data = array('field1'=>'AA', 'field2'=>'BB');
   // C'est correct, car $_POST est convertit automatiquement
   $res = pg_update($db, 'post_log', $_POST, $data);
   if ($res) {
       echo "Les données ont été modifiées : $res\n";
   }
   else {
       echo "Problème dans les données utilisateur\n";
   }
?>
```

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Voir aussi pg_convert .

# 6.32 Exécution de programmes externes

## 6.32.1 Introduction

Ces fonctions fournissent la possibilité de passer directement des commandes au système, mais aussi de protéger le système des commandes passées.

## 6.32.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.32.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.32.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.32.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.32.6 Constantes prédefinies

Cette extension ne définit aucune constante.

## 6.32.7 Voir aussi

Ces fonctions sont complétées par l'opérateur guillemets obliques . De plus, lorsque le safe mode est activé, vous devez configurer l'option safe_mode_exec_dir .


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Voir aussi
- escapeshellarg
- escapeshellcmd
- exec
- passthru
- proc_close
- proc_get_status

- proc_nice
- proc_open
- proc_terminate
- shell_exec
- system

## 6.32.9 escapeshellcmd() : Protège les caractères spéciaux du Shell

string **escapeshellcmd** ( string command )

escapeshellcmd échappe tous les caractères de la chaîne command qui pourraient avoir une signification spéciale dans une commande shell. Cette fonction permet de s'assurer que la commande sera correctement passée à l'exécuteur de commande shell exec et system , ou encore à guillemets obliques . Généralement, cette fonction est utilisée comme ceci :

Exemple avec escapeshellcmd

```
<?php
 system(escapeshellcmd($cmd));
?>
```

Voir aussi escapeshellarg , exec , popen et system .

## 6.32.10 exec() : Exécute un programme externe

string **exec** ( string command , *array output* , *int return_var* )

exec execute la commande command , mais ne renvoie rien comme retour, hormis la dernière ligne du résultat de la commande. Pour exécuter une commande et obtenir le résultat sans aucun traitement, il faut utiliser la fonction passthru .

Si l'argument output est présent, alors ce tableau sera rempli par les lignes retournées par la commande. Il faut noter que si ce tableau contient des éléments, exec ajoutera les nouvelles lignes à la fin du tableau. Si vous ne voulez pas que les nouveaux éléments soient concaténés, utilisez la fonction unset avec ce tableau avant de le passer à exec .

Si l'argument return_var est présent en plus du tableau output , alors le statut de retour d'exécution sera inscrit dans cette variable.

Notez que si vous allez fournir des commandes qui proviennent d'un utilisateur, il est avisé d'utiliser la fonction escapeshellcmd pour s'assurer que l'utilisateur n'essaie pas de profiter des caractères spéciaux pour tromper le système.

Note

Si vous démarrez un programme en utilisant cette fonction, et que vous voulez qu'il continue de fonctionner en tâche de fond, vous devez vous assurez que le résultat de ce programme est redirigé vers un fichier ou une autre méthode d'archivage, car sinon, PHP va attendre la fin de l'exécution du programme.

Voir aussi system , passthru , popen et escapeshellcmd .

## 6.32.11 passthru() : Exécute un programme externe et affiche le résultat brut.

void **passthru** ( string command , *int return_var* )

La fonction passthru est similaire à la fonction exec car les deux exécutent la commande command . Si l'argument return_var est présent, le code de statut de réponse UNIX y sera placé. Cette fonction doit être utilisée de préférence aux commandes exec ou system lorsque le résultat attendu est de type binaire, et doit être passé tel quel à un navigateur. Une utilisation classique de cette fonction est l'exécution de l'utilitaire pbmplus qui peut retourner une image. En fixant le résultat du contenu (Content−Type) à "image/gif" puis en appelant pbmplus pour obtenir une image gif, vous pouvez créer des scripts PHP qui retourne des images.

Voir aussi exec , system , popen et escapeshellcmd .

## 6.32.12 proc_close() : Ferme un processus ouvert par proc_open() et retourne le code de sortie du processus.

int **proc_close** ( resource process )

proc_close est similaire à pclose hormis le fait qu'elle fonctionne avec les processus ouverts par proc_open . proc_close attend que le processus se termine, puis retourne son code de sortie. Si vous avez des pipes ouverts avec ce processus, il faut les fermer avec fclose avant d'appler cette fonction pour éviter des verrouillages : le processus peut ne pas pouvoir sortir tant que les pipes sont ouverts.

## 6.32.13 proc_get_status() : Lit les informations concernant un processus

array **proc_get_status** ( resource process )

proc_get_status lit les données concernant le processus process , créé avec la fonction proc_open . Les informations collectés sont retournées dans un tableau, avec les éléments suivants :

| élément | type | description |
|---|---|---|
| command | string | |
| pid | int | identifiant du processus |
| running | bool | TRUE si le processus fonctionne toujours, et FALSE si il est terminé |
| signaled | bool | Non documenté |
| stopper | bool | Non documenté |
| exitcode | int | Le code retourné par le processu (uniquement si l'élément running vaut FALSE ) |
| termsig | int | Non documenté |
| stopsig | int | Non documenté |

Voir aussi proc_open .

## 6.32.14 proc_nice() : Change la priorité d'exécution du processus courant

bool **proc_nice** ( int priority )

proc_nice modifie la priorité d'exécution du processus courant en lui donnant la nouvelle valeur de priority . Si une erreur survient, comme par exemple l'impossibilité pour l'utilisateur de changer la priorité, une erreur de niveau E_WARNING est générée et FALSE est retournée. Sinon, TRUE est retournée.

| Note |
| --- |
| proc_nice n'est disponible que sur les systèmes qui disposent de capacités NICE. NICE est compatible avec : SVr4, SVID EXT, AT&T, X/OPEN, BSD 4.3. Par exemple, proc_nice n'est pas disponible sous Windows. |

Voir aussi proc_open et proc_terminate .

## 6.32.15 proc_open() : Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties

resource **proc_open** ( string cmd , array descriptorspec , array pipes )

proc_open est similaire à popen mais fournie un plus grand degré de contrôle sur l'exécution du programme. cmd est la commande qui doit être exécutée par le shell. descriptorspec est un tableau indexé, dont les clés représentent le numéro de descripteur, et la valeur représente la méthode avec laquelle PHP va passer ce descripteur au processus fils. pipes recevra un tableau indexé de pointeurs de fichiers, qui correspondent aux sorties dans PHP de tous les pipes qui sont créés. La valeur retournée est une ressource, qui représente le processus. Il faudra la détuire avec la fonction proc_close lorsque vous aurez fini.

Exemple avec proc_open

```
 <?php
$descriptorspec = array(
   0 => array("pipe", "r"),  // stdin est un pipe où le processus va lire
   1 => array("pipe", "w"),  // stdout est un pipe où le processus va écrire
   2 => array("file", "/tmp/error-output.txt", "a"), // stderr est un fichier
);
$process = proc_open("php", $descriptorspec, $pipes);
if (is_resource($process)) {
    // $pipes ressemble à :
     // 0 => fichier accessible en écriture, connecté à l'entrée standard du prcessus fils
    // 1 => fichier accessible en lecture, connecté à la sortie standard du prcessus fils
    // Toute erreur sera ajoutée au fichier /tmp/error-output.txt

    fwrite($pipes[0], "<?php echo \"Bonjour le monde!\"; ?>");
    fclose($pipes[0]);

    while(!feof($pipes[1])) {
        echo fgets($pipes[1], 1024);
    }
    fclose($pipes[1]);
    // Il est important que vous fermiez les pipes avant d'appeler
    // proc_close() afin d'éviter un verrouillage.
    $return_value = proc_close($process);

    echo "La commande a retournée $return_value\n";
}
?>
```

Les numéros de descripteurs de fichiers de descriptorspec ne sont pas limitées à 0, 1 et 2 : vous pouvez spécifier n'importe quel numéro de descripteur valide, et il sera passé au processus fils. Cela permettra à votre script d'inter opérer avec d'autres scripts, et d'être exécuté comme "co-processus". En particulier, c'est très pratique pour passer des mots de passes à des programmes comme PGP, GPG et openssl, avec une méthode très protégée. C'est aussi pratique pour lire des informations de statut fournies par ces programmes, sur des descripteurs auxiliaires.

| Note |
| --- |
| Compatibilité Windows : les descripteurs au-dela de 2 (stderr) sont accessibles au processus fils, sous la forme de pointeurs héritables, mais comme l'architecture Windows n'associe pas de |

nombre aux descripteurs de bas niveau, le processus fils n'a (actuellement) aucun moyen d'y accéder. Stdin, stdout et stderr fonctionnent comme d'habitude.

| Note |
| --- |
| Cette fonction a été introduite en PHP 4.3.0. |

| Note |
| --- |
| Si vous n'avez besoin que d'un processus uni−directionnel, popen sera plus pratique, car plus simple à utiliser. |

Voir aussi stream_select , exec , system , passthru , popen et escapeshellcmd .

## 6.32.16 proc_terminate() : Termine un processus ouvert par proc_open()

int **proc_terminate** ( resource process , *int signal* )

proc_terminate envoie un signal au processus process (crée par proc_open ) pour lui indiquer qu'il doit se terminer. proc_terminate se termine immédiatement après, et n'attend pas l'arrêt réel du processus.

Le paramètre optionnel signal n'est utile que sur les plate−formes POSIX : vous pouvez alors spécifier un signal à envoyer au processus, en utilisant l'appel système kill(2) . La valeur par défaut est alors SIGTERM .

proc_terminate vous permet de conclure un processus, et de continuer les autres tâches. Vous pouvez tester la présence de votre processus en utilisant la fonction proc_get_status .

Voir aussi proc_open , proc_close et proc_get_status .

## 6.32.17 shell_exec() : Exécute une commande via le shell et retourne le résultat sous forme de chaîne.

string **shell_exec** ( string cmd )

shell_exec exécute la commande cmd via le shell et retourne le résultat sous forme de chaîne.

shell_exec est identique aux opérateurs de backtick .

## 6.32.18 system() : Exécute un programme externe et affiche le résultat.

string **system** ( string command , *int return_var* )

system est la version PHP de la fonction C qui exécute la commande command et retourne le résultat. Si une variable est fournie comme second argument, alors le code de statut de la commande y sera affecté.

Notez que si vous allez fournir des commandes qui proviennent d'un utilisateur, il est avisé d'utiliser la fonction escapeshellcmd pour s'assurer que l'utilisateur n'essaie pas de profiter des caractères spéciaux pour tromper le système.

system essaie automatiquement de vider les tampons du serveur web après chaque ligne de résultat PHP, lorsque ce dernier fonctionne comme un module.

system retourne la dernière ligne du retour, en cas de succès, et FALSE en cas d'échec.

Si vous devez exécuter une commande et récupérer tout le résultat sans aucune intervention, utilisez la fonction passthru .

Voir aussi exec , passthru , popen et escapeshellcmd .

# 6.33 Expressions régulières compatibles Perl

## 6.33.1 Introduction

La syntaxe des masques utilisés dans ces fonctions ressemble fort à celle de Perl. Les expressions seront entourées de délimiteurs, slash (/), par exemple. N'importe quel caractère peut servir de délimiteur, tant qu'il n'est pas alpha−numérique ou n'est pas un antislash (\). Si un délimiteur doit être utilisé dans l'expression, il faudra l'échapper avec un antislash. Depuis PHP 4.0.4, vous pouvez utiliser les délimiteurs (), {}, [], et <>, comme en Perl.

Le délimiteur final peut être suivi d'options qui affecteront la recherche. Voir aussi options de recherche .

## 6.33.2 Pré−requis

Le support des expressions régulières est assuré par la librairie PCRE, qui est open source, et écrite par Philip Hazel. Elle est soumise au copyright de l'University of Cambridge, Angleterre. Elle est disponible à ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/ .

## 6.33.3 Installation

A partir de PHP 4.2.0, ces fonctions activées par défaut. Pour les anciennes versions, vous devez configurer et compiler PHP avec l'option −−with−pcre−regex[=DIR] afin de pouvoir les utiliser. Vous pouvez aussi désactiver le support des expressions régulières avec l'option −−without−pcre−regex .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

## 6.33.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.33.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.33.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

| Constante | Description |
|---|---|
| PREG_PATTERN_ORDER | Ordonne les résultats de façon à ce que $matches[0] contienne les résultats qui correspondent au masque entier, $matches[1] contienne les résultats qui correspondent à la première parenthèse capturante, $matches[2] contienne les résultats qui correspondent à la deuxième parenthèse capturante, etc... Cette constante est utilisée avec preg_match_all . |
| PREG_SET_ORDER | Les résultats sont classés de telle façon que $matches[0] contient la première série de résultat, $matches[1] contient la deuxième série de résultat, etc... Cette constante est utilisée avec preg_match_all . |

| | |
|---|---|
| PREG_OFFSET_CAPTURE | Voir la description de PREG_SPLIT_OFFSET_CAPTURE . Cette constante est utilisée depuis PHP 4.3.0 . |
| PREG_SPLIT_NO_EMPTY | Si cette option est activée, seules les sous−chaînes non vides seront retournées par preg_split . |
| PREG_SPLIT_DELIM_CAPTURE | Si cette option est activée, les expressions entre parenthèses entre les délimiteurs de masques seront aussi capturées et retournées. Cette option a été ajoutée en PHP 4.0.5. Cette constante est utilisée avec preg_split . |
| PREG_SPLIT_OFFSET_CAPTURE | Si cette constante est utilisée avec preg_split , l'offset de début de résultat sera retourné, en plus de la chaîne résultat. Notez que cela change la nature du résultat retourné en un tableau contenant une chaîne à l'offset 0 et une chaîne contenant un offset à l'offset 1 . Cette option est disponible depuis PHP 4.3.0. |

## 6.33.7 Exemples

| Exemples de masques valides |
|---|
| • /<\/\w+>/<br>• \|(\d{3})−\d+\|Sm<br>• /^(?i)php[34]/<br>• {^\s+(\s+)?$} |

| Exemples de masques invalides |
|---|
| • /href='(.*)' − délimiteur final manquant<br>• /\w+\s*\w+/J − option 'J' inconnue<br>• 1−\d3−\d3−\d4\| − délimiteur initial manquant |

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- options de recherche
- syntaxe des masques
- preg_grep
- preg_match_all
- preg_match
- preg_quote
- preg_replace_callback
- preg_replace
- preg_split

## 6.33.9 syntaxe des masques() : Fonctionnement des expressions régulières.

La bibliothèque PCRE est un ensemble de fonctions qui implémentent la recherche par expressions régulières, en utilisant la même syntaxe et la même sémantique que le Perl 5, avec quelques nuances (voir ci−dessous). L'implémentation actuelle est celle de Perl 5.005.

## 6.33.10 preg_grep() : Retourne un tableau avec les résultat de la recherche

array **preg_grep** ( string pattern , array input )

preg_grep retourne un tableau qui contient les éléments de input qui satisfont le masque pattern .

Depuis PHP 4.0.4, le tableau retourné par preg_grep est indexé en utilisant les clés issues du tableau input . Si ces clés sont inutiles, utilisez la fonction array_values sur le tableau retourné par

preg_grep pour obtenir le comportement traditionnel.

```
Exemple avec preg_grep
<?php
// recherche les nombres à virgule flottante
preg_grep("/^(\d+)?\.\d+$/", $array);
?>
```

## 6.33.11 preg_match_all() : Expression régulière globale

int **preg_match_all** ( string pattern , string subject , array matches , *int flags* )

preg_match_all analyse subject pour trouver l'expression pattern et met les résultats dans matches , dans l'ordre spécifié par order .

Après avoir trouvé un premier résultat, la recherche continue jusqu'à la fin de la chaîne.

order peut prendre une des deux valeurs suivantes (notez bien qu'il est incohérent d'utiliser PREG_PATTERN_ORDER avec PREG_SET_ORDER ) :

*PREG_PATTERN_ORDER*

L'ordre est tel que $matches[0] est un tableau qui contient les résultats qui satisfont le masque complet, $matches[1] est un tableau qui contient les résultats qui satisfont la première parenthèse capturante, etc..

```
Exemple avec preg_match_all et PREG_PATTERN_ORDER
<?php
 preg_match_all("|<[^>]+>(.*)</[^>]+>|U",
                "<b>exemple : </b><div align=left>un test</div>",
                $out,
                PREG_PATTERN_ORDER);
 print $out[0][0].", ".$out[0][1]."\n";
 print $out[1][0].", ".$out[1][1]."\n";
?>
```

Cet exemple va afficher :

```
Resultat avec PREG_PATTERN_ORDER

 <b>exemple : </b>, <div align=left>un test</div>

exemple : , un test
```

Ainsi, $out[0] est un tableau qui contient les résultats qui satisfont le masque complet, et $out[1] est un tableau qui contient les balises entre > et <.

*PREG_SET_ORDER*

Les résultats sont classés de telle façon que $matches[0] contient la première série de résultat, $matches[1] contient la deuxième série de résultat, etc...

```
Exemple avec preg_match_all et PREG_SET_ORDER
<?php
 preg_match_all("|<[^>]+>(.*)</[^>]+>|U",
                "<b>exemple : </b><div align=left>un test</div>",
                $out,
                PREG_SET_ORDER);
 print $out[0][0].", ".$out[0][1]."\n";
 print $out[1][0].", ".$out[1][1]."\n";
?>
```

Cet exemple va afficher :

```
Résultat avec PREG_SET_ORDER
<b>exemple : </b>, exemple :
<div align=left>un test</div>, un test
```

Dans ce cas, $matches[0] est la première série de résultat, et $matches[0][0] contient le texte qui satisfait le masque complet, $matches[0][1] contient le texte de la première parenthèse capturante, etc... De même, $matches[1] contient le texte qui satisfait le masque complet, etc...

*PREG_OFFSET_CAPTURE*
Si cette option est activée, toutes les sous chaînes qui satisfont le masque seront aussi identifiées par leur offset. Notez que cela modifie le format de la valeur retournée , puisque chaque élément de réponse devient un tableau contenant la sous−chaîne résultat, à l'index 0 dans la chaîne subject constant 1 . Cette option a été ajoutée en PHP 4.3.0 .

Si order est omis, PREG_PATTERN_ORDER est utilisé par défaut.

preg_match_all retourne le nombre de résultat qui satisfont le masque complet, ou FALSE en cas d'échec ou d'erreur.

Extraction de tous les numéros de téléphone d'un texte.

```
<?php
 preg_match_all("/\(?  (\d{3})?  \)?  (?(1)  [\−\s] ) \d{3}−\d{4}/x",
               "Appelez 555−1212 ou 1−800−555−1212", $phones);
?>
```

Recherche les couples de balises HTML (gourmand)

```
<?php
// Cet exemple utilise les références arrières (\\2).
// Elles indiquent à l'analyseur qu'il doit trouver quelquechose qu'il
// a déjà repéré un peu plus tôt
// le nombre 2 indique que c'est le deuxième jeu de parenthèses
// capturante qui doit être utilisé (ici, ([\w]+)).
//  L'antislash est nécessaire ici, car la chaîne est entre guillemets doubles

  $html = "<B>Texte en gras</B><a href=salut.html>clique moi</a>";

preg_match_all ("/(<([\w]+)[^>]*>)(.*)(<\/\\2>)/", $html, $matches);

for ($i=0; $i< count($matches[0]); $i++) {
  echo "Trouvé : ".$matches[0][$i]."\n";
  echo "partie 1 : ".$matches[1][$i]."\n";
  echo "partie 2 : ".$matches[3][$i]."\n";
  echo "partie 3 : ".$matches[4][$i]."\n\n";
}
?>
```

Cet exemple va produire :
Trouvé : <B>Texte en gras</B>
partie 1 : <B>
partie 2 : Texte en gras
partie 3 : </B>

Trouvé : <a href=salut.html>clique moi</a>
partie 1 : <a href=salut.html>
partie 2 : clique moi
partie 3 : </a>

Voir aussi preg_match , preg_replace et preg_split .

## 6.33.12 preg_match() : Expression régulière standard

int **preg_match** ( string pattern , string subject , *array matches* , *int flags* )

preg_match analyse subject pour trouver l'expression pattern .

Si matches est fourni, il sera rempli par les résultats de la recherche. $matches[0] contiendra le texte qui satisfait le masque complet, $matches[1] contiendra le texte qui satisfait la première parenthèse capturante, etc..

Le paramètre flags peut prendre l'une des valeurs suivantes :

*PREG_OFFSET_CAPTURE*
Si cette option est activée, toutes les sous chaînes qui satisfont le masque seront aussi identifiées par leur offset. Notez que cela modifie le format de la valeur retournée , puisque chaque élément de réponse devient un tableau contenant la sous−chaîne résultat, à l'index 0 dans la chaîne subject constant 1 . Cette option a été ajoutée en PHP 4.3.0 .

Le paramètre flags a été ajouté en PHP 4.3.0 .

preg_match retourne le nombre de fois où le masque pattern a été trouvé. Cela peut aller de 0 (pas de solution) à un 1 car preg_match s'arrête dès qu'elle a trouvé une première solution. preg_match_all , au contraire, va continuer jusqu'à la fin de la chaîne subject . preg_match retourne FALSE si une erreur survient.

Extraction d'un numéro de page d'une chaîne.

```
<?php
if (preg_match("/page\s+#(\d+)/i", "Aller à la page numéro 9.", $parts)) {
    echo "La page suivante est $parts[1]";
} else {
    echo "Page introuvable.";
}
?>
```

Trouve le mot "web"

```
<?php
// \b, dans le masque, indique une limite de mot, de façon à ce que le mot
// "web" uniquement soit repéré, et pas seulement des parties de mots comme
// dans "webbing" ou "cobweb"
if (preg_match ("/\bweb\b/i", "PHP est le meilleur langage de script du web.")) {
    echo "Un mot a été trouvé.";
} else {
    echo "Un mot n'a pas été trouvé.";
}
if (preg_match ("/\bweb\b/i", "PHP est le meilleur langage de script pour les webagency.")) {
     echo "Un mot a été trouvé.";
} else {
    echo "Un mot n'a pas été trouvé.";
}
?>
```

Lire un nom de domaine dans une URL

```
<?php
// repérer le nom de l'hôte dans l'URL
  preg_match("/^(http:\/\/)?([^\/]+)/i",
              "http://www.php.net/index.html", $matches);
  $host = $matches[2];
// repérer les deux derniers segments du nom de l'hôte
  preg_match("/[^\.\/]+\.[^\.\/]+$/",$host,$matches);
  echo "Le nom de domaine est : ".$matches[0]."\n";
?>
```

Cet exemple va afficher :
Le nom de domaine est : php.net

Voir aussi preg_match_all , preg_replace et preg_split .

## 6.33.13 preg_quote() : Echappement des caractères spéciaux des expressions régulières.

string **preg_quote** ( string str , *string delimiter* )

preg_quote ajoute un antislash devant tous les caractères de la chaîne str . Cela est très utile si vous avez une chaîne qui va servir de masque, mais qui est générée durant l'exécution.

Si l'argument optionnel delimiter est fourni, il sera aussi échappé. Ceci est pratique pour échapper le délimiteur requis par les fonctions PCRE. Le slash / est le délimiteur le plus répandu.

Les caractères spéciaux qui seront échappés : . \\ + * ? [ ^ ] $ ( ) { } = ! < > | :

```
Protège des caractères spéciaux
 <?php
  $keywords = "$40 pour un g3/400";
  $keywords = preg_quote ($keywords, "/");
  echo $keywords;
   // retourne \$40 pour un g3\/400
?>
```

```
Mise en italique d'un mot dans un texte
 <?php
// Dans cet exemple, preg_quote($word) sert à éviter que les astérisques
// prennent une valeur particulière dans l'exepression régulière.
$textbody = "Ce livre est *très* difficile à trouver.";
$word = "*très*";
$textbody = preg_replace ("/".preg_quote($word)."/",
                          "<B>".$word."</B>",
                          $textbody);
?>
```

## 6.33.14 preg_replace_callback() : Rechercher/remplacer avec une expression régulière et fonction de callback

mixed **preg_replace_callback** ( mixed pattern , callback callback , mixed subject , *int limit* )

Le comportement de preg_replace est presque identique à celui de preg_replace , hormis le fait qu'à la place du paramètre replacement , il faut spécifier une fonction de callback callback qui sera appelée, avec les éléments trouvés en arguments. Cette fonction retourne alors la chaîne de remplacement.

```
Exemple avec preg_replace_callback
 <?php
  // Ce texte était vrai en 2002
  // nous voulons le mettre a jour pour 2003
  $text = "Le premier avril est le 01/04/2002\n";
  $text.= "Le dernier noel était le 12/24/2001\n";

  // Fonction de callback
  function next_year($matches) {
    // comme d'habitude : $matches[0] représente la valeur totale
    // $matches[1] représente la première parenthèse capturante
    return $matches[1].($matches[2]+1);
  }

  echo preg_replace_callback(
              "|(\d{2}/\d{2}/)(\d{4})|",
               "next_year",
              $text);

  // retourne ceci :
   // Le premier avril est le 01/04/2003
  // Le dernier noel était le 12/24/2002
?>
```

Vous aurez souvent besoin de la fonction callback avec preg_replace_callback à un seul endroit. Dans ce cas, vous pouvez simplement utiliser la fonction create_function pour déclarer une fonction anonyme comme fonction de callback pour preg_replace_callback . En faisant cela, vous concentrez toutes les routines liées à votre remplacement à un seul endroit, et nous ne polluez par votre espace de noms de fonctions avec des fonctions à usage unique.

```
Exemples avec preg_replace_callback et create_function
 <?php
  // Un filtre de ligne de commande Unix pour convertir les lettres
  // en majuscule de début des paragraphe en minuscules

    $fp = fopen("php://stdin", "r") or die("Impossible de lire la ligne de commande");
    while (!feof($fp)) {
        $line = fgets($fp);
        $line = preg_replace_callback(
              '|<p>\s*\w|',
               create_function(
                // Les guillemets simples sont très importants ici
```

```
                // ou bien il faut protéger les caractères $ avec \$
                '$matches',
                'return strtolower($matches[0]);'
              ),
              $line
            );
        echo $line;
    }
    fclose($fp);
?>
```

Voir aussi preg_replace et create_function .

## 6.33.15 preg_replace() : Rechercher et remplacer par expression régulière standard

mixed **preg_replace** ( mixed pattern , mixed replacement , mixed subject , *int limit* )

preg_replace analyse subject pour trouver l'expression pattern et remplace les résultats par replacement .

replacement peut contenir des références de la forme \\ n ou, depuis PHP 4.0.4) $n . Cette dernière forme est recommandée. Ces références seront remplacées par le texte capturé par la n '−ième parenthèse capturante du masque. n peut prendre des valeurs de 0 à 99, et \\0 ou $0 , correspondent au texte de qui satisfait le masque complet. Les parenthèses ouvrantes sont comptées de gauche à droite (en commençant à 1) pour déterminer le numéro de parenthèse capturante.

Utilisation des références arrières avec des littéraux numériques

```
 <?php
$string = "Avril 05, 2003";
$pattern = "/(\w+) (\d+), (\d+)/i";
$replacement = "\${1}1,\$3";
print preg_replace($pattern, $replacement, $string);

/* Output
   ======

Avril1,2003

 */
?>
```

Si la recherche n'aboutit à aucun résultat, subject sera inchangé.

Tous les paramètres de preg_replace peuvent être des tableaux.

Note

Lorsque vous utilisez des tableaux avec les paramètres pattern et replacement , les clés sont traitées dans l'ordre dans lequel elles apparaissent dans le tableau. Ce **n'est pas forcément** la même chose que l'ordre des index numérique. Si vous utilisez des index pour identifier quel pattern doit être replacée par quel replacement , il est recommandé de faire un tri ksort sur chaque tableau avant de faire appel à preg_replace .

Utilisation de tableaux indexé avec preg_replace

```
 <?php
$string = "The quick brown fox jumped over the lazy dog.";

$patterns[0] = "/quick/";
$patterns[1] = "/brown/";
$patterns[2] = "/fox/";

$replacements[2] = "bear";
$replacements[1] = "black";
$replacements[0] = "slow";

print preg_replace($patterns, $replacements, $string);
```

```
/* Output
   ======

The bear black slow jumped over the lazy dog.

*/

/* By ksorting patterns and replacements,
   we should get what we wanted. */

ksort($patterns);
ksort($replacements);

print preg_replace($patterns, $replacements, $string);

/* Output
   ======

The slow black bear jumped over the lazy dog.

*/

?>
```

Si subject est un tableau, alors l'opération sera appliquée à chacun des éléments du tableau, et le tableau sera retourné.

Si pattern et replacement sont des tableaux, alors preg_replace prend une valeur de chaque tableau, et l'utilise pour faire la recherche et le remplacement. Si replacement à moins d'éléments que pattern , alors la chaîne vide est utilisé pour le reste des valeurs. Si pattern est un tableau, et que replacement est une chaîne, alors cette chaîne sera utilisée pour chaque valeur de pattern . Le contraire n'aurait pas de sens.

/e force preg_replace à traiter replacement comme du code PHP une fois que les substitutions adéquates ont été faites. Conseil :assurez−vous que replacement est un code PHP valide, car sinon, PHP trouvera une erreur d'analyse (parse error) dans cette ligne.

/F indique que le paramètre replacement doit être considéré comme un nom de fonction. Cette fonction sera appelée, avec un tableau contenant les éléments trouvés comme arguments. La fonction doit retourner la chaîne de remplacement. Cette option a été ajoutée en PHP 4.0.4.

Remplacement de plusieurs valeurs simultanément

```
 <?php
$patterns = array ("/(19|20)(\d{2})-(\d{1,2})-(\d{1,2})/",
                   "/^\s*{(\w+)}\s*=/");
$replace = array ("\\3/\\4/\\1\\2", "$\\1 =");
print preg_replace ($patterns, $replace, "{startDate} = 1999-5-27");
?>
```

Cet exemple va afficher :

```
 $startDate = 5/27/1999
```

Utilisation de l'option /e

```
 <?php
  preg_replace("/(<\/?)(\w+)([^>]*>/e", "'\\1'.strtoupper('\\2').'\\3'", $html_body);
?>
```

Cela va mettre en majuscule toutes les balises HTML du texte.

Conversion HTML en texte

```
 <?php
// $document contient un document HTML
// Ce script va effacer les balises HTML, les javascript
// et les espaces. Il remplace aussi quelques entités HTML
// courante en leur équivalent texte.
$search = array ("'<script[?>]*?>.*?</script>'si",  // Supprime le javascript
                 "'<[\/\!]*?[^<?>]*?>'si",  // Supprime les balises HTML
                 "'([\r\n])[\s]+'",  // Supprime les espaces
                 "'&(quot|#34);'i",  // Supprime les entites HTML
                 "'&(amp|#38);'i",
                 "'&(lt|#60);'i",
                 "'&(gt|#62);'i",
                 "'&(nbsp|#160);'i",
                 "'&(iexcl|#161);'i",
```

```
                "'&(cent|#162);'i",
                "'&(pound|#163);'i",
                "'&(copy|#169);'i",
                "'&#(\d+);'e");  // Evaluation comme PHP
$replace = array ("",
                  "",
                  "\\1",
                  "\"",
                  "&",
                  "<",
                  ?>",
                  " ",
                  chr(161),
                  chr(162),
                  chr(163),
                  chr(169),
                  "chr(\\1)");
$text = preg_replace ($search, $replace, $document);
?>
```

Note

Le paramètre limit a été ajouté à partir de PHP 4.0.1pl2.

Voir aussi preg_match , preg_match_all et preg_split .

## 6.33.16 preg_split() : Eclate une chaîne par expression régulière

array **preg_split** ( string pattern , string subject , *int limit* , *int flags* )

preg_split retourne un tableau contenant les sous−chaînes de subject , séparées par les chaînes qui vérifient pattern .

Si limit est spécifié, alors seules les limit premières sous−chaînes sont retournées et si limit vaut −1, cela signifie en fait "sans limite", ce qui est utile pour passer le paramètre flags .

flags peut être la combinaison des options suivantes (combinées avec l'opérateur | ):

*PREG_SPLIT_NO_EMPTY*
: Si cette option est activée, seules les sous−chaînes non vides seront retournées par preg_split .

*PREG_SPLIT_DELIM_CAPTURE*
: Si cette option est activée, les expressions entre parenthèses entre les délimiteurs de masques seront aussi capturées et retournées. Cette option a été ajoutée en PHP 4.0.5.

Eclatement d'une chaîne de recherche

```
 <?php
// scinde la phrase grâce aux virgules et espacements
// ce qui inclus les " ", \r, \t, \n et \f
$keywords = preg_split ("/[\s,]+/", "langage hypertexte, programmation");
?>
```

Scinder une chaîne en caractères

```
 <?php
$str = 'string';
$chars = preg_split('//', $str, −1, PREG_SPLIT_NO_EMPTY);
print_r($chars);
?>
```

Scinde une chaîne et capture les offsets

```
 <?php
  $str = 'langage hypertexte, programmation';
  $chars = preg_split('/ /', $str, −1, PREG_SPLIT_OFFSET_CAPTURE);
  print_r($chars);
?>
```

va afficher

```
Array
```

```
(
[0] => Array
(
[0] => langage
[1] => 0
)

[1] => Array
(
[0] => hypertexte,
[1] => 8
)

[2] => Array
(
[0] => programmation
[1] => 20
)

)
```

| Note |
| --- |
| Le paramètre flags a été ajouté en PHP Beta 3. |

Voir aussi spliti , split , implode , preg_match , preg_match_all et preg_replace .

# 6.34 Sessions

## 6.34.1 Introduction

Le support des sessions de PHP est un moyen de préserver des données entre plusieurs accès. Cela vous permet de créer des applications personnalisées, et d'augmenter l'attractivité de votre site.

Si vous êtes familier avec la gestion de session de la librairie PHPLIB, vous remarquerez que certains concepts sont similaires.

Chaque visiteur accédant à votre page web se voit assigné un identifiant unique, appelé 'identifiant de session'. Il peut être stocké soit dans un cookie, soit propagé dans l'URL.

Le support des sessions vous permet d'enregistrer un nombre illimité de variables qui doivent être préservées entre les requêtes. Lorsqu'un visiteur accède à votre site, PHP va vérifier automatiquement (si session.auto_start est activé1) ou sur demande (explicitement avec session_start ou implicitement avec session_register ) si il existe une session du même nom. Si c'est le cas, l'environnement précédemment sauvé sera recréé.

Toutes les variables sont sérialisées après l'exécution du script PHP. Les variables qui sont indéfinies sont marquées comme telles. Lors des accès ultérieurs, elles ne seront pas définies, jusqu'à ce que l'utilisateur le fasse.

| Note |
| --- |
| La gestion des sessions a été ajoutée en PHP 4.0. |

## 6.34.2 Sessions et sécurité

Lien externe : Session fixation

Utiliser les sessions ne signifie pas que les données de session ne pourront être vue que par un seul utilisateur. Il est important de garder cela en tête, lorsque vous stockez et affichez des données importantes. Lorsque vous stockez des données dans une session, il faut se demander quels seront les problèmes posés si quelqu'un d'autre accède à cette information, ou comment votre application est affectée si la session est en fait celle d'un autre.

Par exemple, si quelqu'un usurpe une session, il peut alors poster un message dans un forum sous une fausse identitée. Quelle est la gravité de ce problème? Ou bien, il peut accèder aux commandes d'un client, et même, modifier son panier d'achat. A priori, c'est moins problématique pour un fleuriste que pour un pharmacien.

Par conséquent, lorsque vous manipulez des données importantes, il faut exploiter d'autres méthodes pour décider si une session est valide ou pas. Les sessions ne fournissent pas une méthode fiable d'authentification.

Les sessions reposent sur un identifiant de session, ce qui signifie que quelqu'un peut voler cet identifiant, rien qu'en volant l'ID. Ce vol peut être rendu très difficile, comme par exemple en utilisant les cookies, mais en aucun cas cela sera impossible. Les sessions dépendent aussi de la discipline de l'utilisateur qui referme son navigateur à la fin de la session pour tout clore proprement. De plus, même les cookies de session peuvent être surveillés sur un réseau, ou bien notés par un proxy.

## 6.34.3 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.34.4 Installation

Le suport des sessions est activé par défaut. Si vous souhaitez exclure le support des sessions de PHP , vous devez utiliser l'option −−disable−session lors de l'exécution du script de configuration. Pour utiliser la mémoire vive pour le stockage des sessions, compilez PHP avec l'option −−with−mm[=DIR] .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

## 6.34.5 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| session.save_path | "/tmp" | PHP_INI_ALL |
| session.name | "PHPSESSID" | PHP_INI_ALL |
| session.save_handler | "files" | PHP_INI_ALL |
| session.auto_start | "0" | PHP_INI_ALL |
| session.gc_probability | "1" | PHP_INI_ALL |
| session.gc_maxlifetime | "1440" | PHP_INI_ALL |
| session.serialize_handler | "php" | PHP_INI_ALL |
| session.cookie_lifetime | "0" | PHP_INI_ALL |
| session.cookie_path | "/" | PHP_INI_ALL |
| session.cookie_domain | "" | PHP_INI_ALL |
| session.cookie_secure | "" | PHP_INI_ALL |
| session.use_cookies | "1" | PHP_INI_ALL |
| session.use_only_cookies | "0" | PHP_INI_ALL |
| session.referer_check | "" | PHP_INI_ALL |
| session.entropy_file | "" | PHP_INI_ALL |
| session.entropy_length | "0" | PHP_INI_ALL |
| session.cache_limiter | "nocache" | PHP_INI_ALL |
| session.cache_expire | "180" | PHP_INI_ALL |
| session.use_trans_sid | "0" | PHP_INI_SYSTEM\|PHP_INI_PERDIR |
| session.bug_compat_42 | "1" | PHP_INI_ALL |
| session.bug_compat_warn | "1" | PHP_INI_ALL |
| url_rewriter.tags | "a=href,area=href,frame=src,input=src,form=fakeentry" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Le système de sessions dispose d'un grand nombre de directives dans le fichier php.ini . En voici une présentation :

*session.save_handler string*
: Définit le nom du gestionnaire de session qui est utilisé pour stocker et relire les données. Par défaut, c'est le système intégré par fichiers : files .

*session.save_path string*
: Définit le chemin qui doit être passé au gestionnaire de sauvegarde. Si vous décidez de choisir le gestionnaire par défaut (par fichier), cet argument sera utilisé comme dossier de sauvegarde des sessions. Par défaut, il vaut /tmp . Si le niveau d'imbrication de session.save_path est supérieur à 2, le nettoyage des sessions obsolètes ne s'opérera pas.

| Attention |
|---|

Si vous laissez cette option configurée avec un dossier accessible en lecture à tout le monde, comme par exemple /tmp (par défaut), les autres utilisateurs pourront exploiter ces sessions en obtenant la liste de fichiers dans ce dossier.

Note

Les utilisateurs de Windows doivent changer cette valeur de variable pour que les fonctions de sessions de PHP fonctionnent. Indiquez un chemin de dossier valide, comme par exemple : c:/temp .

*session.name string*
: Spécifie le nom de la session, qui sera utilisé comme nom de cookie. Il ne doit contenir que des caractères alphanumérique. Par défaut, c'est PHPSESSID .

*session.auto_start integer*
: Spécifie si le module de session doit démarrer automatiquement au début de chaque script PHP. Par défaut, c'est 0 (désactivé).

*session.cookie_lifetime integer*
: Spécifie la durée de vie du cookie en secondes. La valeur de 0 siginifie : "Jusqu'à ce que le navigateur soit éteind". La valeur par défaut est : 0.

*session.serialize_handler string*
: Définit le nom du gestionnaire qui est utilisé pour linéariser/délinéariser les données. Actuellement, un format interne à PHP (nommé php ) et WDDX (nommé wddx ) sont supportés . WDDX est seulement disponible, si PHP a été compilé avec l'option WDDX . Par défaut, c'est php .

*session.gc_probability integer*
: Specifies the probability that the gc (garbage collection) routine is started on each request in percent. Defaults to 1 .

*session.gc_maxlifetime integer*
: Spécifie la durée de vie des données sur le serveur, en nombre de secondes. Après cette durée, les données seront considérées comme obsolètes, et supprimées.

Note

Si vous utilisez le gestionnaire de session par fichier, qui est fourni par défaut votre système doit garder la trace des date de dernier accès aux fichier (atime). La FAT de Windows ne le fait pas, alors vous faudra trouver un autre système pour gérer les sessions qui ont expirées.

*session.referer_check integer*
: Contient une sous−chaîne que vous souhaitez retrouver dans tous les en−têtes HTTP Referer. Si cet en−tête a été envoyé par le client, et que la sous−chaîne n'a pas été trouvé, l'identifiant de session sera considéré comme invalide. Par défaut, cette option est la chaÎne vide.

*session.entropy_file string*
: Est un chemin jusqu'à une source externe (un fichier), qui sera utilisée comme source additionnelle d'entropie pour la création de l'identifiant de session. Des exemples valides sont /dev/random et /dev/urandom , qui sont disponibles sur tous les systèmes Unix.

*session.entropy_length integer*
: Spécifie le nombre d'octets qui seront lus dans le fichier défini ci−dessus. Par défaut, il vaut 0 , c'est à dire inactif.

*session.use_cookies integer*
: Spécifie si le module utilisera les cookies pour stocker les données de session sur le client. Par défaut, il vaut 1 , c'est à dire actif.

*session.use_only_cookies integer*
: Spécifie si le module doit utiliser **seulement** les cookies pour stocker les identifiants de sessions du coté du navigateur. Par défaut, cette option vaut 0 (inactif, pour compatibilité ascendante). En l'activant, vous éviterez les attaques qui utilisent des identifiants de sessions dans les URL. Cette configuration a été ajoutée en PHP 4.3.0.

*session.cookie_path string*
Spécifie le chemin utilisé lors de la création du cookie. Par défaut, il vaut / .

*session.cookie_domain string*
Spécifie le domaine utilisé lors de la création du cookie. Par défaut, il ne vaut rien.

*session.cookie_secure bool*
Spécifie que les cookies ne doivent être émis que sur des connexion sécurisée. Par défaut, cette option est à off . Cette option a été ajoutée en PHP 4.0.4. Voir aussi session_get_cookie_params et session_set_cookie_params .

*session.cache_limiter string*
Spécifie le type de contrôle de cache utilisé pour les pages avec sessions. Les valeurs possibles sont : none, nocache, private, private_no_expire, public. Par défaut, il vaut nocache .

*session.cache_expire integer*
Spécifie la durée de vie des données de sessions, en minute. Cette option n'a aucune conséquence sur le contrôle de cache. Par défaut, il vaut 180 (3 heures).

*session.use_trans_sid integer*

| Note |
| --- |
| En PHP 4.1.2 ou plus ancien, cette option est activée en utilisant l'option de compilation −−enable−trans−sid . Depuis PHP 4.2.0, trans−sid est toujours compilée.<br><br>Le système de gestion des sessions par URL pose un risque supplémentaire de sécurité : un utilisateur peut envoyer son URL avec l'identifiant de session par email à un ami, ou bien le mettre dans ses signets. Cela diffusera alors l'identifiant de session. |

*session.bug_compat_42 boolean*
Les versions de PHP antérieures à la version 4.2.0 disposaient d'une fonctionnalité/bogue non documentée, qui vous permettait d'initialiser une variable de session dans le contexte global, même si register_globals était désactivé. PHP 4.3.0 et plus récent vous préviendra de l'utilisation de cette fonctionnalité si vous avez aussi activé session.bug_compat_warn .

*session.bug_compat_warn boolean*
Les versions de PHP antérieures à la version 4.2.0 disposaient d'une fonctionnalité/bogue non documentée, qui vous permettait d'initialier une variable de session dans le contexte global, même si register_globals était désactivé. PHP 4.3.0 et plus récent vous préviendra de l'utilisation de cette fonctionnalité si vous avez activé session.bug_compat_42 et session.bug_compat_warn .

*session.url_rewriter.tags string*
Spécifie quels sont les balises HTML qui doivent être réécrites si le support transparent du SID est activé. Par défaut, il vaut a=href,area=href,frame=src,input=src,form=fakeentry .

Les options track_vars et register_globals influencent le comportement des sessions, leur stockage et leur restauration.

| Note |
| --- |
| Depuis PHP 4.0.3, track_vars est toujours activé. |

## 6.34.6 Types de ressources

Cette extension ne définit aucune ressource.

## 6.34.7 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*SID (chaîne de caractères)*
: Constante contenant le nom de la session et l'identifiant en cours, sous la forme "name=ID" .

## 6.34.8 Exemples

Note

Depuis PHP 4.1.0, $_SESSION est disponible comme variable globale, au même titre que $_POST , $_GET , $_REQUEST , etc.. Contrairement à $HTTP_SESSION_VARS , $_SESSION est toujours globale. Par conséquent, vous n'avez pas besoin d'utiliser le mot réservé global avec $_SESSION . Notez que cette documentation a été modifiée pour utiliser $_SESSION . Vous pouvez toujours le remplacer par $HTTP_SESSION_VARS si vous préférez l'ancienne version.

Les clés du tableau $_SESSION sont sujettes aux mêmes limitations que les variables PHP habituelles, c'est à dire qu'elles ne peuvent pas commencer par un nombre, mais commencer par une lettre ou un souligné '_'. Pour plus de détails, reportez vous à la section sur les variables .

Si track_vars est activé et register_globals est désactivé, seuls les éléments du tableau global $_SESSION contiendront les variables enregistrées dans la session. Les variables de sessions relues seront uniquement disponibles dans $_SESSION .

L'utilisation de $_SESSION (ou $HTTP_SESSION_VARS avec PHP 4.0.6 et plus ancien) est recommandé pour une meilleure sécurité et un code plus facilement maintenable. Avec $_SESSION , il n'y a pas besoin d'utiliser les fonctions session_register , session_unregister et session_is_registered . Les variables de sessions sont accessibles comme toute autre variable.

Enregistrer une variable avec $_SESSION.

```
<?php
session_start();
// Utilisez $HTTP_SESSION_VARS avec PHP 4.0.6 ou plus ancien
if (!isset($_SESSION['count'])) {
    $_SESSION['compteur'] = 0;
} else {
    $_SESSION['compteur']++;
}
?>
```

Retirer une variable de session avec $_SESSION et register_globals inactif.

```
<?php
session_start();
// Utilisez $HTTP_SESSION_VARS avec PHP 4.0.6 ou plus ancien
unset($_SESSION['compteur']);
?>
```

Retirer une variable de session avec $_SESSION et register_globals activé, après l'avoir enregistré avec $_SESSION.

```
<?php
session_start();
// Avec PHP 4.3 et plus récent, vous pouvez simplement utiliser l'exemple précédent
session_unregister('compteur');
?>
```

Si register_globals est activé, alors toutes les variables globales peuvent être enregistrées comme variables de session, et toutes les variables de sessions seront reconstituées comme variables globales. Comme PHP doit savoir quels variables globales sont enregistrées comme variables de sessions, l'utilisateur doit enregistrer les variables avec session_register tandis que $HTTP_SESSION_VARS et $_SESSION ne nécessitent pas session_register .

Attention

Si vous utilisez $HTTP_SESSION_VARS et $_SESSION et désactivez register_globals , n'utilisez pas session_register , session_is_registered et session_unregister .

Si vous activez register_globals , session_unregister doit être utilisé, car les variables de session sont enregistrés comme variables globales lorsque les données de sessions sont relues. Désactiver register_globals est recommandé pour des raisons de sécurité et de performances.

Enregistrer une variable avec register_globals activé

```
<?php
if (!session_is_registered('compteur')) {
    session_register("compteur");
    $compteur = 0;
}
else {
    $compteur++;
}
?>
```

Si register_globals est activé, alors les variables globales et les entrées dans le tableau $_SESSION seront des références sur la même valeur pour les valeurs qui auront été enregistrées avant le démarrage de la session (donc, dans les page précédentes).

De plus, si vous enregistrez une nouvelle variable avec la fonction session_register , l'entrée dans l'environnement globale et $_SESSION ne fera pas de référence vers la même valeur jusqu'à la prochaine utilisation de session_start (ceci s'applique à PHP 4.2 est avant seulement). C'est à dire qu'une modification dans les variables globales ne seront pas répercutés dans les entrées de $_SESSION . Il est peu probable que cela ait un impact en pratique, et de plus, cela a été corrigé en PHP 4.3.

## 6.34.9 Passer l'identifiant de session (session ID)

Il y a deux méthodes de propagation de l'identifiant de session :

- Cookies
- Par URL

Le module de session supporte les deux méthodes. Les cookies sont optimaux, mais comme ils ne sont pas sûrs (tous les internautes ne les acceptent pas), ils ne sont pas fiables. La seconde méthode place l'identifiant de session directement dans les URL.

PHP est capable de faire cela de manière transparente, lorsqu'il est compilé avec l'option −−enable−trans−sid . Si vous activez cette option, les URL relatives seront modifiées pour contenir l'identifiant de session automatiquement. Alternativement, vous pouvez utiliser la constante SID , qui est définie, si le client n'a pas envoyé le cookie approprié. SID est soit de la forme session_name=session_id ou une chaîne vide.

Note

L'option arg_separator.output de php.ini vous permet de personnaliser le séparateur d'arguments.

L'exemple suivant vous montre comment enregistrer une variable et comment réaliser un lien correct avec une autre page, avec SID .

Compter le nombre de hits d'un utilisateur sur une page

```
<?php
if (!session_is_registered('compteur')) {
    session_register('compteur');
    $compteur = 1;
}
else {
```

```
    $compteur++;
}
?>

Bonjour visiteur, vous avez vu cette page <?php echo $compteur; ?> fois.<p>;

Pour continuer, <A HREF="nextpage.php?<?php echo SID?>">cliquez ici</A>.
```

<?php echo SID?> (ou bien <?=SID?> peut être utilisé si les short_open_tag sont actifs) est nécessaire pour préserver l'identifiant de session dans le cas où l'utilisateur a désactivé les cookies. <? echo SID?> n'est pas nécessaire, si −−enable−trans−sid a été activé à la compilation de PHP.

| Note |
| --- |
| Les URLS non relatives sont considérées comme externes au site, et ne recevront pas le SID, car c'est une fuite d'information vers un autre site (envoi d'informations importantes). |

## 6.34.10 Gestion personnalisée des sessions

Pour implémenter un stockage en base de données, ou toute autre méthode, vous aurez besoin de la fonction session_set_save_handler pour paramétrer vos propres fonctions de stockage.


- Introduction
- Sessions et sécurité
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- Passer l'identifiant de session (session ID)
- Gestion personnalisée des sessions
- session_cache_expire
- session_cache_limiter
- session_decode
- session_destroy
- session_encode
- session_get_cookie_params
- session_id
- session_is_registered
- session_module_name
- session_name
- session_regenerate_id
- session_register
- session_save_path
- session_set_cookie_params
- session_set_save_handler
- session_start
- session_unregister
- session_unset
- session_write_close


## 6.34.12 session_cache_limiter() : Lit et/ou modifie le limiteur de cache de session

string **session_cache_limiter** ( *string cache_limiter* )

session_cache_limiter retourne la configuration courante du limiteur de cache. Si cache_limiter est fourni, le limiteur de cache est reconfiguré avec la valeur de cache_limiter .

Le limiteur de cache contrôle les en−têtes HTTP envoyés au client. Certains en−têtes détermine les règles de mise en cache de la page sur le navigateur. En configurant ce limiteur à nocache , par exemple, le navigateur ne mettra pas la page dans son cache. La valeur de public , au contraire, permettra le cache. La valeur de private est aussi possible, et est légèrement plus restrictive que public .

En mode private , l'en−tête Expire envoyé au client peut poser des problèmes à certains navigateurs, comme notamment Mozilla . Vous pouvez éviter ce problème avec le mode private_no_expire . L'en−tête Expire n'est jamais envoyé au navigateur pour ce mode.

| Note |
| --- |
| private_no_expire a été ajouté en PHP 4.2.0. |

Le limiteur de cache est remis à la valeur par défaut de session.cache_limiter à chaque démarrage de script PHP . Donc, vous devrez appeler session_cache_limiter à chaque page, et avant session_start .

Exemple avec session_cache_limiter

```
<?php

/* configure le limiteur de cache à 'private' */

session_cache_limiter('private');
$cache_limiter = session_cache_limiter();

echo "Le limiteur de cache vaut maintenant $cache_limiter\n";
?>
```

Voir aussi la directive de configuration session.cache_limiter .

## 6.34.13 session_decode() : Décode les données de session

bool **session_decode** ( string data )

session_decode décode les données de sessions fournies dans le paramètre data , et défini les variables ainsi contenues dans la session.

Voir aussi session_encode

## 6.34.14 session_destroy() : Détruit une session

bool **session_destroy** ( void )

session_destroy détruit toutes les données associées avec la session courante. Cette fonction ne détruit pas les variables globales associées avec la session, de même, elle ne détruit pas le cookie de session.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Destruction d'une session avec session_destroy

```
<?php
```

```
// Initialisation de la session.
// Si vous utilisez un autre nom
 // session_name("something")
session_start();
// Détruit toutes les variables de session
session_unset();
// Finalement, détruit la session
session_destroy();

?>
```

Destruction d'une session avec $_SESSION

```
 <?php

// Initialisation de la session.
// Si vous utilisez un autre nom
 // session_name("something")
session_start();
// Détruit toutes les variables de session
$_SESSION = array();
// Finalement, détruit la session
session_destroy();

?>
```

## 6.34.15 session_encode() : Encode les données de session

string **session_encode** ( void )

session_encode retourne une chaîne contenant les variables de la session courante encodées et prêtes au stockage.

Voir aussi session_decode

## 6.34.16 session_get_cookie_params() : Lit la configuration du cookie de session

array **session_get_cookie_params** ( void )

session_get_cookie_params retourne la configuration courante du cookie de session. Cette fonction retourne un tableau, qui contient les éléments suivants :

- "lifetime" − Durée de vie du cookie.
- "path" − Le chemin où les informations sont stockées.
- "domain" − Le domaine du cookie.
- "secure" − Le cookie ne doit être envoyé que sur des connexions sécurisées (cet élément a été ajouté en PHP 4.0.4).

Voir aussi les directives de configuration session.cookie_lifetime , session.cookie_path , session.cookie_domain , session.cookie_secure et session_set_cookie_params .

## 6.34.17 session_id() : Let et/ou modifie l'identifiant courant de session

string **session_id** ( *string id* )

session_id retourne l'identifiant courant de session.

Si id est fourni, il remplacera l'identifiant courant de session. session_id doit alors être appelé avant session_start . Suivant le gestionnaire de session que vous utilisez, tous les caractères ne seront

pas acceptés dans cette valeur. Par exemple, le gestionnaire de session par défaut, basé sur les fichiers, n'accepte que les lettres et chiffres!

La constante SID peut alors être utilisée pour lire le nom de la session courante, et l'identifiant de session à fournir dans les URL. Notez que SID est uniquement défini si le client n'a pas envoyé le bon cookie. Voyez aussi Gestion de session .

Voir aussi session_start , session_set_save_handler et session.save_handler .

## 6.34.18 session_is_registered() : Vérifie si une variable est enregistrée dans la session

bool **session_is_registered** ( string name )

session_is_registered retourne TRUE si la variable de nom name fait partie de la session courante.

| Note |
| --- |
| Si $_SESSION (ou $HTTP_SESSION_VARS en PHP 4.0.6 et plus ancien) est utilisé, utilisez la fonction isset pour vérifier la présence de la variable dans $_SESSION . |

| Attention |
| --- |
| Si vous utilisez $_SESSION (ou $HTTP_SESSION_VARS ), n'utilisez pas session_register , session_is_registered et session_unregister . |

## 6.34.19 session_module_name() : Lit et/ou modifie le module de session courant

string **session_module_name** ( *string module* )

session_module_name retourne le nom du module courant de session. Si module est fourni, cette valeur sera alors utilisée, et remplacera la valeur courante.

## 6.34.20 session_name() : Lit et/ou modifie le nom de la session

string **session_name** ( *string name* )

session_name retourne le nom de la session courante. Si name est fourni, le nom de la session courante sera remplacé par cette valeur.

Le nom de session est utilisé comme nom de cookie ou nom de variable dans les URL. Il ne doit contenir que des caractère alpha−numériques; il doit être court et descriptif (surtout pour les utilisateurs ayant activé l'alerte cookie). Le nom de session est toujours remis à sa valeur par défaut ( session.name , dans php.ini ). Il faut donc utiliser session_name au début de chaque script, et notamment avant session_start ou session_register .

Exemple avec session_name

```
<?php

/* choisi le nom de session  : MaSession */

$previous_name = session_name("MaSession");

echo "L'ancien nom de la session était $previous_name\n";
```

```
?>
```

Voir aussi la directive de configuration session.name .

## 6.34.21 session_regenerate_id() : Remplace l'identifiant de session courant par un nouveau

bool **session_regenerate_id** ( )

session_regenerate_id va remplacer l'identifiant de session courant par un nouveau, généré automatiquement, tout en conservant les valeurs de session.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Exemple avec session_regenerate_id

```
<?php
session_start();

print session_id();

session_regenerate_id();

// Ce n'est plus pareil
print session_id();

print_r($_SESSION);
?>
```

Note

Depuis PHP 4.3.3, si les cookies de session sont activés, l'utilisation de session_regenerate_id va aussi envoyer un cookie de session avec le nouvel identifiant.

Voir aussi session_id , session_start et session_name .

## 6.34.22 session_register() : Enregistre une variable dans une session

bool **session_register** ( mixed name , *mixed* ... )

session_register enregistre toutes les variables de nom name dans la session courante. Le nombre de variables enregistré est libre. Les noms peuvent être passé comme des chaînes, ou comme des tableaux contenant des chaînes ou des tableaux. Pour chaque nom, session_register place la variable dans la session courante, pour la sauvegarde de fin de script.

Attention

Si vous voulez que votre script fonctionne indépendamment de la configuration de la directive register_globals , vous devez utiliser la variable $_SESSION . Tous les éléments de $_SESSION sont automatiquement enregistrée. Si votre script utilise session_register , il ne fonctionnera pas dans les environnements où register_globals est désactivée.

Note

register_globals : note importante

Depuis PHP 4.2.0, la valeur par défaut de la directive de configuration PHP register_globals vaut **off** . La communauté PHP vous recommande de ne pas dépendre de cette directive, mais de trouver d'autres moyens pour accéder aux données, tels que les superglobals .

Attention

Cette fonction enregistre une variable **global** . Si vous enregistrer une variable globale dans une fonction, vous devez vous assurer de la rendre globale , avec global , ou bien le tableau $GLOBALS[] , ou bien utiliser un des tableaux de session ci−dessous.

Attention

Si vous utilisez $_SESSION (ou $HTTP_SESSION_VARS ), n'utilisez pas session_register , session_is_registered et session_unregister .

session_register retourne TRUE si toutes les variables ont été enregistrées correctement dans la session.

Si session_start n'a pas été appelé avant cette fonction, un appel implicite à session_start sans aucun paramètre ne sera fait.

Vous pouvez aussi créer une variable de session, simplement en ajoutant l'index approprié dans la variable $_SESSION ou $HTTP_SESSION_VARS .

Exemple avec session_register

```
<?php
// L'utilisation de session_register() est déconseillé
$barney = "Un gros dinosaure violet.";
session_register("barney");

// L'utilisationde $_SESSION est encouragé depuis PHP 4.1.0
$_SESSION["zim"] = "Un envahisseur d'un autre planète.";

// L'ancienne méthode avec $HTTP_SESSION_VARS
$HTTP_SESSION_VARS["spongebob"] = "Il a un caleçon carré!";
?>
```

Note

Il n'est actuellement pas possible d'enregistrer des ressources dans les sessions. Par exemple, vous ne pouvez pas créer de connexion à une base de données, et stocker la connexion dans une variable de session. Elle ne sera pas valide lors de la prochaine page. Les fonctions PHP qui retournent des ressources sont identifiées avec le type resource dans leurs définitions. Une liste de fonctions qui retournent des ressources sont disponible dans l'annexe types de ressources .

Si $_SESSION (ou $HTTP_SESSION_VARS pour les versions antérieures à PHP 4.0.6) est utilisé, assignez les variables à $_SESSION : i.e. $_SESSION['var'] = 'ABC';

Voir aussi session_is_registered et session_unregister .

## 6.34.23 session_save_path() : Lit et/ou modifie le chemin de sauvegarde des sessions

string **session_save_path** ( *string path* )

session_save_path retourne le chemin du dossier actuellement utilisé pour sauver les données de sessions. Si path est spécifié, le chemin du dossier sera modifié. session_save_path doit être appelé avant session_start .

Note

Sur certains systèmes d'exploitation, vous aurez à choisir un chemin vers un dossier capable de gérer un grand nombre de petits fichiers efficacement. Par exemple, sous Linux, reiserfs peut se rendre plus efficace que ext2fs.

Voir aussi la directive de configuration session.save_path .

## 6.34.24 session_set_cookie_params() : Modifie les paramètres du cookie de session

void **session_set_cookie_params** ( int lifetime , *string path* , *string domain* , *bool secure* )

session_set_cookie_params modifie les paramètres de configuration du cookie de session, qui a été configuré dans le fichier php.ini . L'effet de cette fonction ne dure que pendant l'exécution du script courant.

| Note |
| --- |
| Le paramètre secure a été ajouté en PHP 4.0.4. |

Voir aussi les directives de configuration session.cookie_lifetime , session.cookie_path , session.cookie_domain , session.cookie_secure et session_get_cookie_params .

## 6.34.25 session_set_save_handler() : Configure les fonctions de stockage de sessions

bool **session_set_save_handler** ( string open , string close , string read , string write , string destroy , string gc )

session_set_save_handler configure les fonctions de stockage de session, et permet de choisir des fonctions utilisateurs pour sauver et relire toutes les sessions. Cette fonction est très pratique lorsqu'il faut sauver les données de sessions en utilisant une autre technique que le système par fichier fourni par défaut : notamment, stockage en base de données, ou dans des démons tels que le SRM . Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

| Note |
| --- |
| Le gestionnaire d'écriture n'est pas exécuté tant que le flot de sortie est ouvert. Par conséquent, l'affichage de messages de debogage durant le gestionnaire d'écriture ne sera donc jamais visible depuis le navigateur. Si vous avez besoin de débogage, nous vous suggérons d'écrire vos messages dans un fichier. |

| Note |
| --- |
| Le gestionnaire d'écriture n'est pas appelé s'il n'y a pas de données dans la session. Cela s'applique aussi si des variables de sessions vides sont enregistrées. Cela diffère du comportement du système de gestion des sessions par fichier, qui crée un fichier vide dans ce cas. |

L'exemple suivant fournit un exemple de stockage des sessions semblable au système par défaut (par fichiers). Cet exemple peut facilement être adapté pour réaliser des sauvegardes en base de données, avec votre serveur préféré.

La fonction de lecture doit toujours retourner une chaîne, pour que le gestionnaire fonctionne comme prévu. Vous devez retourner une chaîne vide s'il n'y a pas de données à lire. Les valeurs retournées par les autres gestionnaires sont converties en booléen avant d'être retournées au script. TRUE pour la réussite, FALSE pour l'échec.

Exemple avec session_set_save_handler

```
<?php
function open ($save_path, $session_name) {
  global $sess_save_path, $sess_session_name;

  $sess_save_path = $save_path;
  $sess_session_name = $session_name;
  return(true);
}
```

```
function close() {
  return(true);
}

function read ($id) {
  global $sess_save_path, $sess_session_name;

  $sess_file = "$sess_save_path/sess_$id";
  if ($fp = @fopen($sess_file, "r")) {
    $sess_data = fread($fp, filesize($sess_file));
    return($sess_data);
  } else {
    return(""); // Doit retourner "" ici.
  }

}

function write ($id, $sess_data) {
  global $sess_save_path, $sess_session_name;

  $sess_file = "$sess_save_path/sess_$id";
  if ($fp = @fopen($sess_file, "w")) {
    return(fwrite($fp, $sess_data));
  } else {
    return(false);
  }

}

function destroy ($id) {
  global $sess_save_path, $sess_session_name;

  $sess_file = "$sess_save_path/sess_$id";
  return(@unlink($sess_file));
}

/*******************************************************
 * ATTENTION - Vous devrez implémenter un       *
 * collecteur de données obosolètes ici. *
 *******************************************************/
function gc ($maxlifetime) {
  return true;
}

session_set_save_handler ("open", "close", "read", "write", "destroy", "gc");

session_start();

// Utilisez vos sessions comme d'habitude
?>
```

Voir aussi la directive de configuration session.save_handler .

## 6.34.26 session_start() : Initialise une session

bool **session_start** ( void )

session_start crée une session (ou restaure la session trouvée sur le serveur, via l'identifiant de session passé).

Si vous voulez utiliser une session nommée, vous devez appeler session_name avant session_start .

session_start retourne toujours TRUE .

| Note |
| --- |
| Si vous utilisez des sessions basées sur les cookies, vous devez appeler session_start avant d'afficher quoi que ce soit au navigateur. |

Exemple de session : page1.php

```
 <?php
// page1.php

session_start();

echo 'Bienvenue à la page numéro 1';
```

```
$_SESSION['favcolor'] = 'vert';
$_SESSION['animal']   = 'chat';
$_SESSION['time']     = time();

// Fonctionne si le cookie a été accepté
echo '<br /><a href="page2.php">page 2</a>';

// Ou bien, en indiquant explicitement l'identfiant de session
echo '<br /><a href="page2.php?' . SID . '">page 2</a>';
?>
```

Après avoir vu la page page1.php avec un navigateur, la seconde page page2.php va magiquement afficher les données de sessions. Lisez la référence sur les sessions pour des informations sur la propagation des identifiants de sessions , et l'utilisation de la constante SID .

Exemple de session : page2.php

```
 <?php
// page2.php

session_start();

echo 'Bienvenue sur la page numéro 2<br />';

echo $_SESSION['favcolor']; // vert
echo $_SESSION['animal'];   // chat
echo date('Y m d H:i:s', $_SESSION['time']);

// Vous pourriez utiliser la constante SID ici, tout comme dans la page page1.php
echo '<br /><a href="page1.php">page 1</a>';
?>
```

Si vous avez utilisé une session nommée, vous devez appeler session_name avant d'appeler session_start .

session_start va configurer un gestionnaire interne d'affichage, pour assurer la réécriture des URL, lorsque le support transparent du SID est activé. Si vous utilisez ob_gzhandler ou ob_start , l'ordre des gestionnaires d'affichage est important pour un affichage correct. Par exemple, l'utilisateur doit enregistrer ob_gzhandler avant que la session ne commence.

Note

L'utilisation de zlib.output_compression est recommandée, à la place de ob_gzhandler .

Voir aussi session_id .

## 6.34.27 session_unregister() : Supprime une variable de la session

bool **session_unregister** ( string name )

session_unregister supprime la variable name de la session courante.

session_unregister retourne TRUE si la variable a pu être supprimée réellement.

Note

Si $_SESSION (ou $HTTP_SESSION_VARS pour PHP 4.0.6 et plus ancien) est utilisé, utilisez unset pour supprimer une variable de session.

Attention

session_unregister ne supprime pas la variable globale correspondant au nom de name : elle empêche simplement que name soit sauvée de la session à la fin du script. Vous devez appeler unset vous−même pour effacer la globale correspondante.

Attention

Si vous utilisez $_SESSION (ou $HTTP_SESSION_VARS ), n'utilisez pas session_register , session_is_registered et session_unregister .

## 6.34.28 session_unset() : Détruit toutes les variables d'une session

void **session_unset** ( void )

session_unset détruit toutes les variables de la session courante.

Note

Si vous utilisez $_SESSION (ou $HTTP_SESSION_VARS pour PHP 4.0.6 ou plus ancien), utilisez unset pour détruire une variable de session, i.e. unset($_SESSION['nomvariable']; .

Attention

NE détruisez PAS tout $_SESSION avec unset($_SESSION) car cela désactivera la possibilité d'enregistrer des variables de session à partir du tableau superglobal $_SESSION .

## 6.34.29 session_write_close() : Ecrit les données de session et ferme la session

void **session_write_close** ( void )

session_write_close termine la session courante, après avoir stocké les données.

Les données de session sont généralement stockées à la fin de votre script, automatiquement, sans besoin d'appeler explicitement session_write_close . Mais durant toute l'exécution du script, les données de session sont verrouillées en écriture, et un seul script peut opérer sur la session en même temps. Lorsque vous utilisez des frames avec des sessions, vous vous en rendrez compte en voyant les frames se rafraîchir les uns après les autres. Vous pouvez réduire le temps de calcul de ces pages en refermant la session aussitôt que possible, et ainsi, libérant les données pour les autres scripts.

# 6.35 Socket

## 6.35.1 Introduction

L'extension socket implémente une interface bas niveau avec les fonctions de communication par socket, basées sur les sockets BSD si populaires, et fournit la possibilité de fonctionner aussi bien sous forme de client que de serveur.

Pour une interface socket cliente plus générique, voyez fsockopen et pfsockopen .

Lorsque vous utilisez ces fonctions, il est important de vous rappeler que si de nombreuses fonctions ont le même nom que leur alter ego en langage C, elles ont souvent des déclarations différentes. Lisez attentivement les descriptions pour éviter des confusions.

Cela dit, ceux qui ne sont pas familiers avec la programmation par socket peuvent toujours trouver beaucoup de documentation dans les pages de manuel Unix appropriées, et il y a une grande quantité d'introductions en langage C sur le web, qui peuvent être facilement réutilisés, avec des adaptations mineures. UNIX Socket FAQ est un bon début.

| Attention |
| --- |
| Ce module est **EXPERIMENTAL** . Cela signifie que le comportement de ces fonctions, leurs noms et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez ce module à vos risques et périls. |

## 6.35.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.35.3 Installation

Les fonctions de socket décrites ici font partie d'une extension PHP qui doit être activée lors de la compilation en utilisant l'option −−enable−sockets , avec la commande configure .

| Note |
| --- |
| Le support des adresses IPv6 a été ajouté en PHP 5.0 . |

## 6.35.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.35.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.35.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*AF_UNIX (entier)*
*AF_INET (entier)*
*SOCK_STREAM (entier)*
*SOCK_DGRAM (entier)*
*SOCK_RAW (entier)*
*SOCK_SEQPACKET (entier)*
*SOCK_RDM (entier)*
*MSG_OOB (entier)*
*MSG_WAITALL (entier)*
*MSG_PEEK (entier)*
*MSG_DONTROUTE (entier)*
*SO_DEBUG (entier)*
*SO_REUSEADDR (entier)*
*SO_KEEPALIVE (entier)*
*SO_DONTROUTE (entier)*
*SO_LINGER (entier)*
*SO_BROADCAST (entier)*
*SO_OOBINLINE (entier)*
*SO_SNDBUF (entier)*
*SO_RCVBUF (entier)*
*SO_SNDLOWAT (entier)*
*SO_RCVLOWAT (entier)*
*SO_SNDTIMEO (entier)*
*SO_RCVTIMEO (entier)*
*SO_TYPE (entier)*
*SO_ERROR (entier)*
*SOL_SOCKET (entier)*
*PHP_NORMAL_READ (entier)*
*PHP_BINARY_READ (entier)*
*SOL_TCP (entier)*
*SOL_UDP (entier)*

## 6.35.7 Socket Errors

L'extension socket a été écrite pour fournir une interface utilisable avec les puissantes sockets fournies par BSD. Un soin particulier a été apporté pour que les fonctions soient aussi efficaces sous Unix que sous Windows32. Presque toutes les fonctions de sockets peuvent échouer dans certaines circonstances, et émettent ainsi un message d'alerte E_WARNING décrivant l'erreur. Parfois, cela ne se fait pas selon les souhaits du développeur. Par exemple, la fonction socket_read peut tout à coup émettre un message E_WARNING car la connexion a été inopinément interrompu. Il est commun de supprimer les erreurs avec l'opérateur @ et de traiter les erreurs avec la fonction socket_last_error , au niveau application. Vous pouvez appeler socket_strerror avec le code d'erreur pour connaître le message d'erreur, humainement lisible. Reportez vous à leur description pour plus de détails.

| Note |
| --- |
| Les messages E_WARNING généré par l'extension socket sont en anglais, mais le message d'erreur s'affichera en fonction de la configuration locale ( LC_MESSAGES ): |

```
Warning - socket_bind() unable to bind address [98]: Die Adresse wird bereits verwendet
```

## 6.35.8 Exemples

Exemple de socket : serveur TCP/IP simple

Cet exemple montre comment créer un serveur simple. Changez les variables address et port pour vous adapter à votre configuration, et exécutez le. Vous pourrez alors vous connecter au serveur avec une commande comme ceci : telnet 192.168.1.53 10000 (en supposant que l'adresse et le port correspondent à votre configuration). Tout ce que vous saisissez après cela sera envoyé au serveur, et affiché en retour. Pour vous déconnecter, tapez 'quit'.

```
 #!/usr/local/bin/php -q
<?php
error_reporting(E_ALL);

/* Autorise l'exécution infinie du script, en attente de connexion. */
set_time_limit(0);

/* Active le vidage implicite des buffers de sortie, pour que nous
 * puissions vois ce que nous lisons au fur et a mesure. */
ob_implicit_flush();

$address = '192.168.1.53';
$port = 10000;

if (($sock = socket_create(AF_INET, SOCK_STREAM, SQL_TCP)) < 0) {
    echo "socket_create() a échoué : raison : " . socket_strerror($sock) . "\n";
}

if (($ret = socket_bind($sock, $address, $port)) < 0) {
    echo "socket_bind() a échoué : raison : " . socket_strerror ($ret) . "\n";
}

if (($ret = socket_listen($sock, 5)) < 0) {
    echo "socket_listen() a échoué : raison : " . socket_strerror ($ret) . "\n";
}

do {
    if (($msgsock = socket_accept($sock)) < 0) {
        echo "socket_accept() a échoué : raison : " . socket_strerror ($msgsock) . "\n";
        break;
    }
    /* Send instructions. */
    $msg = "\Bienvenue sur le serveur de test PHP.\n" .
        "Pour quitter, tapez 'quit'. Pour éteindre le serveur, tapez 'shutdown'.\n";
    socket_write($msgsock, $msg, strlen($msg));

    do {
        if (FALSE === ($buf = socket_read($msgsock, 2048))) {
            echo "socket_read() a échoué : raison : " . socket_strerror ($ret) . "\n";
            break 2;
        }
        if (!$buf = trim($buf)) {
            continue;
        }
        if ($buf == 'quit') {
            break;
        }
        if ($buf == 'shutdown') {
            socket_close ($msgsock);
            break 2;
        }
        $talkback = "PHP: You said '$buf'.\n";
        socket_write($msgsock, $talkback, strlen($talkback));
        echo "$buf\n";
    } while (true);
    socket_close ($msgsock);
} while (true);

socket_close ($sock);
?>
```

Exemple avec les sockets : Client TCP/IP simple

Ce exemple illustre un client HTTP simple. Il se connecte simplement à une page, envoie une requête HEAD, affiche le résultat, et se termine.

```
 <?php
error_reporting(E_ALL);

echo "<h2>Connexion TCP/IP</h2>\n";
```

```
/* Lit le port du service WWW. */
$service_port = getservbyname('www', 'tcp');

/* Lit l'adresse IP du serveur de destination */
$address = gethostbyname('www.exemple.com');

/* Cree une socket TCP/IP. */
$socket = socket_create(AF_INET, SOCK_STREAM, SQL_TCP);
if ($socket < 0) {
    echo "socket_create() a échoué : raison :  " . socket_strerror ($socket) . "\n";
} else {
    echo "OK.\n";
}

echo "Essai de connexion à '$address' sur le port '$service_port'...";
$result = socket_connect($socket, $address, $service_port);
if ($result < 0) {
    echo "socket_connect() a échoué : raison : ($result) " . socket_strerror($result) . "\n";
} else {
    echo "OK.\n";
}

$in = "HEAD / HTTP/1.0\r\n\r\n";
$out = '';

echo "Envoi de la requête HTTP HEAD...";
socket_write($socket, $in, strlen($in));
echo "OK.\n";

echo "Lire la réponse : \n\n";
while ($out = socket_read($socket, 2048)) {
    echo $out;
}

echo "Fermeture de la socket...";
socket_close($socket);
echo "OK.\n\n";
?>
```


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Socket Errors
- Exemples
- socket_accept
- socket_bind
- socket_clear_error
- socket_close
- socket_connect
- socket_create_listen
- socket_create_pair
- socket_create
- socket_get_option
- socket_getpeername
- socket_getsockname
- socket_iovec_add
- socket_iovec_alloc
- socket_iovec_delete
- socket_iovec_fetch
- socket_iovec_free
- socket_iovec_set
- socket_last_error
- socket_listen
- socket_read
- socket_readv
- socket_recv

- socket_recvfrom
- socket_recvmsg
- socket_select
- socket_send
- socket_sendmsg
- socket_sendto
- socket_set_block
- socket_set_nonblock
- socket_set_option
- socket_shutdown
- socket_strerror
- socket_write
- socket_writev

## 6.35.10 socket_bind() : Lie un nom à une socket

bool **socket_bind** ( resource socket , string address , *int port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_bind lie le nom donné par address à la socket décrite par socket , qui doit être une ressource de socket valide, créée par socket_create .

Le paramètre d'adresse address peut être soit une adresse IP numérique (i.e. 127.0.0.1 ), si la socket est de la famille AF_INET ; ou bien le chemin d'une socket de domaine Unix, si la socket est de la famille AF_UNIX .

Le paramètre port n'est utilisé que lors de la connexion à une socket de type AF_INET , et désigne le port sur l'hôte distant, sur lequel une connexion doit être faite.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

Voir aussi socket_connect , socket_listen , socket_create , socket_last_error et socket_strerror .

## 6.35.11 socket_clear_error() : Efface toutes les erreurs précédemment générées par une socket

void **socket_clear_error** ( *resource socket* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_clear_error efface tous les codes d'erreurs qui ont été enregistrés pour la socket socket , ou bien pour la socket générale.

socket_clear_error permet de remettre à zéro les codes d'erreurs d'une socket ou de la socket globale. Cela peut être utile pour détecter l'avénement d'une erreur durant une partie de l'applcation.

Voir aussi socket_last_error et socket_strerror .

## 6.35.12 socket_close() : Ferme une socket

void **socket_close** ( resource socket )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_close ferme la socket socket , et détruit toutes les ressources qui y sont liées.

| Note |
| --- |
| socket_close ne peut pas être utilisée sur les ressources de fichiers créées par fopen , popen , fsockopen , ou psockopen ; cette fonction est destinée à traiter les sockets créées par les fonctions socket_create ou socket_accept . |

Voir aussi socket_bind , socket_listen , socket_create et socket_strerror .

## 6.35.13 socket_connect() : Crée une connexion sur une socket

bool **socket_connect** ( resource socket , string address , *int port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_connect créée une nouvelle connexion en utilisant la ressource socket socket , qui doit être une ressource de socket valide, créée par la fonction socket_create .

Le paramètre d'adresse address peut être soit une adresse IP numérique (i.e. 127.0.0.1 ), si la socket est de la famille AF_INET ; ou bien le chemin d'une socket de domaine Unix, si la socket est de la famille AF_UNIX .

Le paramètre port n'est utilisé que lors de la connexion à une socket de type AF_INET , et désigne le port sur l'hôte distant, sur lequel une connexion doit être faite.

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

Voir aussi socket_bind , socket_listen , socket_create , socket_last_error et socket_strerror .

## 6.35.14 socket_create_listen() : Ouvre une socket sur un port pour accepter les connexions

resource **socket_create_listen** ( int port , *int backlog* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_create_listen est destinée à simplifier la création de nouvelles sockets destinées être mises en attente, et accepter de nouvelles connexions.

socket_create_listen créée une nouvelle ressource de socket, de type AF_INET , mise en attente sur **toutes** les interfaces locales, pour le port port .

Le paramètre backlog définit la taille maximum de la queue de connexions en attente. SOMAXCONN peut être utilisée comme valeur pour le paramètre backlog . Reportez−vous à socket_listen pour plus de détails.

socket_create_listen retourne une nouvelle ressource de socket en cas de succès et FALSE en cas d'erreur.Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

| Note |
| --- |
| Si vous voulez créer une socket qui n'écoute que certaines interfaces, vous devez utiliser socket_create , socket_bind et socket_listen . |

Voir aussi socket_create , socket_bind , socket_listen , socket_last_error et socket_strerror .

## 6.35.15 socket_create_pair() : Crée une paire de socket identiques et les stocke dans fds.

bool **socket_create_pair** ( int domain , int type , int protocol , array &fd )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.16 socket_create() : Crée une socket (point de communication)

resource **socket_create** ( int domain , int type , int protocol )

socket_create crée un point de communication (une socket) et retourne une ressource de socket.

Le paramètre domain sélectionne le domaine (famille de protocole) à utiliser pour les

communications. Actuellement, AF_INET et AF_UNIX sont les valeurs possibles de ce paramètre. AF_INET est typiquement utilisé pour les communications Internet. AF_UNIX utilise des chemins systèmes pour identifier les sockets, et par conséquent, peut être uniquement utilisé pour les communications locales (mais elles seront aussi plus rapides).

Le paramètre type sélectionne le type de socket. Le type est représenté par les constantes suivantes : SOCK_STREAM , SOCK_DGRAM , SOCK_SEQPACKET , SOCK_RAW , SOCK_RDM ou SOCK_PACKET . Les deux types les plus courants sont SOCK_DGRAM pour le protocole UDP (connectionless) et SOCK_STREAM pour le protocole TCP .

protocol sélectionne le protocole qui peut être soit SOL_UDP , soit SOL_TCP .

socket_create retourne une ressource de socket en cas de succès et FALSE sinon. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

Pour plus d'informations sur l'utilisation de socket_create , et sur les significations des différents paramètres, reportez−vous au manuel Unix sur les sockets (2).

| Note |
| --- |
| Si le paramètre domain ou type est donné, socket_create utilisera par défaut les valeurs AF_INET et SOCK_STREAM , respectivement. De plus, elle générera un message d'aleret ( E_WARNING ). |

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen , socket_last_error , et socket_strerror .

## 6.35.17 socket_get_option() : Lit les options de socket

mixed **socket_get_option** ( resource socket , int level , int optname )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

| Note |
| --- |
| Cette fonction était appelée socket_getopt() avant PHP 4.3.0. |

## 6.35.18 socket_getpeername() : Interroge l'autre extrémité de la communication

bool **socket_getpeername** ( resource socket , string &addr , *int &port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

Si la socket socket est de type AF_INET , socket_getpeername retournera **l'adresse IP** de l'hôte, en notation numérique (e.g. 127.0.0.1 ) dans le paramètre address , et si le paramètre optionnel port est présent, elle retournera aussi le port de la communication établie.

Si la socket socket est de type AF_UNIX , socket_getpeername retournera le chemin sur le système de fichiers (e.g. /var/run/daemon.sock ) dans le paramètre address .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. socket_getpeername peut aussi retourner FALSE si le type de la socket n'est ni AF_INET ni AF_UNIX , auquel cas le dernier code d'erreur socket n'est **pas** modifié.

Voir aussi socket_getsockname , socket_last_error et socket_strerror .

## 6.35.19 socket_getsockname() : Interroge la socket locale

bool **socket_getsockname** ( resource socket , string &addr , *int &port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

Si la socket socket est de type AF_INET , socket_getsockname retournera **l'adresse IP** locale, en notation numérique (e.g. 127.0.0.1 ) dans le paramètre address , et si le paramètre optionnel port est présent, elle retournera aussi le port de la communication établie.

Si la socket socket est de type AF_UNIX , socket_getsockname retournera le chemin sur le système de fichiers (e.g. /var/run/daemon.sock ) dans le paramètre address .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. socket_getsockname peut aussi retourner FALSE si le type de la socket n'est ni AF_INET ni AF_UNIX , auquel cas le dernier code d'erreur socket n'est **pas** modifié.

Voir aussi socket_getpeername , socket_last_error et socket_strerror .

## 6.35.20 socket_iovec_add() : Ajoute un nouveau vecteur dans le tableau

bool **socket_iovec_add** ( resource iovec , int iov_len )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.21 socket_iovec_alloc() : Construit un 'struct iovec'

resource **socket_iovec_alloc** ( int num_vectors , *int* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.22 socket_iovec_delete() : Efface un vecteur

bool **socket_iovec_delete** ( resource iovec , int iov_pos )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.23 socket_iovec_fetch() : Lit les données du vecteur specifié par iovec_id[iovec_position]

string **socket_iovec_fetch** ( resource iovec , int iovec_position )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.24 socket_iovec_free() : Détruit le vecteur iovec specifié par iovec_id

bool **socket_iovec_free** ( resource iovec )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.25 socket_iovec_set() : Modifie les données du vecteur iovec_id[iovec_position]

bool **socket_iovec_set** ( resource iovec , int iovec_position , string new_val )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.26 socket_last_error() : Lit la dernière erreur générée par une socket

int **socket_last_error** ( *resource* *socket* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_last_error retourne un code d'erreur socket.

Si un paramètre est passé à cette fonction, la dernière erreur qui aura été générée par cette socket sera retournée. Si la ressource de socket est omise, le dernier code d'erreur généré est retourné. Ce comportement est particulièrement pratique pour des fonctions comme socket_create qui ne retourne pas de socket en cas d'echec, et socket_select qui peut échouer sans raison directement liée à la socket. Le code d'erreur peut être transmis à socket_strerror qui retourne un message d'erreur lisible.

```
<?php
 if (false == ($socket = @socket_create(AF_INET, SOCK_STREAM, SOL_TCP))) {
   die("Impossible de créer la socket, le code d'erreur est : " . socket_last_error() .
       ", le message d'erreur est : " . socket_strerror(socket_last_error()));
 }
?>
```

| Note |
| --- |
| socket_last_error n'efface pas le code d'erreur. Utilisez plutôt la fonction socket_clear_error pour cela. |

## 6.35.27 socket_listen() : Attend une connexion (listen) sur une socket

bool **socket_listen** ( resource socket , *int* *backlog* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

Une fois que la socket socket a été créée avec la fonction socket_create et liée à un nom avec la fonction socket_bind , elle peut être mise en attente de connexion entrantes.

Un nombre maximum de backlog connexions seront mises en attente de traitement. Si une demande de connexion arrive et que la queue est pleine, le client recevra une erreur indiquant ECONNREFUSED , ou, si le protocole de support accepte les retransmissions, la requête sera ignorée pour que les tentatives ultérieures finissent par réussir.

| Note |
| --- |
| Le nombre maximum passé dans le paramètre backlog dépend essentiellement de la plate forme de support. Sur Linux, c'est tronqué automatiquement à SOMAXCONN . Sous Windows32, si la constante SOMAXCONN est passée, le service responsable des sockets choisira une valeur maximum **raisonnable** . Il n'y a pas de méthode pour deviner la valeur réellement choisie. |

socket_listen ne fonctionne qu'avec des sockets de type SOCK_STREAM et SOCK_SEQPACKET .

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

Voir aussi socket_accept , socket_bind , socket_connect , socket_create et socket_strerror .

## 6.35.28 socket_read() : Lit des données d'une socket

string **socket_read** ( resource socket , int length , *int type* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_read lit des données dans la socket socket , qui doit être une ressource créée par socket_accept . Elle lit un maximum de length octets. Sinon, vous pouvez utiliser \r , \n or \0 pour terminer la lecture (suivant la valeur choisie pour type , voir ci−dessous).

socket_read retourne les données sous forme de chaîne en cas de succès, et FALSE sinon. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

| Note |
| --- |
| socket_read peut retourner une chaîne de longueur zéro, qui indique la fin de la communication (l'autre socket a fermé la communication). |

Le paramètre optinnel type peut prendre l'une des valeurs constantes suivantes :

- PHP_BINARY_READ − utilise la fonction système read() . Capable de lire les données binaires. (valeur par défaut en PHP >= 4.1.0)
- PHP_NORMAL_READ − la lecture s'arrête aux \n et \r. (valeur par défaut en PHP <= 4.0.6)

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen , socket_last_error , socket_strerror et socket_write .

## 6.35.29 socket_readv() : Lit des données dans un fd, en utilisant le tableau scatter−gather définit par iovec_id

bool **socket_readv** ( resource socket , resource iovec_id )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.30 socket_recv() : Reçoit des données d'une socket connectée

int **socket_recv** ( resource socket , string &buf , int len , int flags )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.31 socket_recvfrom() : Reçoit des données d'une socket, connectée ou pas

int **socket_recvfrom** ( resource socket , string &buf , int len , int flags , string &name , *int &port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.32 socket_recvmsg() : Reçoit des messages sur uen socket, qu'elle soit orientée connexion ou pas

bool **socket_recvmsg** ( resource socket , resource iovec , array &control , int &controllen , int &flags , string &addr , *int &port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

## 6.35.33 socket_select() : Exécute l'appel système select() un tableau de sockets avec une durée d'expiration

int **socket_select** ( array &read , array &write , array &except , int tv_sec , *int tv_usec* )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

socket_select accepte un tableau de sockets et attend qu'elles changent de statut. Ceux qui sont familiers avec les sockets de BSD reconnaitront dans ces tableaux de sockets les sets de file descriptor. Trois tableaux indépendants de ressources de sockets sont surveillés.

Les sockets listées dans le paramètre read seront surveillées en lecture : pour savoir quand elles sont disponibles en lecture (plus précisément, si une lecture ne va pas bloquer, en particulier, une ressource de socket a déjà atteint une fin de fichier, auquel cas socket_read retournera une chaîne de taille zéro).

Les sockets listées dans write seront surveillées en écriture : pour voir si une écriture ne va pas bloquer.

Les sockets listées dans except seront surveillées pour leurs exceptions.

Attention

En sortie de fonction, les tableaux sont modifiées pour indiquer quelles sockets ont changés d'états.

Vous n'avez pas besoin de passer tous les tableaux à socket_select . Vous pouvez les omettre, ou utiliser un tableau vide, ou encore NULL à la place. N'oubliez pas que ces tableaux sont passés par **référence** et seront modifiés par socket_select .

Exemple avec socket_select

```
<?php
 /* Prépare le tableau read (socket surveillées en lecture) */
 $read = array($socket1, $socket2);

 if (false === ($num_changed_sockets = socket_select($read, $write = NULL, $except = NULL, 0))) {
   /* Gestion des erreurs */
 else if ($num_changed_sockets > 0) {
   /* Au moins une des sockets a été modifié */
 }
?>
```

Note

A cause d'une limitation du Zend Engine actuel, il n'est pas possible de passer une constante comme NULL directement comme paramètre à cette fonction, qui attend une valeur par référence. A la place, utilisez un tableau temporaire ou une expression dont le membre de gauche est une variable temporaire :

Passer NULL à socket_select

```
<?php
 socket_select($r, $w, $e = NULL, 0);
?>
```

Les paramètres tv_sec and tv_usec ensemble forment le paramètre **timeout** (durée de vie). Le **timeout** est la durée maximale de temps avant que socket_select ne se termine. tv_sec peut être zéro , ce qui fera que socket_select retournera immédiatement. C'est très pratique pour faire du polling (sondage). Si tv_sec est NULL (pas de timeout), socket_select peut se bloquer indéfiniment.

En cas de succès, socket_select retourne le nombre de sockets contenus dans les tableaux modifiées. Ce nombre peut être zéro si la durée maximale d'attente a été atteinte. En cas d'erreur, FALSE est retourné. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

Note

Assurez vous bien d'utiliser l'opérateur === lorsque vous vérifiez les erreurs. Etant donnée que socket_select peut retourner 0, la comparaison avec FALSE via == donnerait TRUE :

Analyser le résultat de socket_select

```
<?php
 if (false === socket_select($r, $w, $e = NULL, 0)) {
   echo "socket_select() a échoué. Raison : " .
         socket_strerror(socket_last_error()) . "\n";
 }
?>
```

Note

Méfiez vous des implémentations de sockets, qui doivent être manipulées avec délicatesse. Quelques règles de base :

- Vous devez toujours essayer d'utiliser socket_select sans timeout. Votre programme ne devrait avoir rien à faire si il n'y a pas de données disponibles. Le code qui dépends d'un timeout est généralement peut portable, et difficile à déboguer.
- Une socket ne doit pas être ajoutée à l'un des tableaux en paramètre, si vous ne souhaitez pas vérifier le résultat après l'appel à socket_select . Après le retour de socket_select , toutes les sockets dans tous les tableaux doivent être vérifiées. Toute socket qui est disponible en écriture ou en lecture doit être utilisée pour écrire ou lire.
- Si vous écrivez ou lisez avec une socket retournée dans un tableau, soyez conscient qu'elles ne pourront pas écrire ou lire toutes les données que vous demandez. Soyez prêts à ne pouvoir lire qu'un seul octet.
- Il est commun à la plupart des implémentations de socket que la seule exception interceptée par les sockets dans le tableau except est le cas des données hors limites, reçues par une socket.

Voir aussi socket_read , socket_write , socket_last_error et socket_strerror .

## 6.35.34 socket_send() : Envoie des données à une socket connectée

int **socket_send** ( resource socket , string buf , int len , int flags )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

La fonction socket_send envoie len octets à la socket socket depuis le buffer buf .

Le paramètre flags peut être la combinaison des valeurs suivantes :

| | |
|---|---|
| 0x1 | Traite les données OOB (out−of−band). |
| 0x2 | Traite le message entrant (peek). |
| 0x4 | Ignore le routage, utilise une interface directe. |
| 0x8 | Les données complètent l'enregistrement. |
| 0x100 | Les données complètent la transaction. |

Voir aussi socket_sendmsg et socket_sendto .

## 6.35.35 socket_sendmsg() : Envoie un message à une socket, qu'elle soit orientée connexion ou pas

bool **socket_sendmsg** ( resource socket , resource iovec , int flags , string addr , *int port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.36 socket_sendto() : Envoie une message à une socket, qu'elle soit connectée ou pas

int **socket_sendto** ( resource socket , string buf , int len , int flags , string addr , *int port* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_sendto envoie len octets issus du buffer buf via la socket socket , vers le port port , à l'adresse addr

Le paramètre flags peut prendre l'une des valeurs suivantes :

| | |
| --- | --- |
| 0x1 | Traite les données OOB (out−of−band). |
| 0x2 | Traite le message entrant (peek). |
| 0x4 | Ignore le routage, utilise une interface directe. |
| 0x8 | Les données complètent l'enregistrement. |
| 0x100 | Les données complètent la transaction. |

Exemple avec socket_sendto

```
<?php
   $sh = socket_create(AF_INET,SOCK_STREAM,SOL_TCP);
   if (socket_bind($sh, '127.0.0.1', 4242)) {
       echo "Socket connectée correctement";
   }
   $buf = 'Test Message';
   $len = strlen($buf);
   if (socket_sendto($sh, $buf, $len, 0x100, '192.168.0.2', 4242) !== FALSE) {
       echo "Message envoyé correctement";
   }
   socket_close($sh);
?>
```

Voir aussi socket_send et socket_sendmsg .

## 6.35.37 socket_set_block() : Met la socket en mode bloquant

bool **socket_set_block** ( resource socket )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

Voir aussi socket_set_nonblock et socket_set_option

## 6.35.38 socket_set_nonblock() : Sélectionne le mode non bloquant d'un descripteur de fichier

bool **socket_set_nonblock** ( resource socket )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.35.39 socket_set_option() : Modifie les options de socket

bool **socket_set_option** ( resource socket , int level , int optname , mixed optval )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

| Note |
| --- |
| Cette fonction était appelée socket_setopt() avant PHP 4.3.0. |

## 6.35.40 socket_shutdown() : Eteind une socket en lecture et/ou écriture

bool **socket_shutdown** ( resource socket , *int how* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_shutdown vous permet d'empêcher les données entrantes ou sortantes ou les deux (par défaut), d'être émise via la socket socket

La valeur de how peut être l'une des suivantes :

| | |
| --- | --- |
| 0 | Empêche la lecture de la socket |
| 1 | Empêche l'écriture de la socket |
| 2 | Empêche l'écriture et la lecture de la socket |

## 6.35.41 socket_strerror() : Retourne une chaîne décrivant un message d'erreur

string **socket_strerror** ( int errno )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_strerror prend un code d'erreur comme paramètre errno . Cette valeur est souvent retournée par la fonction socket_last_error . La fonction retourne le message d'erreur correspondant. Cela rend plus facile la compréhension des problèmes survenus. Par exemple, au lieu d'avoir à rechercher un fichier système dans lequel il y a l'explication de '−111', vous avez juste à le passer à socket_strerror , qui vous dira ce qui c'est passé.

Exemple avec socket_strerror

```
 <?php
if (false == ($socket = @socket_create(AF_INET, SOCK_STREAM, SQL_TCP))) {
   echo "socket_create() a échoué : raison : " . socket_strerror(socket_last_error()) . "\n";
}

 if (false == (@socket_bind($socket, '127.0.0.1', 80))) {
   echo "socket_bind() a échoué : raison : " . socket_strerror(socket_last_error($socket)) . "\n";
}
?>
```

L'affichage attendu pour l'exemple du dessus (en supposant que vous essayer d'exécuter le script sans les droits de root) :
socket_bind() a échoué : raison : Permission denied

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen et socket_create .

## 6.35.42 socket_write() : Ecrit dans une socket

int **socket_write** ( resource socket , string buffer , *int length* )

| Attention |
| --- |
| Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls. |

socket_write écrit dans la socket socket les données du buffer buffer .

Le paramètre optionnel length peut spécifier explicitement la taille des données qui doivent être écrites. Si cette longueur est plus grande que la taille du buffer, elle sera ramenée automatiquemetn à la taille du buffer lui−même.

socket_write retourne le nombre d'octets qui ont pus êter écrits dans la socket, ou bien FALSE en cas d'erreur. Le code d'erreur généré peut être obtenu en appelant la fonction socket_last_error . Ce code d'erreur peut être passé à la fonction socket_strerror pour obtenir un message d'erreur, humainement lisible.

| Note |
| --- |
| socket_write n'écrit pas nécessaire tous les octets du buffer fourni. Il est valide que, suivant certaines configuration de buffer réseau, que seulement une certaine quantité de données, même |

un octets, soit écrit, même si votre buffer est plus grand. Vous devez alors vous assurer que vous n'avez pas oublié de transmettre le reste de vos données.

Note

Il est parfaitement valide pour socket_write de retourner zéro, ce qui signifie qu'aucun octet n'a été écrit. Soyez bien sûr d'utiliser l'opérateur === pour comparer le retour de la fonction avec FALSE , et détecter un cas d'erreur.

Voir aussi socket_accept , socket_bind , socket_connect , socket_listen , socket_read et socket_strerror .

## 6.35.43 socket_writev() : Ecrit des données dans un descripteur de fichier, fd, en utilisant le tableau scatter−gather définie par iovec_id

bool **socket_writev** ( resource socket , resource iovec_id )

Attention

Cette fonction est **EXPERIMENTALE** . Cela signifie que le comportement de cette fonction, son nom et concrètement, TOUT ce qui est documenté ici peut changer dans un futur proche, SANS PREAVIS! Soyez−en conscient, et utilisez cette fonction à vos risques et périls.

Attention

Cette fonction n'est pas documentée et seule la liste des arguments est disponible.

# 6.36 Chaînes de caractères (Strings)

## 6.36.1 Introduction

Ces fonctions permettent de manipuler des chaînes de caractères de différentes façons. Certaines fonctionnalités plus spécialisées sont disponibles dans les sections d'expression régulières et de gestion des URL .

Pour plus de détails sur le comportement des chaînes de caractères, notamment avec les guillemets simples et doubles, les séquences d'échappement, reportez−vous à la section Chaînes de caractères dans la section Types .

## 6.36.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.36.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.36.4 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*CRYPT_SALT_LENGTH integer*
*CRYPT_STD_DES integer*
*CRYPT_EXT_DES integer*
*CRYPT_MD5 integer*
*CRYPT_BLOWFISH integer*
*HTML_SPECIALCHARS ( integer )*
*HTML_ENTITIES ( integer )*
*ENT_COMPAT ( integer )*
*ENT_QUOTES ( integer )*
*ENT_NOQUOTES ( integer )*
*CHAR_MAX ( integer )*
*LC_CTYPE ( integer )*
*LC_NUMERIC ( integer )*
*LC_TIME ( integer )*
*LC_COLLATE ( integer )*
*LC_MONETARY ( integer )*
*LC_ALL ( integer )*
*LC_MESSAGES ( integer )*
*STR_PAD_LEFT ( integer )*
*STR_PAD_RIGHT ( integer )*
*STR_PAD_BOTH ( integer )*

## 6.36.5 Voir aussi

Pour des fonctions encore plus puissantes de gestion et manipulation des chaînes, reportez−vous aux expressions régulières POSIX et expressions régulières Perl .

- Introduction
- Pré−requis
- Installation
- Constantes prédefinies
- Voir aussi
- addcslashes
- addslashes
- bin2hex
- chop
- chr
- chunk_split
- convert_cyr_string
- count_chars
- crc32
- crypt
- echo
- explode
- fprintf
- get_html_translation_table
- hebrev
- hebrevc
- html_entity_decode
- htmlentities
- htmlspecialchars
- implode
- join
- levenshtein
- localeconv
- ltrim
- md5_file
- md5
- metaphone
- money_format
- nl_langinfo
- nl2br
- number_format
- ord
- parse_str
- print
- printf
- quoted_printable_decode
- quotemeta
- rtrim
- setlocale
- sha1_file
- sha1
- similar_text
- soundex
- sprintf

- sscanf
- str_ireplace
- str_pad
- str_repeat
- str_replace
- str_rot13
- str_shuffle
- str_split
- str_word_count
- strcasecmp
- strchr
- strcmp
- strcoll
- strcspn
- strip_tags
- stripcslashes
- stripos
- stripslashes
- stristr
- strlen
- strnatcasecmp
- strnatcmp
- strncasecmp
- strncmp
- strpos
- strrchr
- strrev
- strripos
- strrpos
- strspn
- strstr
- strtok
- strtolower
- strtoupper
- strtr
- substr_count
- substr_replace
- substr
- trim
- ucfirst
- ucwords
- vprintf
- vsprintf
- wordwrap

## 6.36.7 addslashes() : Ajoute des anti−slashs dans une chaîne

string **addslashes** ( string str )

addslashes retourne la chaîne str , après avoir échappé tous les caractères qui doivent l'être, pour être utilisé dans une requête de base de données. Ces caractères sont les guillemets simples ( ' ), guillemets doubles ( " ), anti−slash ( \ ) et NUL (le caractère NULL ).

| Note |
| --- |
| magic_quotes_gpc est activé par défaut. |

Voir aussi stripslashes , htmlspecialchars , quotemeta et get_magic_quotes_gpc .

## 6.36.8 bin2hex() : Convertit des données binaires en représentation hexadécimale

string **bin2hex** ( string str )

**bin2hex** retourne la chaîne str dont tous les caractères sont représentés par leur équivalent hexadécimal. La chaîne retourneé est une chaîne ASCII. La conversion supporte les caractères binaires, et utilise les bits de poids forts.

Voir aussi pack et unpack .

## 6.36.9 chop() : Alias de rtrim

chop est un alias de rtrim .

| Note |
| --- |
| chop est différente de son équivalent Perl chop() , qui supprime le dernier caractère dans la chaîne. |

## 6.36.10 chr() : Retourne un caractère spécifique

string **chr** ( int ascii )

**chr** retourne une chaîne d'un seul caractère, dont le code ASCII est donné par le paramètre ascii .

Exemple avec chr

```
<?php
/* Ajoute un caractère d'échappement à la fin de la chaîne $str */
  $str .= chr(27);

 /* Ceci est souvent plus pratique, et réalise la même chose */
  $str = sprintf("The string ends in escape: %c", 27);
?>
```

Vous pouvez trouvez une table des codes ASCII ici : http://www.asciitable.com .

Cette fonction est l'inverse de ord . Voir aussi sprintf , et le caractère de formattage %c .

## 6.36.11 chunk_split() : Scinde une chaîne

string **chunk_split** ( string body , *int chunklen* , *string end* )

chunk_split scinde la chaîne body en segments de chunklen octets de longueur. Cette fonction est très pratique pour convertir des résultats de base64_encode au format de la RFC 2045. Elle insère le paramètre end (par défaut, " \r\n ") tous les chunklen caractères (par défault, 76). Elle retourne la nouvelle chaîne, et ne modifie pas la chaîne originale.

Exemple avec chunk_split

```
<?php
// Formatter des données pour suivre la norme RFC 2045

  $new_string = chunk_split(base64_encode($data));
?>
```

Voir aussi explode , split , wordwrap et RFC 2045 .

## 6.36.12 convert_cyr_string() : Convertit une chaîne d'un jeu de caractères cyrillique à l'autre

string **convert_cyr_string** ( string str , string from , string to )

convert_cyr_string retourne la chaîne str , après l'avoir traduite depuis le jeu de caractères cyrillique from vers to . Les paramètres from et to sont des caractères qui représentent les jeux de caractères d'origine et de destination. Les types suivants sont supportés :

- k − koi8−r
- w − windows−1251
- i − iso8859−5
- a − x−cp866
- d − x−cp866
- m − x−mac−cyrillic

## 6.36.13 count_chars() : Retourne des statistiques sur les caractères utilisés dans une chaîne

mixed **count_chars** ( string string , *int mode* )

**count_chars** compte le nombre d'occurences de tous les octets présents dans la chaîne string et retourne différentes statistiques. Le paramètre optionnel mode vaut par défaut 0. Suivant la valeur de mode , count_chars retourne les informations suivantes :

- 0 − un tableau avec l'octet en index, et la fréquence correspondante pour chaque octet.
- 1 − identique à 0 mais seules les fréquences supérieures à zéro sont listées.
- 2 − identique à 0 mais seules les fréquences nulles sont listées.
- 3 − une chaîne contenant tous les octets utilisés est retournée.
- 4 − une chaîne contenant tous les octets non utilisés est retournée.

Exemple avec count_chars

```
<?php

$data = "Deux D et un F.";

$result = count_chars($data, 0);

for ($i=0; $i < count($result); $i++) {
   if ($result[$i] != 0)
       echo "Il y avait $result[$i] occurence(s) de \"" , chr($i) , "\" dans la phrase.\n";
}

?>
```

Cela affichera :

```
Il y avait 4 occurence(s) de " " dans la phrase.
Il y avait 1 occurence(s) de "." dans la phrase.
Il y avait 2 occurence(s) de "D" dans la phrase.
Il y avait 1 occurence(s) de "F" dans la phrase.
```

```
Il y avait 2 occurence(s) de "e" dans la phrase.
Il y avait 1 occurence(s) de "n" dans la phrase.
Il y avait 1 occurence(s) de "t" dans la phrase.
Il y avait 2 occurence(s) de "u" dans la phrase.
Il y avait 1 occurence(s) de "x" dans la phrase.
```

Voir aussi strpos et substr_count .

## 6.36.14 crc32() : Calcule la somme de contrôle CRC32

int **crc32** ( string str )

crc32 génère la somme de contrôle cyclique CRC 32, calculée sur 32 bits, et appliqué à la chaîne str . Cette fonction est généralement utilisée pour valider l'intégritée de données durant une transmission.

| Note |
| --- |
| Comme le type d'entier de PHP est signé, et que de nombreuses sommes de contrôle CRC32 aboutissent à un nombre négatif, vous devez utiliser le format "%u" de la fonction sprintf ou printf pour obtenir la représentation littérale de la somme de contrôle CRC32. |

Cet exemple illustre comment afficher la somme de contrôle avec la fonction printf :

Afficher une somme de contrôle CRC32

```php
<?php
 $checksum = crc32("Le vif zéphyr jubile sur les kumquats du clown gracieux.");
 printf("%u\n", $checksum);
?>
```

Voir aussi md5 et sha1 .

## 6.36.15 crypt() : Chiffrage indéchiffrable (hashing)

string **crypt** ( string str , *string salt* )

crypt retourne la chaîne str chiffrée avec l'algorithme standard Unix *DES* , ou bien un des algorithmes disponibles sur la machine. L'argument optionnel salt sera utilisé comme base pour le chiffrement. Reportez vous aux pages de manuel Unix pour plus de détails sur la fonction crypt.

Si l'argument salt n'est pas fourni, PHP en générera un lui−même.

Certains systèmes supportent plus d'un type de chiffrement. En fait, il arrive que le chiffrement DES standard soit remplacé par un algorithme de chiffrement MD5. Le choix du type de chiffrement est effectué en se basant sur la valeur du salt . A l'installation, PHP détermine les possibitilités de la fonction crypt(), et acceptera des salt pour d'autres types de chiffrements. Si aucun salt n'est fourni, PHP va en générer un de deux caractères, à moins que le système par défaut soit MD5, auquel cas un salt compatible MD5 sera généré. PHP définit une constante appelée CRYPT_SALT_LENGTH qui vous dira si un salt de deux caractères s'applique à votre système, ou bien si c'est un salt de 12 caractères.

Si vous utilisez le salt généré, pensez bien que le salt est généré une seule fois. Si vous appelez cette fonction récursivement, cela peut avoir des impacts sur la sécurité et l'apparence du résultat.

crypt , lorsqu'elle est utilisée avec le chiffrement standard DES, retourne le salt dans les deux premiers caractères de la chaîne retournée. Elle n'utilise que les 8 premiers caractères de str , ce qui fait que toutes les chaînes plus longues, qui ont les mêmes premiers 8 octets retourneront le même résultat (tant que le salt est toujours le même).

Sur les sytèmes où crypt supporte plusieurs types de chiffrements, les constantes suivantes sont mises à 1 ou 0, suivant que le type correspondant est disponible :

- CRYPT_STD_DES − Chiffrement DES standard à 2 caractères
- CRYPT_EXT_DES − Chiffrement DES étendu à 9 caractères
- CRYPT_MD5 − Chiffrement MD5 à 12 caractères commençant par $1$
- CRYPT_BLOWFISH − Chiffrement Blowfish à 16 caractères commençant par $2$

Note

Il n'existe pas de fonction de déchiffrement, car la fonction crypt utilise un algorithme à un seul sens (injection).

Exemple avec crypt

```
 <?php
$mot_de_passe = crypt("MonM0TdEPa&&e");
 // laissons le salt initialisé par
 <font face="courrier">PHP</font>


/*
  Il vaut mieux passer le résultat complet de crypt() comme salt nécessaire
  pour le chiffrement du mot de passe, pour éviter les problèmes entre les
  algorithmes utilisés (comme nous le disons ci-dessus, le chiffrement
  standard DES utilise un salt de 2 caractères, mais un chiffrement
   MD5 utilise un salt de 12.
*/
if (crypt($utilisateur_input,$mot_de_passe_input) == $mot_de_passe) {
   echo "Mot de passe correct!";
}
?>
```

Voir aussi md5 et l'extension Mcrypt .

## 6.36.16 echo() : Affiche une chaîne de caractères

void **echo** ( string arg1 , *string argn...* )

Affiche tous les paramètres.

echo n'est pas vraiment une fonction (c'est techniquement une structure du langage). Cela fait que vous n'êtes pas obligés d'utiliser des parenthèses. En fait, si vous voulez passer plusieurs paramètres à afficher, vous ne devez pas utiliser de parenthèses. Il n'est pas possible d'utiliser echo dans un contexte de fonction variable . Dans ce cas, il faut utiliser print à la place.

Exemples avec echo

```
 <?php
  echo "Bonjour le monde";

echo "Cet echo() se
répartit sur plusieurs lignes. Il affiche aussi les
nouvelles lignes";

echo "Cet echo() se\nrépartit sur plusieurs lignes. Il affiche aussi les\nnouvelles lignes";

echo "L'échappement de caractères se fait : \"comme ceci\".";

// Vous pouvez utiliser des variables avec echo()
$foo = "foobar";
$bar = "barbaz";

echo "foo vaut \"$foo\""; // foo vaut "foobar"

// Vous pouvez aussi utiliser des tableaux
$bar = array("clef" => "foo");

echo "c'est {$bar['clef']} !"; // c'est foo !

 // Les guillemets simples annulent le remplacement des variables
echo 'foo vaut $foo'; // foo vaut $foo

// Si vous n'utilisez pas d'autres caractères,
 // vous pouvez afficher plusieurs variables
// en les séparant par des virgules
echo $foo;          // foobar
echo $foo,$bar;     // foobarbarbaz
```

```
echo <<<END
Cette syntaxe s'intitule le "here document" et
 permet d'afficher plusieurs lignes avec de
 l'interpolation de variables. Notez que la fin de
la syntaxe doit apparaître sur une nouvelle ligne,
avec uniquement un point-virgule, et pas d'espace
de plus !
END;

// comme echo() n'est pas une fonction, le code suivant n'est pas valide
($some_var) ? echo('Oui'): echo('Non');

// Cependant, les lignes suivantes sont valides :
($some_var) ? print('Oui'): print('Non'); // print est une fonction
echo $some_var ? 'Oui': 'Non';
?>
```

echo dispose aussi d'une version courte, où vous pouvez faire suivre la balise PHP ouvrante d'un signe égal ( = ).

Afficher sans l'aide de echo

```
 J'ai <?=$foo?> fous.
```

Note

Cette syntaxe courte n'est possible que si la configuration de short_open_tag a été activée.

Pour une brève discussion à propos des différences entre print et echo , consultez cet article des FAQTs Knowledge Base : http://www.faqts.com/knowledge_base/view.phtml/aid/1/fid/40

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Voir aussi print , printf et flush .

## 6.36.17 explode() : Coupe une chaîne en segments

array **explode** ( string separator , string string , *int limit* )

**explode** retourne un tableau de chaînes. Ce sont les sous-chaînes, extraites de string , en utilisant le séparateur separator . Si limit est fourni, le tableau retourné aura un maximum de limit éléments, et le dernier élément du tableau contiendra le reste de la chaîne string .

Si separator est une chaîne vide ("""), explode retournera FALSE . Si separator contient une valeur qui n'est pas contenue dans string , alors explode retournera un tableau, contenant la chaîne string entière.

Note

Le paramètre limit a été ajouté en PHP 4.0.1.

Exemple avec explode

```
<?php
 $pizza = "garniture1 garniture2 garniture3 garniture4";
 $pieces = explode(" ", $pizza);

 $data = "foo:*:1023:1000::/home/foo:/bin/sh";
 list($user,$pass,$uid,$gid,$gecos,$home,$shell) = explode(":",$data);
?>
```

Note

bien que pour des raisons historiques, implode accepte les paramètres dans un ordre ou l'autre, explode ne le peut pas. Vous devez vous assurer que l'argument séparateur separator arrive avant

l'argument string .

Voir aussi preg_split , spliti , split et implode .

## 6.36.18 fprintf() : Ecrit une chaîne formatée dans un flux

int **fprintf** ( resource handle , string format , *mixed* *args* )

fprintf écrit la chaîne produite avec le format format dans le flux représenté par handle .

Le paramètre format est composé de zéro, une ou plusieurs directives : des caractères ordinaires (excluant % ) qui sont copiés directement dans le résultat sans modification, et des **spécifications de conversion** , qui sont exploitées avec un des paramètres suivants. Cette technique s'applique à fprintf , sprintf et printf .

Chaque spécification de conversion est constituée d'un signe pourcentage, ( % ), suivi par un ou plusieurs éléments, dans cet ordre :

1. Un **remplisseur** optionnel qui indique quel caractère sera utilisé pour compléter le résultat jusqu'à la longueur requise. Ce peut être le caractère d'espace, ou le caractère 0 . Par défaut, le remplissage se fait avec des espaces. Un autre caractère de remplissage peut être spécifié en le préfixant avec un guillemet simple ( ' ) : voir les exemples ci−dessous.
2. Un **spécificateur d'alignement** qui indique si le résultat doit être aligné à gauche ou à droite. Par défaut, le résultat est aligné à gauche. Le caractère − fera que le résultat est aligné à droite.
3. Un nombre optionnel, **spécificateur de taille** indique le nombre minimum de caractères que cette conversion doit fournir en résultat.
4. Un **spécificateur de précision** qui indique le nombre de décimales qui doivent être affichées pour les nombres à virgule flottante. Cette option n'a pas d'effet pour les autres types que float . Une auter fonction pratique pour formatter les nombres à virgule flottante est number_format .
5. Un **spécificateur de type** qui indique le type avec lequel l'argument sera traité. Plusieurs types possibles :
    - % − un caractère de pourcentage litéral. Aucun argument n'est nécessaire.
    - b − l'argument est traité comme un entier, et présenté comme un nombre binaire.
    - c − l'argument est traité comme un entier, et présenté comme le caractère de code ASCII correspondant.
    - d − l'argument est traité comme un entier, et présenté comme un nombre décimal signé.
    - u − l'argument est traité comme un entier, et présenté comme un nombre décimal non signé.
    - f − l'argument est traité comme un nombre à virgule flottante (type float ), et présenté comme un nombre à virgule flottante.
    - o − l'argument est traité comme un entier, et présenté comme un nombre octal.
    - s − l'argument est traité et présenté comme une chaîne de caractères.
    - x − l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en minuscules).
    - X − l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en majuscules).

Voir aussi printf , sprintf , sscanf , fscanf , vsprintf et number_format .

## 6.36.19 get_html_translation_table() : Retourne la table de traduction des entités utilisée par htmlspecialchars et htmlentities

array **get_html_translation_table** ( int table , *int quote_style* )

get_html_translation_table retourne la table de traduction des entités utilisée en interne par les fonctions htmlspecialchars et htmlentities . Il existe deux constantes ( HTML_ENTITIES et HTML_SPECIALCHARS ), qui vous permettent de spécifier la table que vous souhaitez. Et comme dans les fonctions htmlspecialchars et htmlentities , vous pouvez optionnellement spécifier le type de guillemets que vous voulez utiliser. Le comportement par défaut est le mode ENT_COMPAT. Pour la description de ces modes, reportez vous à htmlspecialchars .

Exemple avec la table de traduction des caractères en entités HTML

```
<?php
 $trans = get_html_translation_table(HTML_ENTITIES);
 $str = "Hallo & <Frau> & Krämer";
 $encoded = strtr($str, $trans);
?>
```

La variable $encoded contient maintenant "Hallo & <Frau> & Krämer".

Un truc pratique est d'utiliser array_flip pour changer la direction de la traduction.

Exemple avec inversion de la table de traduction des caractères en entités HTML

```
<?php
 $trans = array_flip($trans);
 $original = strtr($encoded, $trans);
?>
```

Le contenu de $original sera alors : "Hallo & <Frau> & Krämer" .

Voir aussi htmlspecialchars , htmlentities , strtr et array_flip .

## 6.36.20 hebrev() : Convertit un texte logique hébreux en texte visuel

string **hebrev** ( string hebrew_text , *int max_chars_per_line* )

**hebrev** convertit le texte logique hébreux hebrew_text en texte visuel. Le paramètre optionnel max_chars_per_line indique le nombre maximum de caractères par lignes dans le résultat. hebrev essaie d'éviter de scinder les mots.

Voir aussi hebrevc .

## 6.36.21 hebrevc() : Convertit un texte logique hébreux en texte visuel, avec retours à la ligne

string **hebrevc** ( string hebrew_text , *int max_chars_per_line* )

hebrevc est similaire à hebrev à la différence qu'elle convertit les nouvelles lignes (\n) en "<br>\n". Le paramètre optionnel max_chars_per_line indique le nombre maximum de caractères par lignes dans le résultat. hebrevc essaie d'éviter de scinder les mots.

Voir aussi hebrev .

## 6.36.22 html_entity_decode() : Convertit toutes les entités HTML en caractères normaux

string **html_entity_decode** ( string string , *int quote_style* , *string charset* )

html_entity_decode est la fonction contraire de htmlentities : elle convertit les entités HTML de la chaîne string en caractères normaux.

Le paramètre optionnel quote_style vous permet de définir ce qu'il adviendra des guillemets simples et doubles. Ce paramètre prend l'une des valeurs suivantes (et la valeur par défaut est ENT_COMPAT ) :

| Constante | Description |
|---|---|
| ENT_COMPAT | Convertit les guillemets doubles et ignore les guillemets simples. |
| ENT_QUOTES | Convertit les guillemets doubles et les guillemets simples. |
| ENT_NOQUOTES | Convertit les guillemets simples et ignore les guillemets doubles. |

Le jeu de caractères ISO−8859−1 est utilisé par défaut, comme paramètre charset . Ce paramètre permet de choisir le jeu de caractères utilisés dans la conversion.

Les jeux de caractères suivantes sont disponibles et supportés par PHP 4.3.0 et plus récent.

| Jeux de caractères | Description |
|---|---|
| ISO−8859−1 | Europe occidentale, Latin−1 |
| ISO−8859−15 | Europe occidentale, Latin−9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manque au Latin−1(ISO−8859−1). |
| UTF−8 | Unicode 8 bits multi octets, compatible avec l'ASCII |
| cp1252 | Jeux de caractères spécifique de Windows pour l'Europe occidentale. |
| BIG5 | Chinois traditionnel, principalement utilisé à Taiwan. |
| GB2312 | Chinois simplifié, officiel. |
| BIG5−HKSCS | Big5 avec les extensions de Hong Kong, chinois traditionnel. |
| Shift_JIS | Japonais |
| EUCJP | Japonais |

Note

Les autres jeux de caractères ne sont pas reconnus, et le ISO−8859−1 sera utilisé à la place.

Décoder des entités HTML

```
<?php
 $orig = "J'ai \"sorti\" le <b>chien</b> tout à l'heure";
 $a = htmlentities($orig);
 $b = html_entity_decode($a);

 echo $a; // J'ai "sorti" le <b>chien</b> tout à l'heure
 echo $b; // J'ai "sorti" le <b>chien</b> tout à l'heure

// Pour les utilisateurs ayant des versions antérieures à PHP 4.3.0 :
 function unhtmlentities ($string) {
   $trans_tbl = get_html_translation_table (HTML_ENTITIES);
   $trans_tbl = array_flip ($trans_tbl);
   return strtr ($string, $trans_tbl);
 }

 $c = unhtmlentities($a);
 echo $c;
 // J'ai "sorti" le <b>chien</b> tout à l'heure
?>
```

Note

Vous pourriez vous demander pouquoi trim(html_entity_decode(' ')); ne résuit pas la chaîne à la chaîne vide. C'est parceque l'entité ' ' n'est pas un code ASCII 32 (qui serait supprimé par trim ) mais un code ASCII 160 (0xa0) dans le jeu de caractères par défaut ISO 8859−1.

Voir aussi htmlentities , htmlspecialchars , get_html_translation_table , htmlspecialchars et urldecode .

## 6.36.23 htmlentities() : Convertit tous les caractères éligibles en entités HTML

string **htmlentities** ( string string , *int* *quote_style* , *string* *charset* )

**htmlentities** est identique à la fonction htmlspecialchars , sauf que tous les caractères qui ont des équivalents en entités HTML sont effectivement traduits. Comme htmlspecialchars , cette fonction prend un deuxième argument optionnel, qui indique comment doivent être traités les guillemets doubles et simples. Vous pouvez utiliser l'une des constantes suivantes :

| Nom | Description |
|---|---|
| ENT_COMPAT | Convertit les guillemets doubles, et ignore les guillemets simples |
| ENT_QUOTES | Convertit les guillemets doubles et les guillemets simples |
| ENT_NOQUOTES | Ignore les guillemets doubles et les guillemets simples |

Le support du paramètre optionnel quote a été ajouté en PHP 4.0.3.

Comme htmlspecialchars , cette fonction prend un troisième argument optionnel qui définit le jeux de caractères utilisé durant la conversion. Le support de cet argument a été ajouté en PHP 4.1.0. Actuellement, le jeu de caractères ISO−8859−1 est utilisé par défaut.

Les jeux de caractères suivantes sont disponibles et supportés par PHP 4.3.0 et plus récent.

| Jeux de caractères | Description |
|---|---|
| ISO−8859−1 | Europe occidentale, Latin−1 |
| ISO−8859−15 | Europe occidentale, Latin−9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manque au Latin−1(ISO−8859−1). |
| UTF−8 | Unicode 8 bits multi octets, compatible avec l'ASCII |
| cp1252 | Jeux de caractères spécifique de Windows pour l'Europe occidentale. |
| BIG5 | Chinois traditionnel, principalement utilisé à Taiwan. |
| GB2312 | Chinois simplifié, officiel. |
| BIG5−HKSCS | Big5 avec les extensions de Hong Kong, chinois traditionnel. |
| Shift_JIS | Japonais |
| EUCJP | Japonais |

| Note |
|---|
| Les autres jeux de caractères ne sont pas reconnus, et le ISO−8859−1 sera utilisé à la place. |

Si vous voulez réaliser l'opération inverse, vous devez utiliser la fonction html_entity_decode .

Exemple avec htmlentities

```
<?php
$str = "Un 'apostrophe' en <b>gras</b>";

// Affiche : Un 'apostrophe' en <b>gras</b>
echo htmlentities($str);

// Affiche : Un &#039;apostrophe&#039; en <b>gras</b>
echo htmlentities($str, ENT_QUOTES);
?>
```

Voir aussi html_entity_decode , get_html_translation_table , htmlspecialchars , nl2br et urlencode .

## 6.36.24 htmlspecialchars() : Convertit les caractères spéciaux en entités HTML

string **htmlspecialchars** ( string string , *int* *quote_style* , *string* *charset* )

Certains caractères ont des significations spéciales en HTML, et doivent être remplacés par des

entités HTML pour être affichés. htmlspecialchars remplace tous ces caractères par leur équivalent dans la chaîne string . Cette conversion est très pratique pour la programmation web. Si vous devez remplacer tous les caractères, utilisez plutôt htmlentities à la place.

htmlspecialchars est pratique pour éviter que des données fournies par les utilisateurs contiennent des balises HTML, comme pour un forum ou un chat. Cette fonction prend un deuxième argument optionnel, qui indique comment doivent être traités les guillemets doubles et simples. Vous pouvez utiliser l'une des constantes suivantes : ENT_COMPAT , la constante par défaut, va convertir les guillemets doubles, et ignorer les guillemets simples; ENT_QUOTES va convertir les guillemets doubles et les guillemets simples; ENT_NOQUOTES va ignorer les guillemets doubles et les guillemets simples.

Les remplacements effectués sont :

- '&' (ampersand) devient '&'
- '"' (guillemets doubles) devient '"' lorsque ENT_NOQUOTES n'est pas utilisé.
- "' (single quote) devient ''' uniquement lorsque ENT_QUOTES est utilisé.
- '<' (supérieur à) devient '<'
- '>' (inférieur à) devient '>'

Exemple avec htmlspecialchars

```
 <?php
$new = htmlspecialchars("<a href='test'>Test</a>", ENT_QUOTES);
echo $new;
 // <a href=&#039;test&#039;>Test</a>
?>
```

Notez que cette fonction ne fait aucun autre remplacement que ceux qui sont listés ci−dessus. Pour faire un remplacement total, voyez plutôt htmlentities . Le support pour le second argument optionnel a été ajouté en PHP 3.0.17 et PHP 4.0.3.

cette fonction prend un troisième argument optionnel qui définit le jeux de caractères utilisé durant la conversion. Le support de cet argument a été ajouté en PHP 4.1.0.

Les jeux de caractères suivantes sont disponibles et supportés par PHP 4.3.0 et plus récent.

| Jeux de caractères | Description |
|---|---|
| ISO−8859−1 | Europe occidentale, Latin−1 |
| ISO−8859−15 | Europe occidentale, Latin−9. Dispose du signe Euro, des caractères spéciaux français et finlandais, qui manque au Latin−1(ISO−8859−1). |
| UTF−8 | Unicode 8 bits multi octets, compatible avec l'ASCII |
| cp1252 | Jeux de caractères spécifique de Windows pour l'Europe occidentale. |
| BIG5 | Chinois traditionnel, principalement utilisé à Taiwan. |
| GB2312 | Chinois simplifié, officiel. |
| BIG5−HKSCS | Big5 avec les extensions de Hong Kong, chinois traditionnel. |
| Shift_JIS | Japonais |
| EUCJP | Japonais |

Note

Les autres jeux de caractères ne sont pas reconnus, et le ISO−8859−1 sera utilisé à la place.

Voir aussi get_html_translation_table , htmlentities et nl2br .

## 6.36.25 implode() : Rassemble les éléments d'un tableau en une chaîne

string **implode** ( string glue , array pieces )

**implode** retourne une chaîne contenant la représentation en chaîne de caractères de tous les éléments du tableau pieces , dans le même ordre, avec la chaîne glue , placée entre deux

éléments.

Exemple avec implode

```
<?php
 $array = array('nom', 'email', 'telephone');
 $comma_separated = implode(",", $array);

 print $comma_separated;
  // nom,email,telephone
?>
```

Note

implode peut, pour des raisons historiques, accepter les paramètres dans un sens ou dans l'autre. Pour des raisons de cohérence avec explode , toutefois, il est préférable d'utiliser l'ordre des arguments tels que documenté.

Note

Depuis PHP 4.3.0, le paramètre the glue de implode est optionnel, et par défaut, il vaut la chaîne vide '' .

Voir aussi explode et split .

## 6.36.26 join() : Rassemble les éléments d'un tableau en une chaîne

string **join** ( string glue , array pieces )

join est un alias de implode .

Voir aussi explode , implode et split .

## 6.36.27 levenshtein() : Calcule la distance Levenshtein entre deux chaînes

int **levenshtein** ( string str1 , string str2 ) int **levenshtein** ( string str1 , string str2 , int cost_ins , int cost_rep , int cost_del ) int **levenshtein** ( string str1 , string str2 , function cost )

levenshtein calcule la distance Levenshtein entre deux chaînes de caractères. Elle retournera −1 si l'un des deux arguments contient plus de 255 caractères (cela devrait être plus que suffisant pour faire des comparaisons dans un dictionnaire ou annuaire, et personne de sérieux ne fera de comparaison génétique en PHP ).

La distance Levenshtein est définie comme le nombre minimal de caractères qu'il faut remplacer, insérer ou modifier pour transformer la chaîne str1 en str2 . La complexité de l'algorithme est en O(m*n) , où n et m sont les tailles respectives de str1 et str2 : c'est plutôt bien, en comparaison de similar_text , qui est en O(max(n,m)**3) , mais cela reste très coûteux.

Dans sa forme la plus simple, levenshtein va prendre uniquement deux chaînes de caractères comme paramètres, et calculer simplement le nombre d'insertions, de remplacements et d'effacements nécessaires pour tranformer str1 en str2 .

La deuxième variante de la fonction prend trois paramètres supplémentaires qui représentent les coûts d'insertions, de remplacements et d'effacements. C'est une version plus générale de la première fonction, mais qui est un peu moins efficace.

La troisième variante (qui n'est pas implémentée actuellement), est la version la plus générale, mais la plus lente. Elle appelera une fonction utilisateur qui déterminera le coût de chaque opération.

La fonction utilisateur qui sera appelée reçoit les arguments suivants :

- Opération à réaliser : 'I', 'R' ou 'D'
- Caractère dans str1
- Caractère dans str2
- Position dans str1
- Position dans str2
- Caractères restants dans str1
- Caractères restants dans str2

Cette fonction doit retourner un entier positif, qui représente le coût de cette opération particulière. Il peut ne prendre en compte que certains des paramètres fournis.

Grâce à cette fonction utilisateur, il est possible de prendre en compte la pertinence ou la valeur des caractères eux−mêmes, ou encore le contexte, pour définir le coûts d'une insertion, d'un effacement ou d'un remplacement. Cela se fait en perdant toutes les optimisations faîtes en terme d'exploitation du CPU et des buffers.

Voir aussi soundex , similar_text et metaphone .

## 6.36.28 localeconv() : Lit la configuration locale

array **localeconv** ( void )

localeconv retourne un tableau associatif contenant les informations de formats localisée pour les nombres et la monnaie.

localeconv retourne les formats en fonction de la configuration effectuée avec setlocale . Le tableau associatif qui est retourné contient les index suivants :

| Index du tableau | Description |
|---|---|
| decimal_point | Séparateur décimal |
| thousands_sep | Séparateur de milliers |
| grouping | Tableau contenant les regroupements numériques |
| int_curr_symbol | Symbole monétaire international (i.e. EUR) |
| currency_symbol | Symbole monétaire local (i.e. €) |
| mon_decimal_point | Séparateur décimal monétaire |
| mon_thousands_sep | Séparateur de milliers monétaires |
| mon_grouping | Tableau contenant les regroupements numériques monétaires |
| positive_sign | Signe des valeurs positives |
| negative_sign | Signe des valeurs négatives |
| int_frac_digits | Nombre international de décimales |
| frac_digits | Nombre local de décimales |
| p_cs_precedes | TRUE si le currency_symbol précède une valeur positive et FALSE s'il lui succède. |
| p_sep_by_space | TRUE si un espace sépare currency_symbol d'une valeur positive, et FALSE sinon. |
| n_cs_precedes | TRUE si currency_symbol précède une valeur négative, et FALSE s'il lui succède. |
| n_sep_by_space | TRUE si un espace sépare currency_symbol d'une valeur négative, et FALSE sinon. |
| p_sign_posn | • 0<br>• Des parenthèses entourent la valeur et le symbole monétaire<br>• 1<br>• Le signe précède la valeur et le symbole monétaire<br>• 2<br>• Le signe suit la valeur et le symbole monétaire<br>• 3<br>• Le signe précède immédiatement la valeur et le symbole monétaire<br>• 4<br>• Le signe suit immédiatement la valeur et le symbole monétaire |
| n_sign_posn | • 0 |

| | |
|---|---|
| | • Des parenthèses entourent la valeur et le symbole monétaire<br>• 1<br>• Le signe précède la valeur et le symbole monétaire<br>• 2<br>• Le signe suit la valeur et le symbole monétaire<br>• 3<br>• Le signe précède immédiatement la valeur et le symbole monétaire<br>• 4<br>• Le signe suit immédiatement la valeur et le symbole monétaire |

Les champs de regroupements contiennent des tableaux qui définissent la manière dont les nombres doivent être regroupés. Par exemple, le champ de regroupement pour en_US contiendra deux éléments, avec les valeurs 3 et 3 . Si un élément de tableau contient CHAR_MAX , aucun autre regroupement n'est fait. Si un élément de tableau contient 3 , l'élément précédent doit être utilisé.

Exemple avec localeconv

```php
<?php
setlocale(LC_ALL, "en_US");

$locale_info = localeconv();

echo "<PRE>\n";
echo "-----------------------------------------------------------\n";
echo "  Informations monétaires pour la configuration locale : \n";
echo "-----------------------------------------------------------\n";

echo "int_curr_symbol:   {$locale_info["int_curr_symbol"]}\n";
echo "currency_symbol:   {$locale_info["currency_symbol"]}\n";
echo "mon_decimal_point: {$locale_info["mon_decimal_point"]}\n";
echo "mon_thousands_sep: {$locale_info["mon_thousands_sep"]}\n";
echo "positive_sign:     {$locale_info["positive_sign"]}\n";
echo "negative_sign:     {$locale_info["negative_sign"]}\n";
echo "int_frac_digits:   {$locale_info["int_frac_digits"]}\n";
echo "frac_digits:       {$locale_info["frac_digits"]}\n";
echo "p_cs_precedes:     {$locale_info["p_cs_precedes"]}\n";
echo "p_sep_by_space:    {$locale_info["p_sep_by_space"]}\n";
echo "n_cs_precedes:     {$locale_info["n_cs_precedes"]}\n";
echo "n_sep_by_space:    {$locale_info["n_sep_by_space"]}\n";
echo "p_sign_posn:       {$locale_info["p_sign_posn"]}\n";
echo "n_sign_posn:       {$locale_info["n_sign_posn"]}\n";
echo "</PRE>\n";
?>
```

La constante CHAR_MAX est aussi définie pour être utilisée comme indiqué ci−dessus.

Voir aussi setlocale .

## 6.36.29 ltrim() : Supprime les caractères invisibles de début de chaîne

string **ltrim** ( string str , *string charlist* )

ltrim retourne la chaîne str , après avoir supprimé les caractères invisibles de début de chaîne. Si le second paramètre charlist a été omis, ltrim supprimera les caractères suivants :

- " " ( ASCII 32 ( 0x20 )), un espace ordinaire.
- " " ( ASCII 9 ( 0x09 )), une tabulation.
- "\n" ( ASCII 10 ( 0x0A )), une nouvelle ligne (line feed).
- "\r" ( ASCII 13 ( 0x0D )), un retour chariot (carriage return).
- "\0" ( ASCII 0 ( 0x00 )), le caractère NUL .
- "\x0B" ( ASCII 11 ( 0x0B )), une tabulation verticale.

Il est aussi possible de spécifier les caractères à supprimer en utilisant le paramètre charlist . Listez simplement les caractères que vous voulez supprimer dans ce paramètre. Avec .. , vous pourrez spécifier des intervalles de caractères.

Exemple avec ltrim

```
<?php

$text = "        Voici quelques mots :) ...  ";
$trimmed = ltrim($text);
// $trimmed = "Voici quelques mots :) ...  "
$trimmed = ltrim($text,"     .");
// $trimmed = "Voici quelques mots :) ...  "
$clean = ltrim($binary,"\0x00..\0x1F");
// supprime tous les caractères de contrôle ASCII au début de la chaîne de caractères.
// (de 0 à 31 inclus)
?>
```

| Note |
| --- |
| Le second paramètre a été ajouté en PHP 4.1.0. |

Voir aussi trim et rtrim .

## 6.36.30 md5_file() : Calcule le md5 d'un fichier

string **md5_file** ( string filename , *bool raw_output* )

md5_file calcule le MD5 du fichier filename en utilisant l'algorithme RSA Data Security, Inc. MD5 Message−Digest Algorithm , puis retourne la valeur ainsi calculée. Le résultat est un nombre de 32 caractères hexadécimaux. Si le paramètre optionnel raw_output vaut TRUE , le digest MD5 est retourné au format binaire brut, avec une taille de 16.

| Note |
| --- |
| Le paramètre optionnel raw_output a été ajouté en PHP 5.0.0 et vaut par défaut FALSE . |

md5_file a la même utilité que la commande en ligne md5sum.

Voir aussi md5 , crc32 et sha1_file .

## 6.36.31 md5() : Calcule le md5 d'une chaîne

string **md5** ( string str , *bool raw_output* )

**md5** calcule le MD5 de la chaîne de caractères str en utilisant l'algorithme RSA Data Security, Inc. MD5 Message−Digest Algorithm , et retourne le résultat. Le résultat est un nombre de 32 caractères hexadécimaux.

| Note |
| --- |
| Le paramètre raw_output a été ajouté en PHP 5.0.0 et vaut par défaut FALSE . |

Exemple avec md5

```
<?php
$str = 'pomme';

if (md5($str) === 'ede0f9c3a1d2093e3f48fcafd3c70915') {
    echo "Voulez vous une golden ou une spartan?";
    exit;
}
?>
```

Voir aussi crc32 , md5_file et sha1 .

## 6.36.32 metaphone() : Calcule la clé métaphone

string **metaphone** ( string str )

metaphone calcule la clé metaphone de la chaîne de caractères str .

metaphone est similaire à la fonction soundex : elle créée une clé similaire pour des mots dont la prononciation est proche. C'est une fonction qui est plus précise que soundex car elle prend en compte la prononciation anglaise. La clé metaphone générée est de taille variable.

metaphone fonctionne en anglais, et n'utilise pas les spécificités de la langue française. Les résultats sur des termes français peuvent être des plus aléatoires.

Metaphone a été développé par Lawrence Philips <lphilips@verity.com>. Cette méthode est décrite dans le livre ["Practical Algorithms for Programmers", Binstock & Rex, Addison Wesley, 1995].

## 6.36.33 money_format() : Met un nombre au format monétaire

string **money_format** ( string format , float number )

money_format retourne le nombre number avec le format financier. Cette fonction fait l'interface avec la fonction strmon() de la librairie C, à la différence près que cette implémentation ne convertit qu'un nombre à la fois.

Le paramètre de format est constitué de la séquence suivante :

- un caractère %
- une configuration optionnelle
- une taille de champ optionnelle
- une précision à gauche optionnelle
- une précision à droite optionnelle
- un caractère de conversion obligatoire

Configuration

Une ou plusieurs des configurations suivantes sont utilisables :

*= f*
: Le caractère = suivi par un octet unique f qui sera utilisé comme caractère de remplissage. Le caractère de remplissage par défaut est espace.

^
: Désactive le groupage de caractères (tel que définit dans la configuration locale).

*+ ou (*
: Specifie le style de formatage pour lesnombres positifs et négatifs. Si + est utilisé, les équivalents dans la configuration locale de + et − seront utilisés. Si ( est utilisé, les sommes négatives seront placées entre parenthèses. Si aucune spécification n'est fournie, la valeur par défaut est + .

*!*
: Supprime le simple monétaire dans la chaîne finale.

–
: Si fourni, cette configuration fait que les champs seront justifiés à gauche (complétés à droite), au contraire de la configuration par défaut qui est justifiée à droite, et complétée à gauche.

Taille de champs

*w*

Un nombre décimal qui spécifie la taille minimale du champ. Le champ sera complété à gauche, à moins que la configuration − ne soit utilisée. Par défaut, cette valeur est de 0.

Précision à gauche

*# n*

Le nombre maximal de chiffres ( n ) attendus à gauche du séparateur décimal (e.g. la virgule). Cette option est généralement utilisée pour conserver l'alignement de colonnes de nombres, en utilisant un caractère pour compléter le nombre si ce dernier a moins de n chiffres. Si le nombre réel de chiffres est plus grand que n , cette specification est ignorée.

Si le groupage n'a pas été supprimé via la configuration ^ , les séparateurs de groupage seront insérés avant le caractère de remplissage (le cas échéant). Les séparateurs ne seront pas appliqués aux caractères de remplissage, même si ce caractère est un nombre.

Pour s'assurer de l'alignement, tous les caractères apparaissant avant et après le nombre formaté, tels que les symboles monétaires ou les signes négatif et positif, seront placés au même endroit grâce à des espaces supplémentaires, afin que toutes les tailles des nombres soient les mêmes.

Précision à droite

*. p*

Un point suivi par un nombre de décimales ( p ). Si la valeur de p est 0 (zéro), le séparateur décimal et les décimales seront supprimées. Si aucune précision à droite n'est précisée, la valeur par défaut sera lue dans la configuration locale. Le nombre formaté sera alors arrondi pour satisfaire les contraintes d'affichage.

Caractères de conversion

*i*

Le nombre est formaté suivant le format monétaire international de la configuration locale (e.g. pour la France : 1 234,56 F).

*n*

Le nomre est formaté en fonction du format monétaire national (e.g. pour la configuation de_DE : DM1.234,56).

*%*

Retourne le caractère % .

| Note |
| --- |
| La catégorie LC_MONETARY de la configuration locale affecte le comportement de cette fonction. Utilisez setlocale pour configurer correctement PHP avant d'utiliser cette fonction.<br><br>Les caractères placés avant et après la chaîne de formattage seront laissés intacts. |

| Exemple avec money_format |
| --- |
| Voici plusieurs exemples d'utilisation de la fonction money_format avec différentes chaînes de formatage, et configurations locales. |

```php
<?php

   $number = 1234.56;

   // Affichons ce nombre au format international pour en_US
   setlocale(LC_MONETARY, 'en_US');
   echo money_format('%i', $number)."\n";
   // USD 1,234.56

   // Et au format italien national avec 2 decimales
   setlocale(LC_MONETARY, 'it_IT');
   echo money_format('%.2n', $number)."\n";
   // L. 1.234,56

   // Utilisation d'un nombre négatif
   $number = -1234.5672;

   // Format US national, avec les parenthèeses pour les nombres négatifs
   // et 10 chiffres de précision à gauche
   setlocale(LC_MONETARY, 'en_US');
   echo money_format('%(#10n', $number)."\n";
   // ($        1,234.57)

   // Format similaire au précédent, en ajoutant 2 décimales
   // pour la précision à droite, et en utilisant le caractère de remplissage '*'
   echo money_format('%=*(#10.2n', $number)."\n";
   // ($********1,234.57)

   // Utilisons maintenant la justification à gauche, avec un champ de 14 caractères
   // de longs, sans groupage de chiffres, et en utilisant le format international
   // pour de_DE
   setlocale(LC_MONETARY, 'de_DE');
   echo money_format('%=*^-14#8.2i', 1234.56)."\n";
   // DEM 1234,56****

   // Ajoutons encore à l'exemple précédent
    setlocale(LC_MONETARY, 'en_GB');
   $fmt = 'La valeur finale est %i (après 10% de remise)';
   echo money_format($fmt, 1234.56)."\n";
   // La valeur finale est GBP 1,234.56 (après 10% de remise)

?>
```

Voir aussi setlocale , number_format , sprintf , printf et sscanf .

## 6.36.34 nl_langinfo() : Rassemble des informations sur la langue et la configuration locale.

string **nl_langinfo** ( int item )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

## 6.36.35 nl2br() : Insère un retour à la ligne HTML à chaque nouvelle ligne

string **nl2br** ( string string )

nl2br retourne string après avoir inséré '<br />' devant toutes les nouvelles lignes.

| Note |
| --- |
| Depuis PHP 4.0.5, nl2br est désormais compatible XHTML. Toutes les versions antérieures à 4.0.5 retourneront l'argument string avec '<br>' inséré devant les nouvelles lignes, au lieu de '<br />'. |

Voir aussi htmlspecialchars , htmlentities , wordwrap et str_replace .

## 6.36.36 number_format() : Formate un nombre pour l'affichage

string **number_format** ( float number , *int decimals* ) string **number_format** ( float number , int decimals , string dec_point , string thousands_sep )

number_format retourne une chaîne représentant number formaté. number_format accepte un, deux ou 4 paramètres (mais pas trois).

Si seul le paramètre number est donné, il sera formaté sans partie décimale, mais avec une virgule entre chaque millier.

Si les deux paramètres number et decimals sont fournis, number sera formaté avec decimals décimales, un point (".") comme séparateur décimal et une virgule entre chaque millier.

Avec quatre paramètres, number sera formaté avec decimals décimales, dec_point comme séparateur décimal, et thousands_sep comme séparateur de milliers.

| Note |
| --- |
| Seul le premier caractère du paramètre thousands_sep est utilisé. Par exemple, si vous utilisez foo comme séparateur de milliers, sur le nombre 1000 , number_format retournera 1f000 . |

En notation française, on utilise généralement deux chiffres après la virgule, une virgule comme séparateur décimal, et un espace comme séparateur de milliers. Cela donne :

Exemple avec number_format

```
<?php
  $nombre = 1234.56;
// Notation anglaise (par défaut)
  $english_format_number = number_format($nombre);
// 1,234.56
// Notation française
  $nombre_format_francais = number_format($nombre, 2, ',', ' ');
// 1 234,56
?>
```

Voir aussi sprintf , printf et sscanf .

## 6.36.37 ord() : Retourne le code ASCII d'un caractère

int **ord** ( string string )

**ord** retourne le code ASCII du premier caractère de string . Cette fonction est le contraire de chr .

Exemple avec ord

```
<?php
  if (ord($str) == 10) {
    echo "Le premier caractère de \$str est une nouvelle ligne\n";
  }
?>
```

Vous pouvez trouver une table des codes ASCII ici : http://www.asciitable.com .

Voir aussi chr .

## 6.36.38 parse_str() : Analyse une requête HTTP

void **parse_str** ( string str , *array arr* )

parse_str analyse la chaîne de caractères str comme s'il s'agissait d'une requête HTTP, passée via l'URL, en méthode GET ou POST. Toutes les variables qu'elle y repère sont alors créées, avec leurs valeurs respectives. Si le second paramètre arr est fourni, les variables y seront stockées, sous forme d'index de tableau.

| Note |
| --- |
| Le support du paramètre optionnel arr a été ajouté en PHP 4.0.3. |

| Note |
| --- |
| Pour accéder à l'URL appelante **QUERY_STRING** , vous devez utiliser la variable $_SERVER['QUERY_STRING'] . Il est aussi intéressant de lire la section sur les variables externes à PHP . |

Exemple avec parse_str

```
 <?php
$str = "first=value&arr[]=foo+bar&arr[]=baz";
parse_str($str);
echo $first;  // value
echo $arr[0]; // foo bar
echo $arr[1]; // baz

parse_str($str, $output);
echo $output['first'];  // value
echo $output['arr'][0]; // foo bar
echo $output['arr'][1]; // baz
?>
```

Voir aussi parse_url , pathinfo , set_magic_quotes_runtime et urldecode .

## 6.36.39 print() : Affiche une chaîne de caractères

int **print** ( string arg )

**print** affiche la chaîne de caractères arg . **print** retourne toujours 1 .

print n'est pas vraiment une fonction (c'est techniquement une structure du langage). Cela fait que vous n'êtes pas obligés d'utiliser des parenthèses.

Exemples avec print

```
 <?php
  print("Bonjour le monde");

  print "print() fonctionne aussi sans les parenthèses.";

  print "Ce print() se
répartit sur plusieurs lignes. Il affiche aussi les
nouvelles lignes";

  print "Ce print() se\nrépartit sur plusieurs lignes. Il affiche aussi les\nnouvelles lignes";

  print "L'échappement de caractères se fait : \"comme ceci\".";

// Vous pouvez utiliser des variables avec print
$foo = "foobar";
$bar = "barbaz";

print "foo vaut $foo"; // foo vaut foobar

// Vous pouvez aussi utiliser des tableaux
$bar = array("clef" => "foo");

print "c'est {$bar['clef']} !"; // c'est foo !

// Les guillemets simples annulent le remplacement des variables
print 'foo vaut $foo'; // foo vaut $foo

print <<<END
Cette syntaxe s'intitule le "here document" et
 permet d'afficher plusieurs lignes avec de
 l'interpolation de variables. Notez que la fin de
la syntaxe doit apparaître sur une nouvelle ligne,
avec uniquement un point-virgule, et pas d'espace
```

```
de plus !
END;
?>
```

Pour une brève discussion à propos des différences entre print et echo , consultez cet article des FAQTs Knowledge Base : http://www.faqts.com/knowledge_base/view.phtml/aid/1/fid/40

| Note |
| --- |
| Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions . |

Voir aussi echo , printf et flush .

## 6.36.40 printf() : Affiche une chaîne de caractères formattée

void **printf** ( string format , *mixed* *args* )

printf affiche les arguments args formatés sous forme de chaîne de caractères grâce à args . Les options de formatage sont documentées avec la fonction sprintf .

Voir aussi print , sprintf , sscanf , fscanf et flush .

## 6.36.41 quoted_printable_decode() : Convertit une chaîne quoted−printable en chaîne 8 bits

string **quoted_printable_decode** ( string str )

**quoted_printable_decode** retourne la chaîne de caractères str , après l'avoir convertie du format quoted printable en format binaire 8 bits. Cette fonction est similaire à imap_qprint , hormis le fait qu'elle n'exige pas le module IMAP, pour fonctionner.

## 6.36.42 quotemeta() : Echappe les méta−caractères

string **quotemeta** ( string str )

**quotemeta** retourne la chaîne str après avoir introduit un anti−slash ( \ ) devant tous les caractères suivants : . \\ + * ? [ ^ ] ( $ )

Voir aussi addslashes , htmlentities , htmlspecialchars , nl2br et stripslashes .

## 6.36.43 rtrim() : Supprime les espaces de fin de chaîne

string **rtrim** ( string str , *string* *charlist* )

rtrim retourne la chaîne str , après avoir supprimé tous les caractères blancs de fin de chaîne. Appelée sans le second paramètre, rtrim supprimera les caractères suivants :

- " " ( ASCII 32 ( 0x20 )), un espace ordinaire.
- "\t" ( ASCII 9 ( 0x09 )), une tabulation.
- "\n" ( ASCII 10 ( 0x0A )), une nouvelle ligne (line feed).

- "\r" ( ASCII 13 ( 0x0D )), un retour chariot (carriage return).
- "\0" ( ASCII 0 ( 0x00 )), le caractère NUL .
- "\x0B" ( ASCII 11 ( 0x0B )), une tabulation verticale.

Il est aussi possible de spécifier les caractères à supprimer en utilisant le paramètre charlist . Listez simplement les caractères que vous voulez supprimer dans ce paramètre. Avec .. , vous pourrez spécifier des intervalles de caractères.

Exemple avec rtrim

```
<?php

$text = "        Voici quelques mots :) ...  ";
$trimmed = rtrim($text);
// $trimmed = "        Voici quelques mots :) ..."
$trimmed = rtrim($text,"     .");
// $trimmed = "        Voici quelques mots :)"
$clean = rtrim($binary,"\0x00..\0x1F");
// supprime tous les caractères de contrôle ASCII au début de la chaîne de caractères.
// (de 0 à 31 inclus)
?>
```

Note

Le second paramètre a été ajouté en PHP 4.1.0.

Voir aussi trim et ltrim .

## 6.36.44 setlocale() : Modifie les informations de localisation

string **setlocale** ( mixed category , string locale , *string ...* ) string **setlocale** ( mixed category , array locale )

setlocale retourne la nouvelle configuration locale, ou FALSE si la localisation n'est pas implémentée sur votre plate−forme, si la variable de localisation n'existe pas, ou si la catégorie spécifiée n'est pas valide. Un nom de catégorie invalide générera un message d'alerte.

category est une constante (ou une chaîne) qui spécifie la catégorie de fonction affectées par la configuration de localisation :

- LC_ALL pour toutes les constantes suivantes
- LC_COLLATE pour la comparaison de chaînes de caractères. Voir strcoll
- LC_CTYPE pour la classification et la conversion de caractères. Voir strtoupper
- LC_MONETARY pour localeconv
- LC_NUMERIC pour le spéarateur décimal. Voir localeconv
- LC_TIME pour le format de date et d'heure avec strftime

Si locale est la chaîne vide "" , les noms de locales seront pris dans l'environnement, à partir des variables de même noms que les catégories ci−dessus, ou depuis "LANG".

Si locale est NULL ou vaut "0" , la configuration locale ne sera pas modifiée, et la configuration courante sera retournée.

Note

La possibilité de passer plusieurs locales est disponible depuis PHP 4.3.0.

setlocale retourne la valeur de la locale courante, ou bien FALSE si la fonctionnalité de localisation n'est pas implémentée sur votre plate−forme, si la locale spécifiée n'existe pas, ou si la catégorie n'existe pas. Une catégorie invalide causera aussi l'affichage d'un message d'alerte.

Si locale vaut zéro ou "0" , la configuration de localisation ne sera pas modifiée, et seule la valeur courante sera retournée.

| Note |
| --- |
| Passer plusieurs locales n'était pas disponible avant PHP 4.3.0. |

setlocale retourne la nouvelle valeur locale, ou bien FALSE si la fonctionnalité n'est pas implantée sur votre plate forme, que la configuration spécifiée n'existe pas ou que le nom de catégorie est invalide. Un nom de catégorie invalide causera aussi l'affichage d'une alerte. Les noms de catégories et de configurations locales sont disponibles dans la RFC 1766 et ISO 639 .

| Note |
| --- |
| La valeur retournée par setlocale dépend du système d'explotiation qui fait tourner PHP . La valeur retournée est exactement celle que la fonction setlocale() du système retourne. |

Array

Exemple avec setlocale

```
 <?php
    /* Configure le script en hollandais */
    setlocale (LC_ALL, 'nl_NL');

    /* Affiche : vrijdag 22 december 1978 */
    echo strftime ("%A %e %B %Y", mktime (0, 0, 0, 12, 22, 1978));

    /* Essai de différentes valeurs possible pour l'allemand depuis

<font face="courrier">PHP</font>
 4.3.0 */
       $loc_de = setlocale (LC_ALL, 'de_DE@euro', 'de_DE', 'de', 'ge');
    echo "L'identifiant de l'allemand sur ce système est '$loc_de'";
?>
```

Exemple avec setlocale pour Windows

```
 <?php
    /* Configure le script en hollandais */
    setlocale (LC_ALL, 'nld_nld');

     /* Affiche : vrijdag 22 december 1978 */
    echo strftime ("%A %d %B %Y", mktime (0, 0, 0, 12, 22, 1978));

     /* Essai de différentes valeurs possible pour l'allemand depuis

<font face="courrier">PHP</font>
 4.3.0 */
    $loc_de = setlocale (LC_ALL, 'de_DE@euro', 'de_DE', 'deu_deu');
       echo "L'identifiant de l'allemand sur ce système est '$loc_de'";
?>
```

## 6.36.45 sha1_file() : Calcule le sha1 d'un fichier

string **sha1_file** ( string filename , *bool raw_output* )

**sha1_file** calcule le sha1 du fichier filename en utilisant US Secure Hash Algorithm 1 , puis retourne ce sha1. Le sha1 est un nombre hexadécimal de 40 caractères. En cas d'erreur, FALSE est retourné.

Voir aussi sha1 , crc32 et md5_file

## 6.36.46 sha1() : Calcule le sha1 d'une chaîne de caractères

string **sha1** ( string str , *bool raw_output* )

sha1 calcule le sha1 de la chaîne de caractères str en utilisant US Secure Hash Algorithm 1 , puis retourne cette valeur. Un sha1 est un nombre hexadécimal de 40 caractères.

Exemple avec sha1

```
<?php
$str = 'pomme';

if (sha1($str) === '752c14ea195c460bac3c3b7896975ee9fd15eeb7') {
    echo "Voulez-vous une golden ou une spartan?";
    exit;
}
?>
```

Voir aussi sha1_file , crc32 et md5

## 6.36.47 similar_text() : Calcule la similarité de deux chaînes

int **similar_text** ( string first , string second , *float* *percent* )

similar_text calcule la similarité entre les deux chaînes first et second , selon la méthode d'Oliver [1993]. Notez que cette implémentation n'utilise pas la méthode de pile comme dans le pseudo code d'Oliver, mais des appels récursifs, ce qui accélère ou pas le processus. Notez que la complexité de l'algorithme est en O(N**3) où N est la taille de la plus grand chaîne.

Le troisième argument percent est une référence, où similar_text va calculer la similarité en pourcentage automatiquement. Sinon, elle retourne le nombre de caractères identiques dans les deux chaînes.

## 6.36.48 soundex() : Calcule la clé soundex

string **soundex** ( string str )

**soundex** calcule la clé soundex de la chaîne str .

La clé soundex possède la propriété qui fait que deux mots prononcés similairement auront la même clé soundex. Cette fonction est donc utilisée pour simplifier les recherches dans les bases de données, où vous connaissez la prononciation d'un mot ou nom, mais pas son orthographe exacte. Cette fonction retourne une chaîne de 4 caractères, commençant par une lettre.

L'implémentation de la fonction soundex de PHP a été décrite par Donald Knuth dans "The Art Of Computer Programming, vol. 3: Sorting And Searching", Addison−Wesley (1973), pp. 391−392.

Notez bien que soundex prend en compte la prononciation à l'anglaise, et non pas à la française. Les résultats pourront donc être très différents de ce que l'on peut attendre. Des versions en langue française de soundex existent, mais elles ne sont pas intégrées dans PHP .

Exemples avec soundex

```
<?php
 soundex("Euler") == soundex("Ellery") == 'E460';
 soundex("Gauss") == soundex("Ghosh") == 'G200';
 soundex("Hilbert") == soundex("Heilbronn") == 'H416';
 soundex("Knuth") == soundex("Kant") == 'K530';
 soundex("Lloyd") == soundex("Ladd") == 'L300';
 soundex("Lukasiewicz") == soundex("Lissajous") == 'L222';
?>
```

Voir aussi levenshtein , metaphone et similar_text .

## 6.36.49 sprintf() : Retourne une chaîne formattée

string **sprintf** ( string format , *mixed* *args* )

**sprintf** retourne une chaîne formatée, avec le format format et en utilisant les arguments args .

La chaîne de format est composée de zéro, une ou plusieurs directives : les caractères ordinaires (à l'exception de % ) qui sont copiés directement dans le résultat, et des **spécifications de conversion** , qui exploitent chacune un des arguments passé après la chaîne de format. Ces formats s'appliquent à sprintf et printf .

Chaque spécification de conversion est consistituée d'un signe de pourcentage ( % ), suivi d'un ou plusieurs des éléments suivants, dans cet ordre :

1. Un **remplisseur** optionnel qui indique quel caractère sera utilisé pour compléter le résultat jusqu'à la longueur requise. Ce peut être le caractère d'espace, ou le caractère 0 . Par défaut, le remplissage se fait avec des espaces. Un autre caractère de remplissage peut être spécifié en le préfixant avec un guillemet simple ( ' ) : voir les exemples ci−dessous.
2. Un **spécificateur d'alignement** qui indique si le résultat doit être aligné à gauche ou à droite. Par défaut, le résultat est aligné à gauche. Le caractère − fera que le résultat est aligné à droite.
3. Un nombre optionnel, **spécificateur de taille** indique le nombre minimum de caractères que cette conversion doit fournir en résultat.
4. Un **spécificateur de précision** qui indique le nombre de décimales qui doivent être affichées pour les nombres à virgule flottante. Cette option n'a pas d'effet pour les autres types que float . Une autre fonction pratique pour formatter les nombres à virgule flottante est number_format .
5. Un **spécificateur de type** qui indique le type avec lequel l'argument sera traité. Plusieurs types possibles :
   - % − un caractère de pourcentage litéral. Aucun argument n'est nécessaire.
   - b − l'argument est traité comme un entier, et présenté comme un nombre binaire.
   - c − l'argument est traité comme un entier, et présenté comme le caractère de code ASCII correspondant.
   - d − l'argument est traité comme un entier, et présenté comme un nombre décimal signé.
   - u − l'argument est traité comme un entier, et présenté comme un nombre décimal non signé.
   - f − l'argument est traité comme un nombre à virgule flottante (type float ), et présenté comme un nombre à virgule flottante.
   - o − l'argument est traité comme un entier, et présenté comme un nombre octal.
   - s − l'argument est traité et présenté comme une chaîne de caractères.
   - x − l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en minuscules).
   - X − l'argument est traité comme un entier, et présenté comme un nombre hexadécimal (les lettres en majuscules).

Depuis PHP 4.0.6, la chaîne de format supporte le numérotage et l'échange d'arguments. Par exemple :

Echange d'arguments

```
<?php
 $format = "Il y a %d singes dans le %s";
 printf($format,$num,$location);
?>
```

Ce exemple peut afficher par exemple : "Il y a 5 signes dans le bananier". Mais imaginez que la chaîne de format soit créée dans un script séparé (une librairie), ca nous devons internationaliser cet affichage. Nous aurons peut être à écrire :

Echange d'arguments (2)

```
<?php
 $format = "Le %s a %d singes";
 printf($format,$num,$location);
?>
```

Maintenant, nous avons un problème. L'ordre des arguments a été changé, et ne correspond pas à l'ordre des arguments dans le script PHP . Nous souhaitons laisser le code PHP tel quel, et simplement indiquer dans la chaîne de formattage l'ordre dans lequel les arguments doivent être repris. La chaîne de format peut être réécrite ainsi :

Echange d'arguments (3)

```
<?php
 $format = "Le %2\$s a %1\$d signes";
 printf($format,$num,$location);
?>
```

Un des avantages est que vous pouvez désormais exploiter plusieurs fois les arguments sans les répéter. Ainsi :

Echange d'arguments (4)

```
<?php
 $format = "Le %2\$s a %1\$d singes.
          C'est un beau %2\$s, avec %1\$d signes.";
 printf($format, $num, $location);
?>
```

Voir aussi printf , sscanf , fscanf et number_format .

## 6.36.50 sscanf() : Analyse une chaîne à l'aide d'un format

mixed **sscanf** ( string str , string format , *string var1* )

sscanf est l'inverse de la fonction printf . sscanf lit des données dans la chaîne str , et l'interprète en fonction du format format . Si seuls deux paramètres sont fournis, les valeurs trouvées seront retournées sous forme de tableau. Sinon, les valeurs seront placées dans les arguments var1 .

Tous les caractères blancs dans la chaîne format correspondent à un caractère blanc dans la chaîne str . Cela signifie que même une tabulation dans la chaîne de format peut correspondre à un simple espace dans la chaîne str .

Exemple avec sscanf

```
<?php
// Lecture d'un numéro de série
 $serial = sscanf("SN/2350001","SN/%d");
// et la date de fabrication
 $mandate = "January 01 2000";
 list($month, $day, $year) = sscanf($mandate,"%s %d %d");
 echo "Le produit $serial a été fabriqué le : $year-".substr($month,0,3)."-$day\n";
?>
```

Si des paramètres optionnels sont passés, sscanf retournera le nombre de valeurs assignées. Les paramètres optionnels doivent être passés par référence.

Exemple avec sscanf : utilisation des paramètres optionnels

```
<?php
// lit les informations d'auteur, et génère une entrée DocBook
 $auth = "24    Lewis Carroll";
 $n = sscanf($auth,"%d    %s %s", &$id, &$first, &$last);
 echo "<author id="$id">
   <firstname>$first</firstname>
   <surname>$last</surname>
</author>\n";
?>
```

Voir aussi fscanf , printf et sprintf .

## 6.36.51 str_ireplace() : Version insensible à la casse de str_replace

mixed **str_ireplace** ( mixed search , mixed replace , mixed subject , *int &count* )

str_ireplace retourne une chaîne de caractères ou un tableau dont toutes les occurences de search dans subject (en ignorant la casse), ont été remplacées par la valeur de replace . Si vous n'avez pas besoin de règles de recherche complexe, il est recommandé d'utiliser cette fonction qui est plus rapide que eregi_replace ou preg_replace , exploitant l'option i .

Si subject est un tableau, le remplacement est effectué sur chacun des éléments du sujet subject , et la valeur retournée est un aussi un tableau.

Si search et replace sont des tableaux, alors str_replace prendra une valeur de chaque tableau, et l'utilisera pour faire le remplacement dans subject . Si replace a moins de valeurs que search , alors une chaîne vide sera utilisée pour effectuer les remplacements. Si search est un tableau et que replace est une chaîne, alors la chaîne de remplacement sera utilisée pour chaque élément de search .

Exemple avec str_ireplace

```
<?php
 $bodytag = str_ireplace("%body%", "black", "<body text=%BODY%>");
?>
```

Cette fonction est compatible avec les données binaires.

Note

Depuis PHP 5.0, le nombre de valeurs de needle trouvées et remplacées seront retournées dans le paramètre count passé par référence.

Voir aussi str_replace , ereg_replace , preg_replace et strtr .

## 6.36.52 str_pad() : Complète une chaîne jusqu'à une taille donnée

string **str_pad** ( string input , int pad_length , *string pad_string* , *int pad_type* )

**str_pad** retourne la chaîne input , complétée à droite, à gauche ou dans les deux sens, avec la chaîne pad_string jusqu'à ce qu'elle atteigne la taille de pad_length . Si le paramètre optionnel pad_string est omis, input sera complétée avec des espaces. Sinon, cette chaîne sera utilisée.

L'argument optionnel pad_type peut être l'une des constantes suivantes : STR_PAD_RIGHT , STR_PAD_LEFT , ou STR_PAD_BOTH . Si pad_type n'est pas spécifié, il prend la valeur par défaut de STR_PAD_RIGHT .

Si la valeur de pad_length est négative, ou inférieure à la taille courante de la chaîne input , input est retournée inchangée.

Exemple avec str_pad

```
<?php
 $input = "Alien";
 print str_pad($input, 10);                      // affiche "Alien     "
 print str_pad($input, 10, "-=", STR_PAD_LEFT);  // affiche "-=-=-Alien"
 print str_pad($input, 10, "_", STR_PAD_BOTH);   // affiche "__Alien___"
?>
```

## 6.36.53 str_repeat() : Répéte une chaîne

string **str_repeat** ( string input , int multiplier )

**str_repeat** retourne la chaîne input_str répétée multiplier fois. multiplier doit être positif ou nul. Si multiplier vaut 0, la fonction retourne la chaîne vide.

Exemple avec str_repeat

```
<?php
 echo str_repeat("-=", 10);
?>
```

Ceci affichera "-=-=-=-=-=-=-=-=-=-=".

Voir aussi str_pad et substr_count .

## 6.36.54 str_replace() : Remplace toutes les occurrences dans une chaîne

mixed **str_replace** ( mixed search , mixed replace , mixed subject , *int &count* )

str_replace retourne une chaîne ou un tableau, dont toutes les occurences de search dans subject ont été remplacée par replace . Si vous n'avez pas besoin de règles de remplacement compliquées, utilisez cette fonction de préférence à ereg_replace et preg_replace .

En PHP 4.0.5 et plus récent, tous les paramètres de str_replace peuvent être des tableaux.

Si subject est un tableau, alors le remplacement se fera sur chaque élément de celui-ci, et la valeur retournée sera aussi un tableau.

Si search et replace sont des tableaux, alors str_replace prendra une valeur de chaque tableau, et l'utilisera pour faire le remplacement dans subject . Si replace a moins de valeurs que search , alors une chaîne vide sera utilisée pour effectuer les remplacements. Si search est un tableau et que replace est une chaîne, alors la chaîne de remplacement sera utilisée pour chaque élément de search .

Exemple avec str_replace

```
<?php
// Génère : <body text='black'>
echo $bodytag = str_replace("%body%", "black", "<body text='%body%'>");

// Génère : Bnjr l mnd
$voyelles = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
echo $consonnes = str_replace($voyelles, "", "Bonjour le monde");

// Génère : Vous devriez manger des pizzas, des bièrse et des gâteaux tous les jours.
$phrase  = "Vous devriez manger des fruits, des légumes et des fibres tous les jours.";
$regime = array("fruits", "légumes", "fibres");
$bonne_chere   = array("pizzas", "bières", "gâteaux");

echo $newphrase = str_replace($regime, $bonne_chere, $phrase);

// Utilisation du compteur d'occurences en PHP 5.0.0
$str = str_replace("a", "", "La disparition est un lipogramme en E.", $count);
echo $count; // 3
?>
```

str_replace est compatible avec les chaînes binaires.

Note

str_replace a été ajoutée en PHP 3.0.6, mais était buggée jusqu'en PHP 3.0.8.

| Note |
| --- |
| Cette fonction gère les chaînes binaires |

| Note |
| --- |
| Depuis PHP 5.0, le nombre de valeurs de search trouvées et remplacées seront retournées dans le paramètre count passé par référence. |

Voir aussi str_ireplace , substr_replace , ereg_replace , preg_replace et strtr .

## 6.36.55 str_rot13() : Effectue une transformation rot13

string **str_rot13** ( string str )

str_rot13 effectue un encodage ROT13 de la chaîne str et retourne le résultat. L'encodage ROT13 décale toutes les lettres de 13 dans l'alphabet, et laisse tous les autres caractères inchangés. L'encodage et le décodage est fait par la même fonction : passer le résultat de str_rot13 à nouveau comme argument, retournera la chaîne originale.

## 6.36.56 str_shuffle() : Mélange les caractères d'une chaîne de caractères

string **str_shuffle** ( string str )

str_shuffle mélange les caractères d'une chaîne de caractères. Une permutation de tous les caractères est créée.

Exemple str_shuffle

```
<?php
$str = 'abcdef';
$shuffled = str_shuffle($str);

// Cela va affichier une ligne comme par exemple : bfdaec
print $shuffled;
?>
```

Voir aussi shuffle et rand .

## 6.36.57 str_split() : Convertit une chaîne de caractères en tableau

array **str_split** ( string string , *int split_length* )

Convertit une chaîne de caractères en tableau. Si le paramètre optionnel split_length est spécifié, le tableau retourné sera découpé en sous−parties, chacune de taille split_length , sinon, chaque sous−partie aura une taille d'un caractère.

FALSE est retourné si split_length est inférieur à 1. Si la longueur de split_length est supérieure à celle de string , la chaîne entière est retournée dans le premier (et seul) élément du tableau.

Exemples d'utilisation de str_split

```
<?php

$str = "Salut Dave";

$arr1 = str_split($str);
$arr2 = str_split($str, 3);

print_r($arr1);
print_r($arr2);
```

```
/* L'affichage donnera :

Array
(
    [0] => S
    [1] => a
    [2] => l
    [3] => u
    [4] => t
    [5] =>
    [6] => D
    [7] => a
    [8] => v
    [9] => e
)

Array
(
    [0] => Sal
    [1] => ut
     [2] => Dav
    [3] => e
)
*/
?>
```

Exemples liés à str_split

```
 <?php

$str = "Salut Dave";

print $str{0};  // S
print $str{8};  // v

// Crée : array('S','a','l','u','t',' ','D','a','v','e')
$arr1 = preg_split('//', $str, -1, PREG_SPLIT_NO_EMPTY);

?>
```

Voir aussi preg_split , split , count_chars et str_word_count .

## 6.36.58 str_word_count() : Compte le nombre de mots utilisés dans une chaîne

mixed **str_word_count** ( string string , *int format* )

str_word_count compte le nombre de mots dans la chaîne string . Si le paramètre optionnel format n'est pas spécifié, alors la valeur retournée sera un entier, représentant le nombre de mots trouvés. Si format est spécifié, la valeur retournée sera un tableau, qui dépend du format format . Les valeurs possibles pour format et le tableau retourné sont :

- 1 − retourne un tableau contenant tous les mots trouvés dans la chaîne string .
- 2 − retourne un tableau associatif, où les clés sont représentent la position du mot en regard, dans la chaîne string .

Dans cette fonction, la notion de 'mot' est dépendant de la configuration de localisation. C'est une chaîne qui contient tous les caractères alphabétiques, et qui peut contenir, mais pas commencer par "'" and "−".

Exemples d'utilisation de str_word_count

```
 <?php

$str = "Salut l'ami, vous
         avez          une belle mine!";

$a   = str_word_count($str, 1);
$b   = str_word_count($str, 2);
$c   = str_word_count($str);

print_r($a);
print_r($b);
print $c;
```

```
/* Le résultat est :
 Array
(
    [0] => Salut
    [1] => l'ami
    [2] => vous
    [3] => avez
    [4] => une
    [5] => belle
    [6] => mine
)
Array
(
    [0] => Salut
    [6] => l'ami
    [13] => vous
    [27] => avez
    [41] => une
    [45] => belle
    [51] => mine
)
7

*/
?>
```

Voir aussi explode , preg_split , split , count_chars et substr_count .

## 6.36.59 strcasecmp() : Comparaison de chaînes binaires

int **strcasecmp** ( string str1 , string str2 )

strcasecmp retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est plus grande que str2 , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est insensible à la casse.

Exemple avec strcasecmp

```
<?php
 $var1 = "Hello";
 $var2 = "hello";
 if (strcasecmp($var1, $var2) == 0) {
   echo "$var1 est égale à $var2 (comparaison insensible à la casse)";
 }
?>
```

Vior aussi ereg , strcmp , substr , stristr , strncasecmp et strstr .

## 6.36.60 strchr() : Trouve la première occurence d'un caractère dans une chaîne

string **strchr** ( string haystack , string needle )

strchr est un alias de strstr .

## 6.36.61 strcmp() : Comparaison binaire de chaînes

int **strcmp** ( string str1 , string str2 )

strcmp retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est sensible à la casse.

Voir aussi ereg , strcasecmp , substr , stristr , strncasecmp , strncmp et strstr .

## 6.36.62 strcoll() : Comparaison de chaînes localisée

int **strcoll** ( string str1 , string str2 )

strcoll retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales. Si la configuration de localisation courante est C ou POSIX, cette fonction est équivalente à strcmp .

Notez que cette comparaison est sensible à la casse, et que contrairement à strcmp , elle n'est pas compatible avec les chaînes binaires.

Voir aussi ereg , strcmp , strcasecmp , substr , stristr , strncasecmp , strncmp , strstr et setlocale .

## 6.36.63 strcspn() : Trouve un segment de chaîne ne contenant pas certains caractères

int **strcspn** ( string str1 , string str2 )

strcspn retourne la taille du premier segment de str1 qui ne contient **aucun** des caractères de str2 .

Voir aussi strspn .

## 6.36.64 strip_tags() : Supprime les balises HTML et PHP d'une chaîne

string **strip_tags** ( string str , *string allowable_tags* )

strip_tags retourne la chaîne str après avoir supprimé toutes les balises PHP et HTML du code. Elle génère des alertes si les balises sont incomplètes ou erronées. Elle utilise le même moteur de recherche que fgetss .

Le paramètre optionnel allowable_tags permet d'indiquer les balises qui doivent être conservées.

| Note |
| --- |
| allowable_tags a été ajouté en PHP 3.0.13 et PHP 4.0b3. |

Exemple avec strip_tags

```
<?php
 $string = strip_tags($string, '<a><b><i><u>');
?>
```

| Attention |
| --- |
| strip_tags ne modifie pas les attributs des balises que vous autorisez via le paramètre allowable_tags , y compris le style et les onmouseover , que des utilisateurs mal intentionnés peuvent utiliser. |

## 6.36.65 stripcslashes() : Supprime les anti−slash d'une chaîne C

string **stripcslashes** ( string str )

stripcslashes retourne la chaîne str après avoir supprimé tous les anti−slash. stripcslashes respecte les séquences spéciales du C, telles que \n , \r ..., les nombres octaux et hexadécimaux.

Voir aussi addcslashes .

## 6.36.66 stripos() : Rercherche la position d'une occurence dans une chaîne, sans tenir compte de la casse

int **stripos** ( string haystack , string needle , *int offset* )

stripos retourne la position numérique de la première occurence de needle dans la chaîne haystack . Contrairement à strpos , stripos est insensible à la casse. Et, contrairement à strrpos , cette fonction peut prendre une chaîne complète comme paramètre needle et cette chaîne complète sera utilisée pour la recherche.

Si le paramètre needle n'est pas trouvé, strpos retournera FALSE .

| Attention |
| --- |
| Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur exacte de cette fonction. |

Exemples avec stripos

```
 <?php
$findme    = 'a';
$mystring1 = 'xyz';
$mystring2 = 'ABC';

$pos1 = stripos($mystring1, $findme);
$pos2 = stripos($mystring2, $findme);

// Non, 'a' ne fait pas partie de 'xyz'
if ($pos1 === false) {
    echo "La chaîne '$findme' n'a pas été trouvée dans la chaîne '$mystring'";
}

// Notez l'utilisation de ===.  Un simple == ne donnerait pas le résultat escompté
// car la lettre 'a' est à la position 0th (la première).
if ($pos2 !== false) {
    echo "La chaîne '$findme' a été trouvée dans la chaîne '$mystring'";
    echo " et à la position $pos";
}
?>
```

Si needle n'est pas une chaîne, elle est convertie en entier, et utilisé comme caractère de code ASCII correspondant.

Le paramètre optionnel offset vous permet de spécifier à partir de quel caractère dans haystack vous souhaitez commencer la recherche. La position retournée sera toujours relative au début de la chaîne haystack .

Voir aussi strpos , strrpos , strrchr , substr , stristr , strstr et str_ireplace .

## 6.36.67 stripslashes() : Supprimer les anti−slash d'une chaîne

string **stripslashes** ( string str )

stripslashes retourne une chaîne dont les anti−slash on été supprimés. \' devient ' , etc... Les doubles anti−slash sont réduits à un seul anti−slash. stripslashes est la fonction inverse de addslashes .

```
Exemple avec stripslashes
 <?php
$str = "Avez vous l\\'oreille dure?";

// Affiche : Avez vous l'oreille dure?
echo stripslashes($str);
?>
```

Voir aussi addslashes .

## 6.36.68 stristr() : Trouve la première occurence dans une chaîne (insensible à la casse)

string **stristr** ( string haystack , string needle )

stristr retourne une sous−chaîne de haystack , allant de la première occurrence de needle jusqu'à la fin de la chaîne. needle et haystack sont traitées sans tenir compte de la casse.

Si needle est introuvable, stristr retourne FALSE .

Si needle n'est pas une chaîne, elle sera convertie en entier, et utilisée comme code ascii du caractère correspondant.

```
Exemple avec stristr
 <?php
  $email = 'USER@EXEMPLE.com';
  $domaine = stristr($email, 'e');
  print $domaine;
 // Affiche ER@EXEMPLE.com
?>
```

Voir aussi strchr , strrchr , substr et ereg .

## 6.36.69 strlen() : Calcule la taille d'une chaîne

int **strlen** ( string str )

strlen retourne la taille de la chaîne string .

```
Exemple avec strlen
 <?php
$str = 'abcdef';
echo strlen($str); // 6

$str = ' ab cd ';
echo strlen($str); // 7
?>
```

Voir aussi count .

## 6.36.70 strnatcasecmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel" (insensible à la casse)

int **strnatcasecmp** ( string str1 , string str2 )

strnatcasecmp implémente l'algorithme de comparaison qui ordonne les chaînes tel qu'un homme le ferai. Cette fonction est similaire à la fonction strnatcmp , mais la comparaison n'est pas sensible à la casse. Pour plus de détails, reportez vous à Natural Order String Comparison de Martin Pool

(en anglais).

strnatcasecmp retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales.

Voir aussi ereg , strcasecmp , substr , stristr , strcmp , strncmp , strncasecmp , strnatcmp et strstr .

## 6.36.71 strnatcmp() : Comparaison de chaînes avec l'algorithme d'"ordre naturel"

int **strnatcmp** ( string str1 , string str2 )

strnatcmp implémente l'algorithme de comparaison qui ordonne les chaînes tel qu'un homme le ferai. Cette fonction est similaire à la fonction strnatcmp , mais la comparaison n'est pas sensible à la casse. Un exemple de la différence de traitement avec l'algorithme standard est présenté ci−dessous :

Exemple avec strnatcmp

```
<?php
 $arr1 = $arr2 = array("img12.png","img10.png","img2.png","img1.png");
 echo "Tri de chaînes standard\n";
 usort($arr1,"strcmp");
 print_r($arr1);
 echo "\nTri de chaînes \"ordre naturel\"\n";
 usort($arr2,"strnatcmp");
 print_r($arr2);
?>
```

Le code ci−dessus générera ce qui suit :

```
Tri de chaînes standard
Array
(
[0] => img1.png
[1] => img10.png
[2] => img12.png
[3] => img2.png
)

Tri de chaînes "ordre naturel"
Array
(
[0] => img1.png
[1] => img2.png
[2] => img10.png
[3] => img12.png
)
```

Pour plus de détails, reportez vous à Natural Order String Comparison de Martin Pool (en anglais).

strnatcmp retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est sensible à la casse.

Voir aussi ereg , strcasecmp , substr , stristr , strcmp , strncmp , strncasecmp , strnatcasecmp , strstr , natsort et natcasesort .

## 6.36.72 strncasecmp() : Compare en binaire des chaînes de caractères

int **strncasecmp** ( string str1 , string str2 , int len )

strncasecmp est similaire à strcasecmp , à la différence près qu'elle permet de limiter le nombre de caractères utilisés pour comparer str1 et str2 , avec le paramètre len . Si une des chaînes est plus courte que len , alors la longueur de cette chaîne sera utilisée pour effectuer la comparaison.

strncasecmp retourne < 0 si str1 est plus petite que str2 ; > 0 si str1 est plus grande que str2 , et 0 si elles sont égales.

Voir aussi ereg , strcasecmp , strcmp , substr , stristr et strstr .

## 6.36.73 strncmp() : Comparaison binaire des n premiers caractères

int **strncmp** ( string str1 , string str2 , int len )

**strncmp** est identique à la fonction strcmp , avec la différence que vous pouvez spécifier le nombre maximum de caractères à utiliser pour la comparaison de str1 avec str2 grâce au paramètre len . Si l'une des deux chaînes est plus petite que len , alors la longueur de cette chaîne sera utilisée.

strncmp retourne < 0 si str1 est inférieure à str2 ; > 0 si str1 est supérieure à str2 , et 0 si les deux chaînes sont égales.

Notez que cette comparaison est sensible à la casse.

Voir aussi ereg , strncasecmp , strcasecmp , substr , stristr , strcmp et strstr .

## 6.36.74 strpos() : Trouve la position d'un caractère dans une chaîne

int **strpos** ( string haystack , string needle , *int offset* )

strpos retourne la position numérique de la première occurence de needle dans la chaîne de caractères haystack . Contrairement à la fonction strrpos , celle−ci peut prendre une chaîne de caractères complète comme paramètre needle et cette chaîne sera utilisée en totalité.

Si needle n'est pas trouvée, la fonction retourne FALSE .

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur exacte de cette fonction.

Exemple avec strpos

```
<?php
 $mystring = 'abc';
 $findme   = 'a';
 $pos = strpos($mystring, $findme);

// Notez l'utilisation de ===.  Un simple == ne donnerait pas le résultat escompté
// car la lettre 'a' est à la position 0 (la première).
 if ($pos === false) {
   echo "La chaîne '$findme' n'a pas été trouvée dans la chaîne '$mystring'";
 } else {
   echo "La chaîne '$findme' a été trouvée dans la chaîne '$mystring'";
   echo " et à la position $pos";
 }
?>
```

Si needle n'est pas une chaîne, elle est convertie en entier, et utilisé comme caractère de code ASCII correspondant.

Le paramètre optionnel offset vous permet de spécifier à partir de quel caractère dans haystack vous souhaitez commencer la recherche. La position retournée sera toujours relative au début de la chaîne haystack .

Voir aussi strrpos , stripos , strripos , strrchr , substr , stristr et strstr .

## 6.36.75 strrchr() : Retourne la fin de la chaîne

string **strrchr** ( string haystack , string needle )

strrchr retourne le segment de la chaîne haystack qui commence avec la dernière occurrence de needle , jusqu'à la fin de la chaîne haystack .

strrchr retourne FALSE si needle n'est pas trouvé.

Si needle contient plus d'un caractère, seul le premier sera utilisé.

Si needle n'est pas une chaîne, elle est convertie en entier, et utilisée comme caractère de code ASCII correspondant.

Exemple avec strrchr

```
<?php
// Lit le dernier dossier dans le chemin $PATH
  $dir = substr(strrchr($PATH, ":"), 1);

 // Lit tout, après la dernière nouvelle ligne
  $text = "Ligne 1\nLigne 2\nLigne 3";
  $last = substr(strrchr($text, 10), 1 );
?>
```

Voir aussi strchr , substr , stristr et strstr .

## 6.36.76 strrev() : Inverse une chaîne

string **strrev** ( string string )

**strrev** retourne la chaîne string , après avoir changé l'ordre des caractères.

Inverser une chaîne avec strrev

```
<?php
echo strrev("Bonjour le monde!");
 // affiche "ednom el ruojnoB"
?>
```

## 6.36.77 strripos() : Trouve la position de la dernière occurence d'une chaîne dans une autre de façon insensible à la casse.

int **strripos** ( string haystack , string needle )

Retourne la position numérique de la dernière occurence de needle dans la chaîne de caractères haystack . Contrairement à strrpos , strripos est insensible à la casse. Notez aussi que les positions dans les chaînes commencent à 0, et non à 1.

Si needle n'est pas trouvé, la fonction retourne FALSE .

Attention

Cette fonction peut retourner FALSE , mais elle peut aussi retourner une valeur équivalent à FALSE , utilisable dans une condition if simple. Utilisez l' opérateur === pour tester la valeur exacte de cette fonction.

Un exemple simple avec strripos

```
 <?php
$haystack = 'ababcd';
$needle   = 'aB';

$pos      = strripos($haystack, $needle);

if ($pos === false) {
    echo "Désolé, impossible de trouver ($needle) dans ($haystack)";
} else {
    echo "Félicitations !\n";
    echo "Nous avons trouvé le dernier ($needle) dans ($haystack) à la position ($pos)";
}

/* Outputs:

   Félicitations !
   Nous avons trouvé le dernier (aB) dans (ababcd) à la position (2)

*/
?>
```

Voir aussi strrpos , strrchr , substr et stristr .

## 6.36.78 strrpos() : Trouve la position de la dernière occurence d'un caractère dans une chaîne

int **strrpos** ( string haystack , char needle )

strrpos retourne la position numérique de la dernière occurence du caractère needle dans la chaîne haystack . Notez que needle doit être un caractère unique. Si une chaîne est passée dans ce paramètre, seul le premier caractère sera utilisé.

Si needle n'est pas trouvé, strrpos retournera FALSE .

Note

Il est facile de confondre le résultat "caractère trouvé à la position 0" et "caractère non trouvé". Voici comment faire la différence :

Exemple avec strrpos

```
 <?php
// A partir de PHP 4.0b3 :
   $pos = strrpos($machaine, "b");
  if ($pos === false) { // note : trois signes égal
    // pas trouvé
  }

// en versions plus anciennes que 4.0b3 :
  $pos = strrpos($machaine, "b");
  if (!is_integer($pos)) {
    // pas trouvé
  }
?>
```

Si needle n'est pas une chaîne, elle est convertie en entier, et utilisée comme caractère de code ASCII correspondant.

Voir aussi strpos , strripos , strrchr , substr , stristr et strstr .

## 6.36.79 strspn() : Trouve le premier segment de chaîne

int **strspn** ( string str1 , string str2 )

strspn retourne la taille du segment initial de la chaîne str1 qui est entièrement constitué de caractères contenus dans str2 .

Exemple avec strspn

```
<?php
 echo strspn("421 est la réponse, mais quelle est la question ...", "1234567890");
?>
```

va afficher 3, car la chaîne "421" est le plus long segment initial de "421 est la réponse, mais quelle est la question ...", dont tous les caractères sont contenus dans "1234567890".

Voir aussi strcspn .

## 6.36.80 strstr() : Trouve la première occurence dans une chaîne

string **strstr** ( string haystack , string needle )

strstr retourne une sous−chaîne de haystack , allant de la première occurrence de needle jusqu'à la fin de la chaîne.

Si needle est introuvable, strstr retourne FALSE .

Si needle n'est pas une chaîne, elle sera convertie en entier, et utilisée comme code ascii du caractère correspondant.

Note

strstr est sensible à la casse. Pour une fonctionnalité identique, mais insensible à la casse, reportez−vous à stristr .

Exemple avec strstr

```
<?php
 $email = 'user@example.com';
 $domaine = strstr($email, '@');
 print $domaine;
// Affiche @example.com
?>
```

Voir aussi ereg , preg_match , strchr , stristr , strpos , strrchr et substr .

## 6.36.81 strtok() : Coupe une chaîne en segments

string **strtok** ( string arg1 , string arg2 )

strtok coupe la chaîne arg1 en segments, chaque segment étant délimité par arg2 . Par exemple, si vous avez une chaîne telle que "Voici un exemple", vous pouvez en extraire les différents mots en utilisant cette fonction :

Exemple avec strtok

```
<?php
 $string = "Ceci est    un exemple\ninteressant";
/* Utilisez aussi les nouvelles lignes et les tabulations comme séparateur de mots */
 $tok = strtok($string," \n    ");
 while ($tok) {
   echo "Mot = $tok<br>";
```

```
    $tok = strtok(" \n    ");
 }
?>
```

Notez que seul le premier appel à strtok nécessite les deux arguments. Tous les appels ultérieurs à strtok ne nécessitent que la chaîne à découper. Pour initialiser à nouveau strtok , ou pour recommencer, fournissez à nouveau le paramètre arg2 . La chaîne arg1 sera découpé dès que l'un des caractères de arg2 est trouvé.

Le comportement de cette fonction avec la chaîne vide a changé depuis PHP 4.1.0. L'ancien comportement était de retourner une chaîne vide, tandis que le nouveau comportement, plus correct, retourne FALSE

```
Ancien comportement de strtok
 $first_token  = strtok('/chose', '/');
    $second_token = strtok('/');
    var_dump ($first_token, $second_token);

/* Affiche :
    string(0) ""
    string(9) "chose"
*/
```

```
Nouveau comportement de strtok
 $first_token  = strtok('/chose', '/');
    $second_token = strtok('/');
    var_dump ($first_token, $second_token);

/* Affiche :
    string(9) "chose"
    bool(false)
*/
```

De plus, soyez prudents avec les séparateurs qui sont égaux à "0". Ces valeurs s'évaluent comme FALSE dans les expression conditionnelles.

Voir aussi split et explode .

## 6.36.82 strtolower() : Renvoie une chaîne en minuscules

string **strtolower** ( string str )

strtolower retourne string , après avoir converti tous les caractères alphabétique en minuscules.

Notez que la notion d'"alphabétique" est déterminée par la configuration de localisation. Cela signifie que pour la configuration par défaut "C", les caractères tels que les voyelles accentuées (comme é, è ou à) ne seront pas convertis.

```
Exemple avec strtolower
 <?php
  $str = "Marie A un Petit Agneau, et l'aime TRès fORt.";
  $str = strtolower($str);
  print $str;
 // Affiche : marie a un petit agneau, et l'aime très fort.
?>
```

Voir aussi strtoupper , ucfirst et ucwords .

## 6.36.83 strtoupper() : Renvoie une chaîne en majuscules

string **strtoupper** ( string string )

strtoupper retourne string , après avoir converti tous les caractères alphabétique en majuscules.

Notez que la notion d'"alphabétique" est déterminée par la configuration de localisation. Cela signifie que pour la configuration par défaut "C", les caractères tels que les voyelles accentuées (comme é, è ou à) ne seront pas converties.

Exemple avec strtoupper

```
<?php
 $str = "Marie A un Petit Agneau, et l'aime TRès fORt.";
 $str = strtoupper($str);
 print $str;
// Affiche : MARIE A UN PETIT AGNEAU, ET L'AIME TRES FORT.
?>
```

Voir aussi strtolower , ucfirst et ucwords .

## 6.36.84 strtr() : Remplace des chaînes dans une chaîne

string **strtr** ( string str , string from , string to ) string **strtr** ( string str , array replace_pairs )

strtr retourne la chaîne str , après avoir remplacé chaque caractère du paramètre from par son équivalent dans le paramètre to .

Si from et to sont de tailles différentes, les caractères en trop dans l'un ou l'autre seront ignorés.

Exemple avec strtr

```
<?php
 $addr = strtr($addr, "äâö", "aao");
?>
```

strtr peut être appelé avec uniquement deux arguments. Dans ce cas, elle se comporte d'une autre façon : from doit alors être un tableau associatif de paires, qui seront remplacées dans la chaîne str . strtr commence toujours par rechercher la chaîne la plus longue, et ne travaille pas sur des segments qu'elle a déjà modifié.

Exemple avec strtr avec deux arguments

```
<?php
 $trans = array("Bonjour" => "salut", "salut" => "bonjour");
 echo strtr("Bonjour à tous, j'ai dit bonjour", $trans);
?>
```

Cet exemple va afficher :

```
Salut à tous, j'ai dit bonjour
```

Note

Les paramètres optionnels to et from ont été ajoutés en PHP 4.0.0.

Voir aussi ereg_replace .

## 6.36.85 substr_count() : Compte de le nombre d'occurrences de segments dans une chaîne

int **substr_count** ( string haystack , string needle )

substr_count retourne le nombre d'occurences de needle dans la chaîne haystack .

```
Exemple avec substr_count
 <?php
  print substr_count("Ceci est un test", "ci");
   // affiche 1
?>
```

## 6.36.86 substr_replace() : Remplace un segment dans une chaîne

string **substr_replace** ( string string , string replacement , int start , *int length* )

substr_replace remplace un segment de la chaîne string par la chaîne replacement . Le segment est délimité par start et éventuellement par length . Le résultat est retourné.

Si start est positif, le remplacement se fera à partir du start −ième caractère de string .

Si start est négatif, le remplacement se fera à partir du start −ième caractère en partant de la fin de la chaîne string .

Si length est fourni et positif, il représentera la longueur du segment de code remplacé dans la chaîne string . S'il est négatif, il représentera la longueur du segment remplacé, mais compté dans l'ordre inverse de la chaîne string . S'il est omis, il prendra la valeur par défaut de la taille de la chaîne, et remplacera tout jusqu'à la fin de la chaîne string .

```
Exemple avec substr_replace
 <?php
$var = 'ABCDEFGH:/MNRPQR/';
echo "Original : $var<hr>\n";

// Remplace toute la chaîne $var par  'bob'.
 echo substr_replace($var, 'bob', 0) . "<br>\n";
echo substr_replace($var, 'bob', 0, strlen($var)) . "<br>\n";

// Insert 'bob' au début de la chaîne
echo substr_replace($var, 'bob', 0, 0) . "<br>\n";

// Remplace la séquence 'MNRPQR' par 'bob'.
 echo substr_replace($var, 'bob', 10, -1) . "<br>\n";
echo substr_replace($var, 'bob', -7, -1) . "<br>\n";

// Efface la séquence 'MNRPQR' de $var.
 echo substr_replace($var, '', 10, -1) . "<br>\n";
?>
```

Voir aussi str_replace et substr .

## 6.36.87 substr() : Retourne un segment de chaîne

string **substr** ( string string , int start , *int length* )

substr retourne le segment de string défini par start et length .

Si start est positif, la chaîne retournée commencera au caractère numéro start , dans la chaîne string . Le premier caractère est numéroté zéro.

```
Exemple avec substr
 <?php
$rest = substr("abcdef", 1);    // retourne "bcdef"
$rest = substr("abcdef", 1, 3); // retourne "bcd"
$rest = substr("abcdef", 0, 4); // retourne "abcd"
$rest = substr("abcdef", 0, 8); // retourne "abcdef"

// Accéder aux caractères via les accolades
$string = 'abcdef';
echo $string{0};                // retourne a
echo $string{3};                // retourne d
?>
```

Si start est négatif, la chaîne retournée commencera au caractère numéro start dans la chaîne string .

Exemple de start négatif

```
<?php
$rest = substr("abcdef", -1);    // retourne "f"
$rest = substr("abcdef", -2);    // retourne "ef"
$rest = substr("abcdef", -3, 1); // retourne "d"
?>
```

Si length est fourni et positif, la chaîne retournée contiendra au plus length caractères, en commençant à partir du caractère start (en fonction de la taille de la chaîne string ). Si string est plus petite que start , substr retournera FALSE .

Si length est fourni et négatif, alors le même nombre de caractères sera omis, en partant de la fin de la chaîne string . Si start représente une position hors de la chaîne, une chaîne vide sera retournée.

Utiliser une valeur négative de length

```
<?php
$rest = substr("abcdef", 0, -1);  // retourne "abcde"
$rest = substr("abcdef", 2, -1);  // retourne "cde"
$rest = substr("abcdef", 4, -4);  // retourne ""
$rest = substr("abcdef", -3, -1); // retourne "de"
?>
```

Voir aussi strrchr , substr_replace , ereg et trim .

## 6.36.88 trim() : Supprime les espaces en début et fin de chaîne

string **trim** ( string str , *string charlist* )

trim retourne la chaîne str , après avoir supprimé les caractères invisibles en début et fin de chaîne. Si le second paramètre charlist a été omis, trim supprimera les caractères suivants :

- " " ( ASCII 32 ( 0x20 )), un espace ordinaire.
- "\t" ( ASCII 9 ( 0x09 )), une tabulation.
- "\n" ( ASCII 10 ( 0x0A )), une nouvelle ligne (line feed).
- "\r" ( ASCII 13 ( 0x0D )), un retour chariot (carriage return).
- "\0" ( ASCII 0 ( 0x00 )), le caractère NUL .
- "\x0B" ( ASCII 11 ( 0x0B )), une tabulation verticale.

Il est aussi possible de spécifier les caractères à supprimer en utilisant le paramètre charlist . Listez simplement les caractères que vous voulez supprimer dans ce paramètre. Avec .. , vous pourrez spécifier des intervalles de caractères.

Exemple avec trim

```
<?php

$text = "        Voici quelques mots :) ...  ";
$trimmed = trim($text);
// $trimmed = "Voici quelques mots :) ..."
$trimmed = trim($text,"     .");
// $trimmed = "Voici quelques mots :)"
$clean = trim($binary,"\0x00..\0x1F");
// supprime tous les caractères de contrôle ASCII au début de la chaîne de caractères.
// (de 0 à 31 inclus)
?>
```

Note

Le second paramètre a été ajouté en PHP 4.1.0.

Voir aussi ltrim et rtrim .

## 6.36.89 ucfirst() : Met le premier caractère en majuscule

string **ucfirst** ( string str )

**ucfirst** retourne la chaîne str après avoir remplacé le premier caractère par sa majuscule, si le premier caractère est alphabétique.

Notez que la notion d'"alphabétique" est déterminée par la configuration de localisation. Cela signifie que pour la configuration par défaut "C", les caractères tels que les voyelles accentuées (comme é, è ou à) ne seront pas convertis.

Exemple avec ucfirst

```
 $foo = 'bonjour tout le monde!';
$foo = ucfirst($foo);             // Bonjour tout le monde!

$bar = 'BONJOUR TOUT LE MONDE!';
$bar = ucfirst($bar);             // BONJOUR TOUT LE MONDE!
$bar = ucfirst(strtolower($bar)); // Bonjour tout le monde!
```

Voir aussi strtolower , strtoupper et ucwords .

## 6.36.90 ucwords() : Met en majuscule la première lettre de tous les mots

string **ucwords** ( string str )

**ucwords** retourne la chaîne str après avoir mis en majuscule la première lettre de tous les mots, si ce caractère est alphabétique.

Exemple avec ucwords

```
 <?php
$foo = 'bonjour tout le monde!';
$foo = ucwords($foo);             // Bonjour Tout Le Monde!

 $bar = 'BONJOUR TOUT LE MONDE!';
$bar = ucwords($bar);             // BONJOUR TOUT LE MONDE!
$bar = ucwords(strtolower($bar)); // Bonjour Tout Le Monde!
 ?>
```

Note

La définition d'un mot est : toute séquence de caractères qui suit immédiatement un caractère blanc (espace, tabulation, nouvelle ligne, retour chariot, tabulation verticale).

Voir aussi strtoupper , strtolower et ucfirst .

## 6.36.91 vprintf() : Affiche une chaîne formatée

void **vprintf** ( string format , array args )

vprintf affiche le tableau args , sous forme de chaîne formatée grâce à format Le format est le même que celui utilisé par sprintf .

vprintf fonctionne comme printf mais accepte un tableau comme argument, au lieu d'une liste d'arguments.

Voir aussi printf , sprintf et vsprintf

## 6.36.92 vsprintf() : Retourne une chaîne formatée

string **vsprintf** ( string format , array args )

vsprintf retourne une chaîne formatée à partir du tableau de valeurs args , et en utilisant le format format . Le format est le même que celui utilisé par sprintf .

vsprintf fonctionne comme sprintf mais accepte un tableau comme argument, au lieu d'une liste d'arguments.

Voir aussi sprintf , vsprintf et vprintf

## 6.36.93 wordwrap() : Effectue la césure d'une chaîne

string **wordwrap** ( string str , *int width* , *string break* , *boolean cut* )

wordwrap retourne la chaîne str , après avoir inséré break tous les width caractères.

Par défaut, wordwrap va automatiquement insérer une nouvelle ligne tous les 75 caractères, si width et break ne sont pas fournis.

Si le paramètre cut est mis à 1, la chaîne sera toujours césurée à la taille width . Si vous avez un mot qui est plus long que la taille de césure, il sera coupé en morceaux.

Note

Le paramètre optionnel cut a été ajouté en PHP 4.0.3.

Exemple avec wordwrap

```
<?php
 $text = "Portez ce vieux whisky au juge blond qui fume.";
 $newtext = wordwrap( $text, 20 );

 echo "$newtext\n";
?>
```

Cet exemple va afficher :

```
Portez ce vieux
whisky au juge
blond qui fume.
```

Exemple avec wordwrap avec césure

```
<?php
 $text = "Un mot tres tres looooooooooooooooong.";
 $newtext = wordwrap( $text, 8, "\n", 1);

 echo "$newtext\n";
?>
```

Cet exemple va afficher :

```
Un mot
tres
tres
looooooo
oooooooo
ooong.
```

Voir aussi nl2br .

# 6.37 URL

## 6.37.1 Introduction

Fonctions permettant de traiter les URL : encoder, décoder, et analyser.

## 6.37.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.37.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.37.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.37.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.37.6 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- base64_decode
- base64_encode
- get_meta_tags
- http_build_query
- parse_url
- rawurldecode
- rawurlencode
- urldecode
- urlencode

## 6.37.8 base64_encode() : Encode une chaîne en MIME base64

string **base64_encode** ( string data )

base64_encode retourne data encodé en base64. Cet encodage est fait pour permettre aux informations binaires d'être manipulées par les systèmes qui ne gèrent pas correctement les 8 bits, comme par exemple, les corps de mail.

Une chaîne encodée Base64 prend, grosso modo, 33% de plus que les données initiales.

Exemple avec base64_encode

```
<?php
 $str = 'Ceci est une chaine encodee';
 echo base64_encode($str);
?>
```

Cet exemple va afficher :

```
Q2VjaSBlc3QgdW5lIGNoYWluZSBlbmNvZGVl
```

Voir aussi base64_decode , chunk_split et la RFC 2045 section 6.8.

## 6.37.9 get_meta_tags() : Extrait tous les méta tags d'un fichier

array **get_meta_tags** ( string filename , *int use_include_path* )

get_meta_tags ouvre le fichier filename et l'analyse ligne par ligne à la recherche de balises <meta>, de la forme

Exemple de Meta Tags

```
 <meta name="author" content="name">
<meta name="keywords" content="php documentation">
<meta name="DESCRIPTION" content="a php manual">
<meta name="geo.position" content="49.33;-86.59">
</head>
```

Attention au format des nouvelles lignes : PHP utilise une fonction native pour analyser le fichier, et les fichiers Mac ne seront pas reconnus.

La valeur de la propriété sera utilisé comme clé du tableau, et sa valeur comme valeur correspondante de la clé. Vous pourrez ainsi passer en revue facilement ce tableau avec les fonctions de tableaux standard. Les caractères spéciaux présents dans la valeur seront replacés par un souligné ( "_" ), et le reste est converti en minuscule.

En donnant la valeur de 1 au paramètre optionnel use_include_path fera que get_meta_tags recherchera aussi le fichier dans l'include path.

Résultat de get_meta_tags

```
 <?php
// Supposons que les balises ci-dessus sont disponibles sur exemple.com
  $tags = get_meta_tags('http://www.exemple.com/');

// Notez que les clés sont en minuscule, et
 // le . a été remplacé par _ dans la clé
  print $tags['author'];       // nom
  print $tags['keywords'];     // documentation php
   print $tags['description'];  // un manuel PHP
  print $tags['geo_position']; // 49.33;-86.59
?>
```

Note

Depuis PHP 4.0.5, get_meta_tags supporte les attributs HTML sans guillemets.

Voir aussi htmlentities et urlencode .

## 6.37.10 http_build_query() : Génère une chaîne de requête en encodage URL

string **http_build_query** ( array formdata , *string numeric_prefix* )

http_build_query génère une chaîne en encodage URL, construite à partir du tableau indexé ou associatif formdata . formdata peut être un tableau ou un objet contenant une structure mono−dimensionnelle simple, ou un tableau de tableaux (qui peuvent aussi contenir des tableaux). Si des indices numériques sont utilisés dans le tableau de base et que numeric_prefix est fourni, il sera utilisé pour préfixer les noms des index pour les éléments du tableau de base seulement. Cela permet de générer des noms de variables valides, si les données sont ensuites décodées par PHP ou une application CGI.

Utilisation simple de http_build_query

```
<?php
$data = array('foo'=>'bar',
              'baz'=>'boom',
              'cow'=>'milk',
              'php'=>'hypertext processor');

echo http_build_query($data); // foo=bar&baz=boom&cow=milk&php=hypertext+processor

?>
```

http_build_query avec tableau indexé.

```
<?php
$data = array('foo', 'bar', 'baz', 'boom', 'cow' => 'milk', 'php' =>'hypertext processor');

echo http_build_query($data);
/* affiche :
      0=foo&1=bar&2=baz&3=boom&cow=milk&php=hypertext+processor
 */

 echo http_build_query($data, 'myvar_');
/* affiche :
      myvar_0=foo&myvar_1=bar&myvar_2=baz&myvar_3=boom&cow=milk&php=hypertext+processor
 */
?>
```

http_build_query avec tableau complexe

```
<?php
$data = array('user'=>array('name'=>'Bob Smith',
                            'age'=>47,
                            'sex'=>'M',
                            'dob'=>'5/12/1956'),
              'pastimes'=>array('golf', 'opera', 'poker', 'rap'),
              'children'=>array('bobby'=>array('age'=>12,
                                               'sex'=>'M'),
                                'sally'=>array('age'=>8,
                                               'sex'=>'F')),
              'CEO');

echo http_build_query($data, 'flags_');
?>
```

cet exemple va afficher : (sur plusieurs lignes pour lisibilité)

```
user[name]=Bob+Smithr[sex]=M%1F1956/> pastimes[0]=golfar
children[bobby][age]=12sex]=Mage]=8/> children[sally][sex]=F/>
```

Note

Seuls les éléments indéxés numériquement ("CEO") dans le tableau de base sont préfixés. Les autres indices numériques à d'autres niveaux n'ont pas besoins d'être préfixés pour avoir des noms valides.

Utilisation de http_build_query avec un objet

```
 <?php
class myClass {
  var $foo;
  var $baz;

  function myClass() {
    $this->foo = 'bar';
    $this->baz = 'boom';
  }
}

$data = new myClass();

echo http_build_query($data); // foo=bar&baz=boom

?>
```

Voir aussi parse_str , parse_url , urlencode et array_walk .

## 6.37.11 parse_url() : Analyse une URL et retourne ses composants

array **parse_url** ( string url )

parse_url retourne un tableau associatif contenant les composants de l'URL. Si l'un des composants n'est pas présent dans l'URL, l'entrée correspondante dans le tableau ne sera pas créée. Les composants sone :

- scheme − e.g. http
- host
- port
- user
- pass
- path
- query − après le point d'interrogation ?
- fragment − après le signe dièse #

Cette function **n'est pas** destinée à valider des URL, mais à les scinder en parties. Les URL partielles sont aussi acceptés, parse_url essaie alors de l'analyser correctement.

Note

parse_url ne fonctionne que avec les URL absolues.

Exemple avec parse_url

```
$ php -r 'print_r( parse_url("http://username:password@hostname/path?arg=value#anchor"));'
Array
(
[scheme] => http
[host] => hostname
[user] => username
[pass] => password
[path] => /path
[query] => arg=value
[fragment] => anchor
)

$ php -r 'print_r( parse_url("http://invalid_host..name/"));'
Array
(
[scheme] => http
[host] => invalid_host..name
[path] => /
)
```

Voir aussi pathinfo , parse_str , dirname et basename .

## 6.37.12 rawurldecode() : Décode une chaîne URL

string **rawurldecode** ( string str )

rawurldecode retourne la chaîne str dont les séquences de caractères %xy , avec xy deux valeurs hexadécimales, auront été remplacées par le caractère ASCII correspondant.

base64_decode example

```
<?php
$str = 'foo%20bar%40baz';
echo rawurldecode($str);
?>
```

This example will produce :

```
foo bar@baz
```

Voir aussi rawurlencode , urldecode et urlencode .

## 6.37.13 rawurlencode() : Encode une chaîne en URL, selon la RFC1738

string **rawurlencode** ( string str )

rawurlencode retourne une chaîne dont tous les caractères non−alpha−numériques (hormis −_. ) auront été remplacés par des séquences %xy ( % ), avec xy deux valeurs hexadécimales. Ce codage est conforme à la RFC1738 qui évite que les caractères spéciaux soient interprétés comme des délimiteurs, et pour protéger les URL lors du transfert (contrairement à certains systèmes email). Par exemple, si vous voulez mettre un mot de passe dans une URL de ftp :

Exemple avec rawurlencode

```
echo '<a href="ftp://user:', rawurlencode('foo @+%/'),
    '@ftp.my.com/x.txt">';
```

Ou, si vous transmettez un chemin dans une URL

Exemple avec rawurlencode

```
echo '<a href="http://x.com/department_list_script/',
   rawurlencode('ventes et marketing/Bruxelles'), '">';
```

Voir aussi rawurldecode , urldecode et urlencode , et la RFC 2045 section 6.8 .

## 6.37.14 urldecode() : Décode une chaîne encodée URL

string **urldecode** ( string str )

urldecode décode toutes les séquences % ## et les remplace par leur valeur. La chaîne ainsi décodée est retournée.

Exemple avec urldecode

```
<?php
 $a = split('&', $QUERY_STRING);
 $i = 0;
 while ($i < count ($a)) {
   $b = split ('=', $a[$i]);
   echo 'La valeur du paramètre ', htmlspecialchars(urldecode($b[0])),
        ' est ', htmlspecialchars(urldecode($b[1])), "<BR>";
   $i++;
 }
?
```

Voir aussi urlencode , rawurlencode et rawurldecode .

## 6.37.15 urlencode() : Encode une chaîne en URL

string **urlencode** ( string str )

urlencode retourne une chaîne dont les caractères non alpha−numériques (hormis −_. ) sont remplacés par des séquences commençant par un caractère pourcentage ( % ), suivi de deux chiffres hexadécimaux. Les espaces sont remplacés par des signes plus ( + ). Ce codage est celui qui est utilisé pour poster des informations dans les formulaires HTML. Le type MIME est application/x−www−form−urlencoded . Ce codage est différent de celui spécifié dans la RFC1738 (voir rawurlencode ) : pour des raisons historiques, les espaces sont remplacés par des signes plus ( + ). urlencode est pratique pour transmettre des informations via une URL. C'est aussi un moyen de passer des informations d'une page à l'autre.

Exemple avec urlencode

```
<?php
 echo '<A HREF="moncgi?foo=', urlencode ($userinput), '">';
?>
```

Note: Faites bien attention aux variables qui ressemblent à des entités HTML, comme par exemple &amp, &copy et &pound, qui sont analysées par le client web et remplacée par leur valeur. C'est un vrai problème qui a été montré par le W3C depuis longtemps. La référence est ici : http://www.w3.org/TR/html4/appendix/notes.html#h−B.2.2 . PHP supporte le remplacement de séparateur d'arguments par un point−virgule, comme recommandé par le W3C, grâce à la directive arg_separator .ini. Malheureusement, la plupart des clients web n'envoient pas leurs données de formulaire avec des points−virgules. Une solution plus portable est d'utiliser & à la place de & comme séparateur. Vous n'avez alors pas à changer la directive arg_separator. Laissez−la à &, mais encodez vos URL avec htmlentities .

Exemple avec urlencode et htmlentities

```
<?php
 echo '<A HREF="moncgi?foo=', htmlentities (urlencode ($userinput) ), '">';
?>
```

Voir aussi urldecode , htmlentities , rawurldecode et rawurlencode .

# 6.38 Variables

## 6.38.1 Introduction

Pour plus de détails sur le comportement des variables, reportez vous à la section Variables du chapitre Référence du langage .

## 6.38.2 Pré−requis

Ces fonctions sont disponibles dans le module PHP standard, qui est toujours accessible.

## 6.38.3 Installation

Il n'y pas d'installation nécessaire pour utiliser ces fonctions, elles font parties du coeur de PHP.

## 6.38.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

| Nom | Par défaut | Modifiable |
|---|---|---|
| unserialize_callback_func | "" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*unserialize_callback_func string*

La fonction de callback d'unserialize sera appelée (avec le nom de la classe non définie en paramètre), si le délinéarisateur ne trouve pas la classe spécifiée, ou si la fonction n'implémente pas la classe manquante. Ne configurez cette option que si vous souhaitez vraiment implémenter une telle fonction.

Voir aussi unserialize .

## 6.38.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.38.6 Constantes prédefinies

Cette extension ne définit aucune constante.


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources

- Constantes prédefinies
- doubleval
- empty
- floatval
- get_defined_vars
- get_resource_type
- gettype
- import_request_variables
- intval
- is_array
- is_bool
- is_callable
- is_double
- is_float
- is_int
- is_integer
- is_long
- is_null
- is_numeric
- is_object
- is_real
- is_resource
- is_scalar
- is_string
- isset
- print_r
- serialize
- settype
- strval
- unserialize
- unset
- var_dump
- var_export

## 6.38.8 empty() : Détermine si une variable contient une valeur non nulle

bool **empty** ( mixed var )

| Note |
| --- |
| empty est une structure de langage. |

empty retourne la valeur FALSE si la variable var est affectée ou bien a une valeur différente de 0; la valeur TRUE dans les autres cas.

Exemple avec empty

```
<?php
$var = 0;
 if (empty($var)) { // retourne
<font face="courrier">TRUE</font>

   print 'soit $var vaut 0, soit il n\'est pas défini';
 }
  if (!isset($var)) { // retourne
<font face="courrier">FALSE</font>

   print '$var n'est pas définie';
 }
?>
```

Notez que cette fonction n'a pas de sens si elle est utilisée sur autre chose qu'une variable. i.e. empty (addslashes ($name)) n'a pas de sens, car cela revient à vérifier une entité qui n'est pas une variable.

| Note |
| --- |
| Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions . |

Voir aussi isset et unset .

## 6.38.9 floatval() : Convertit une chaîne en nombre à virgule flottante

float **floatval** ( mixed var )

floatval retourne la valeur de type float (nombre à virgule flottante), extraite à partir du paramètre var .

var peut être n'importe quel type scalaire. Vous ne pouvez pas utiliser floatval sur un tableau ou un objet.

Exemple avec floatval

```
<?php
 $var = '122.34343Le';
 $float_value_of_var = floatval ($var);
 print $float_value_of_var; // Affiche 122.34343
?>
```

Voir aussi intval , strval , settype et Transtypage .

## 6.38.10 get_defined_vars() : Liste toutes les variables définies

array **get_defined_vars** ( void )

get_defined_vars retourne un tableau multidimensionnel contenant la liste de toutes les variables définies, qu'elles soient des variables d'environnement, de serveurs ou définies par l'utilisateur.

Exemple avec get_defined_vars

```
<?php
 $b = array(1,1,2,3,5,8);

 $arr = get_defined_vars();

// affiche $b
 print_r($arr["b"]);

// affiche le chemin jusqu'à l'interpréteur
 // CGI PHP (si PHP est utilisé en CGI)
// i.e. /usr/local/bin/php
 echo $arr["_"];

// affiche la ligne de commande, s'il y en a une
 print_r($arr["argv"]);

// affiche toutes les variables serveurs
 print_r($arr["HTTP_SERVER_VARS"]);

// affiche toutes les clés disponibles dans les tableaux de variables
 print_r(array_keys(get_defined_vars()));
?>
```

Voir aussi get_defined_functions et get_defined_constants .

## 6.38.11 get_resource_type() : Retourne le type de ressource

string **get_resource_type** ( resource handle )

get_resource_type retourne une chaîne représentant le type de ressources de handle . Si le paramètre n'est pas une ressource valide, une erreur est générée.

Exemple avec get_resource_type

```
<?php
 $c = mysql_connect();
 echo get_resource_type($c)."\n";
// affiche : mysql link
// (lien mysql)
 $fp = fopen("foo","w");
 echo get_resource_type($fp)."\n";
// affiche : file
// (fichier)
 $doc = new_xmldoc("1.0");
 echo get_resource_type($doc->doc)."\n";
// affiche : domxml document
// (document domxml)
?>
```

## 6.38.12 gettype() : Retourne le type de la variable

string **gettype** ( mixed var )

gettype retourne le type de la variable PHP var .

Attention

N'utilisez jamais gettype pour vérifier un type, car la chaîne retournée est sujette à modification sans préavis dans une version ultérieure. De plus, c'est bien plus long comme cela, car cela implique une comparaison de chaîne.

A la place, utilisez les fonctions is_* .

Les chaînes de caractères que peut retourner la fonction sont les suivantes :

- " boolean " (depuis PHP 4)
- " integer "
- " double " (pour des raisons historiques, "double" est retournée lorsqu'une valeur de type float est fournie, au lieu de la chaîne "float")
- " string "
- " array "
- " object "
- " resource " (depuis PHP 4)
- " NULL " (depuis PHP 4)
- " user function " (PHP 3 uniquement, abandonné)
- " unknown type "

En PHP 4, il est recommandé d'utiliser function_exists et method_exists pour remplacer l'utilisation de gettype sur une fonction.

Voir aussi settype , is_array , is_bool , is_float , is_integer , is_null , is_numeric , is_object , is_resource , is_scalar et is_string .

## 6.38.13 import_request_variables() : Importe les variables de GET/POST/Cookie dans l'environnement global

bool **import_request_variables** ( string types , *string prefix* )

import_request_variables importe les variables GET/POST/Cookie dans l'environnement global. C'est pratique si vous avez désactivé register_globals , mais que vous voulez enregistrer des variables dans l'environnement global.

En utilisant le paramètre types , vous pouvez spécifier les variables que vous voulez importer. Vous pouvez utiliser 'G', 'P' et 'C' pour désigner respectivement GET, POST et Cookie. Ces caractères ne sont pas sensible la casse, et vous pouvez les combiner entre eux. POST inclus les fichiers uploadés. Notez que l'ordre des lettres est important. En utilisant par exemple "gp", les variables de POST écraseront par les variables de GET, ayant le même nom. Toute autre lettre, autre que GPC est ignorée.

Le paramètre prefix est utilisé comme un préfixe de nom de variable, qui sera ajoutée au début de tous les noms de variables importées. Si vous avez par exemple une variable GET du nom de " userid ", et utilisez le préfixe " pref_ ", la variable ainsi générée s'appellera $pref_userid .

Si vous souhaitez importer d'autres variables dans l'environnement global, comme par exemple $_SERVER , considérez l'utilisation de la fonction extract .

| Note |
| --- |
| Bien que le paramètre prefix soit optionnel, il sera généré une alerte E_NOTICE si vous ne spécifiez pas de préfixe, ou si vous utilisez une chaîne vide comme préfixe. C'est potentiellement un trou de sécurité. Les alertes de ce niveau ne sont pas affichées par défaut. |

Exemple avec import_request_variables

```
<?php
// Ce code va importer les variables GET et POST
// avec un préfixe "rvar_"
import_request_variables("gP", "rvar_");

print $rvar_foo;
?>
```

Voir aussi $_REQUEST , register_globals , variables prédéfinies et extract .

## 6.38.14 intval() : Retourne la valeur numérique entière équivalent d'une variable

int **intval** ( mixed var , *int base* )

intval retourne la valeur numérique entière (integer) de la variable var , en convertissant la valeur dans la base spécifiée (par défaut en base 10).

var peut être de type scalaire. Vous ne pouvez pas utiliser la fonction intval avec un tableau ou un objet.

Voir aussi doubleval , strval , settype et Transtypage .

## 6.38.15 is_array() : Détermine si une variable est un tableau

bool **is_array** ( mixed var )

is_array renvoie la valeur TRUE si la variable var est un array , FALSE sinon.

Voir aussi is_float , is_int , is_integer , is_string et is_object .

## 6.38.16 is_bool() : Détermine si une variable est un booléen

bool **is_bool** ( mixed var )

is_bool retourne TRUE si var est un boolean .

Exemple avec is_bool

```
<?php
$a = false;
$b = 0;

// Comme $a est un booléen, ceci est vrai
if (is_bool($a)) {
    print "Oui, c'est un booléen";
}

// Comme $b n'est pas un booléen, ceci est faux
if (is_bool($b)) {
    print "Oui, c'est un booléen";
}
?>
```

Voir aussi is_array , is_float , is_int , is_integer , is_string et is_object .

## 6.38.17 is_callable() : Détermine si l'argument peut être appelé comme fonction

bool **is_callable** ( mixed var , *bool syntax_only* , *string callable_name* )

**is_callable** vérifie qu'une variable peut être appelée comme fonction. Cette fonction peut vérifier qu'une variable contient un nom de fonction valide, ou bien qu'elle contient un tableau, avec un objet et un nom de méthode.

Le paramètre var peut être le nom d'une fonction stocké dans une variable de chaîen, ou un objet, et un nom de méthode pour cet objet, sous la forme : array( $unObjet, 'NomDeMethode' )

Si l'argument syntax_only vaut TRUE , la fonction ne va vérifier que si var peut être une fonction ou une méthode. Il va simplement rejeter les variables qui ne sont pas des chaînes, ou des tableaux qui n'ont pas la bonne structure pour être utilisé comme fonction de call back. Les tableaux valides sont supposés n'avoir que deux entrées, le premier étant un objet ou une chaîne, et de second une chaîne.

L'argument callable_name recoit le nom à utiliser ("callable name"). Dans l'exemple ci−dessous, il vaut "uneClasse:uneMethode" . Notez que, bien que uneClasse:uneMethode() puisse être appelé sous forme static, ce n'est pas le cas.

Exemple avec is_callable

```
<? php
//  Comment vérifier qu'une variable peut être appelée
 //  comme fonction?
```

```
//
//  Variable simple contenant une fonction
//

function uneFonction() {
}

$fonctionVariable = 'uneFonction';

var_dump( is_callable( $fonctionVariable, false, $callable_name ));  // bool(true)

echo $callable_name, "\n";  // uneFonction

//
//  Tableu contenant une méthode
//

class uneClasse {

  function uneMethode() {
  }

}

$unObjet = new uneClasse();

$methodeVariable = array( $unObjet, 'uneMethode' );

var_dump( is_callable( $methodeVariable, true, $callable_name ));  //  bool(true)

echo $callable_name, "\n";  //  uneClasse:uneMethode

?>
```

## 6.38.18 is_double() : Alias de is_float

is_double est un alias de is_float .

## 6.38.19 is_float() : Détermine si une variable est de type float

bool **is_float** ( mixed var )

Cette fonction est un alias de la fonction is_double .

Voir aussi is_double , is_bool , is_real , is_int , is_integer , is_string , is_object , is_array , et is_long .

## 6.38.20 is_int() : Détermine si une variable est de type integer

bool **is_int** ( mixed var )

is_int est un alias de la fonction is_long .

Voir aussi is_bool , is_double , is_float , is_integer , is_string , is_real , is_object , is_array , et is_long .

## 6.38.21 is_integer() : Détermine si une variable est de type int

bool **is_integer** ( mixed var )

Cette fonction est un alias de la fonction is_long .

Voir aussi is_bool , is_double , is_float , is_int , is_string , is_real , is_object , is_array , et is_long .

## 6.38.22 is_long() : Alias de is_int

is_long est un alias de is_int .

## 6.38.23 is_null() : Indique si une variable vaut NULL

bool **is_null** ( mixed var )

is_null retourne TRUE , si var est NULL , et FALSE .

Reportez−vous au type NULL pour savoir quand une variable est considérée comme contenant la valeur NULL ou non.

Voir aussi NULL , is_bool , is_numeric , is_float , is_int , is_string , is_object , is_array , is_integer et is_real

## 6.38.24 is_numeric() : Détermine si une variable est un type numérique

bool **is_numeric** ( mixed var )

is_numeric retourne TRUE si var est un nombre, ou une chaîne numérique, ou FALSE sinon.

Voir aussi is_bool , is_float , is_int , is_string , is_object , is_array et is_integer .

## 6.38.25 is_object() : Détermine si une variable est de type object

bool **is_object** ( mixed var )

is_object renvoie TRUE si la variable var est un objet, FALSE sinon.

Voir aussi is_bool , is_int , is_integer , is_float , is_string et is_array .

## 6.38.26 is_real() : Détermine si une variable est de type real

bool **is_real** ( mixed var )

Cette fonction est un alias de la fonction is_double .

Voir aussi is_bool , is_long , is_int , is_integer , is_float , is_double , is_object , is_string , et is_array .

## 6.38.27 is_resource() : Détermine si une variable est une ressource

bool **is_resource** ( mixed var )

is_resource retourne TRUE si la variable var est une ressource PHP, sinon FALSE .

Les ressources peuvent être des pointeurs de fichiers, des identifiants de résultats SQL, qui sont allouées et libérées en interne, par PHP, et qui peuvent demander un peut de nettoyage lorsqu'elle sont devenues inutiles, mais pas encore supprimées.

## 6.38.28 is_scalar() : Indique si une variable est un scalaire

bool **is_scalar** ( mixed var )

is_scalar retourne TRUE si la variable var est scalaire, et FALSE sinon.

Les variables scalaires sont celles qui contiennent des entiers, des nombres à virgule flottante, des chaînes de caractères ou des booléens. Par exemple :

Exemple avec is_scalar

```
<?php
function show_var($var) {
    if (is_scalar($var))
        echo $var;
    else
        var_dump($var);
}
$pi = 3.1416;
$proteines = array("hemoglobine", "cytochrome c oxidase", "ferredoxine");
show_var($pi);
// affiche : 3.1416
show_var($proteines)
// affiche:
// array(3) {
//   [0]=>
//   string(10) "hemoglobine"
//   [1]=>
//   string(20) "cytochrome c oxidase"
//   [2]=>
//   string(10) "ferredoxine"
// }
?>
```

Voir aussi is_bool , is_numeric , is_float , is_int , is_real , is_string , is_object , is_array et is_integer .

## 6.38.29 is_string() : Détermine si une variable est de type chaîne

bool **is_string** ( mixed var )

is_string renvoie TRUE si la variable var est du type chaîne de caractères ( string ), FALSE sinon.

Voir aussi is_bool , is_int , is_integer , is_float , is_real , is_object et is_array .

## 6.38.30 isset() : Détermine si une variable est affectée

bool **isset** ( mixed var , *mixed var* , ... )

Note

isset est une structure du langage.

isset renvoie TRUE si la variable var est définie, FALSE sinon.

Si une variable a été détruite avec la fonction unset , la fonction isset renverra FALSE . isset reverra FALSE lors du test d'une variable de valeur NULL . Notez aussi que le caractère nul ( "\0" ) n'est pas équivalent à la constante PHP NULL .

Exemple avec isset

```
<?php

$var = '';

// Ceci est vrai, alors le texte est affiché
if (isset($var)) {
    print "Cette variable existe, donc je peux l'afficher.";
}

// Dans les exemples suivants, nous utilisons var_dump() pour afficher
 // le retour de la fonction isset().

$a = "test";
$b = "anothertest";

var_dump( isset($a) );      // TRUE
var_dump( isset ($a, $b) ); // TRUE

unset ($a);

var_dump( isset ($a) );     // FALSE
var_dump( isset ($a, $b) ); // FALSE

$foo = NULL;
var_dump( isset ($foo) );   // FALSE

?>
```

Cela fonctionne aussi avec les tableaux :

Exemple avec isset et un tableau

```
<?php

$a = array ('test' => 1, 'bonjour' => NULL);

var_dump( isset ($a['test']) );            // TRUE
var_dump( isset ($a['foo']) );             // FALSE
var_dump( isset ($a['bonjour']) );           // FALSE

// La clé 'bonjour' vaut NULL est est considérée comme non existante
// Si vous voulez vérifier l'existance de cette clée, utilisez cette fonction
var_dump( array_key_exists('bonjour', $a) ); // TRUE

?>
```

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Voir aussi empty , unset defined , array_key_exists et l'opérateur de contrôle de rapport d'erreur @ .

## 6.38.31 print_r() : Affiche des informations lisibles pour une variable

bool **print_r** ( mixed expression , *bool return* )

print_r affiche des informations à propos d'une variable, de manière à ce qu'elle soit lisible. Pour une chaîne, un entier ou un double, la valeur sera elle même sera affichée. Pour les tableaux, les valeurs seront présentées dans un format qui montre les clés et les valeurs. Une notation similaire est disponible pour les objets.

Gardez en tête que print_r place le pointeur de tableau à la fin du tableau. Utilisez reset pour le ramener au début.

Exemple avec print_r

```
<pre>
<?php
    $a = array ('a' => 'pomme',
                'b' => 'banane',
                'c' => array ('x',
                              'y',
                              'z'));
    print_r ($a);
```

```
?>
</pre>
```

va afficher
<pre>
Array
(
[a] => pomme
[b] => banane
[c] => Array
(
[0] => x
[1] => y
[2] => z
)
)
</pre>

Si vous voulez obtenir le résultat de print_r dans une chaîne, utilisez le paramètre return . Si ce paramètre vaut TRUE , print_r retournera ce qu'il aurait affiché.

Utilisation du paramètre return de print_r

```
<?php
   $b = array ('m' => 'mangue', 'foo' => 'bar', 'x' => array ('x', 'y', 'z'));
   $resultat = print_r ($b, true);
    //$resultat contient maintenant le résultat de print_r()
?>
```

Note

Si vous devez récupérer le résultat de print_r avec une version de PHP plus ancienne que 4.3.0, utilisez les fonctions de bufferisation de sortie .

Note

Le paramètre return a été ajouté en PHP 4.3.0.

Note

Avant PHP 4.0.4, print_r bouclera à l'infini si un array ou un object contient une référence sur lui même. Un exemple classique est notamment la ligne print_r($GLOBALS) car $GLOBALS est lui même une variable globale, qui donc, contient une référence sur elle même.

Voir aussi ob_start , var_dump et var_export .

## 6.38.32 serialize() : Linéarise une variable

string **serialize** ( mixed value )

serialize retourne une chaîne contenant une représentation linéaire de value , pour stockage.

C'est une technique pratique pour stocker ou passer des valeurs de PHP entre scripts, sans perdre ni leur structure, ni leur type.

Pour récupérer une variable linéarisée, et retrouver une variable, utilisez unserialize . serialize acceptent les types integer , double , string , array (multidimensionnels) et object (les propriétés des objets seront linéarisées, mais pas les méthodes).

Exemple avec serialize

```
<?php
// $session_data contient un tableau multi-dimensionnel , avec les
// informations de session de l'utilisateur courant. On utilise serialize()
// pour les stocker dans une base de données
$conn = odbc_connect ("webdb", "php", "chicken");
$stmt = odbc_prepare ($conn,
      "UPDATE sessions SET data = ? WHERE id = ?");
$sqldata = array (serialize($session_data), $PHP_AUTH_USER);
if (!odbc_execute ($stmt, &$sqldata)) {
    $stmt = odbc_prepare($conn,
     "INSERT INTO sessions (id, data) VALUES(?, ?)");
    if (!odbc_execute($stmt, &$sqldata)) {
        /* Un problème est survenu ! */
    }
}
?>
```

## 6.38.33 settype() : Affecte un type à une variable

bool **settype** ( mixed var , string type )

settype force le type de la variable var en type .

Les valeurs possibles pour le paramètre type sont :

- " boolean " (ou, depuis PHP 4.2.0, " bool ")
- " integer " (ou, depuis PHP 4.2.0, " int ")
- " float " (uniquement depuis PHP 4.2.0. Pour les anciennes versions, utilisez l'alternative " double ")
- " string "
- " array "
- " object "
- " null " (depuis PHP 4.2.0)

Cette fonction retourne TRUE en cas de succès, FALSE en cas d'échec.

```
Exemple avec settype
 $foo = "5bar"; // chaîne
$bar = true;   // booléen

settype($foo, "integer"); // $foo vaut maintenant 5   (integer)
settype($bar, "string");  // $bar vaut maintenant "1" (string)
```

Voir aussi gettype , transtypage et définition du type .

## 6.38.34 strval() : Retourne la valeur de la variable, au format chaîne

string **strval** ( mixed var )

strval retourne la valeur de la variable var , au format chaîne de caractères.

var peut être un scalaire. Vous ne pouvez pas utiliser la fonction strval avec des tableaux ou des objets.

Voir aussi doubleval , intval , settype et Transtypage .

## 6.38.35 unserialize() : Crée une variable PHP à partir d'une valeur linéarisée

mixed **unserialize** ( string str , *string callback* )

unserialize prend une variable linéarisée (voir serialize ) et la convertit en variable PHP. La valeur convertie est retournée par la fonction, et peut être de type integer , double , string , array ou object . Les objets linéarisés perdent leurs méthodes.

Note

Directive unserialize_callback_func

Il est possible de définir une fonction de callback qui sera appelée si une classe undéfinie est utilisée lors de la délinéarisation (ce qui évitera de voir l'objet recevoir le type d'objet incomplet object "__PHP_Incomplete_Class"). Utilisez dans votre fichier php.ini ou le fichier .htaccess , ou encore avec la fonction ini_set , pour définir une fonction 'unserialize_callback_func'. Chaque fois qu'une classe non définie sera instantiée, cette fonction sera appelée. Pour désactiver cette fonctionnalité, laissez la simplement vide. Notez aussi que cette directive n'est disponible que depuis PHP 4.2.0.

Note

Le paramètre callback est disponible depuis PHP 4.2.0.

Si la variable délinéarisée est un objet, après avoir réussi à le reconstruire, PHP appelera automatiquement la méthode __wakeup() si elle existe.

Exemple avec la directive unserialize_callback_func

```
<?php
 $serialized_object='O:1:"a":1:{s:5:"value";s:3:"100";}';

 // directive unserialize_callback_func disponible depuis PHP 4.2.0
 ini_set('unserialize_callback_func','moncallback');
  // configure votre fonction de callback :  moncallbackfunction

 function moncallback($classname) {
   // Incluez simplement in fichier contenant votre définition de classe
   // vous saurez quelle classe grâce à $classname
 }
?>
```

Note

En PHP 3, les méthodes ne sont pas préservées lors de la délinéarisation d'un bojet. PHP 4 lève cette limitation, et stocke désormais les méthodes et les membres. Voyez le chapitre sur la Linéarisation des objets du chapitre Classes et Objets .

Exemple avec unserialize

```
<?php
// Ici, on utilise <function>unserialize</function> pour charger les données de sessions
// depuis la base de données, dans $session_data. Cet exemple complète
// celui fourni avec <function>serialize</function>.
$conn = odbc_connect ("webdb", "php", "chicken");
$stmt = odbc_prepare ($conn, "SELECT data FROM sessions WHERE id = ?");
$sqldata = array ($PHP_AUTH_USER);
if (!odbc_execute ($stmt, &$sqldata) || !odbc_fetch_into ($stmt, &$tmp)) {
    // si la préparation ou la lecture échoue, on crée un tableau vide
    $session_data = array();
} else {
    // les données sauvées sont dasn $tmp[0].
    $session_data = unserialize ($tmp[0]);
    if (!is_array ($session_data)) {
    // Erreur... initialisation à tableau vide
    $session_data = array();
    }
}
?>
```

Voir aussi serialize .

## 6.38.36 unset() : Détruit une variable

void **unset** ( mixed var , *mixed var* , ... )

unset détruit les variables var . Notez qu'en PHP 3, unset retournait toujours TRUE (en fait, la valeur entière 1). unset n'est plus une véritable fonction : c'est une structure du langage, ce qui fait qu'elle ne retourne pas de valeur. Lire la valeur retournée par unset (dans une variable, par exemple), retourne une erreur d'analyse.

Exemple avec unset

```
<?php
// Destruction d'une seule variable
unset ($foo);
// Destruction d'un élément de tableau
unset ($bar['quux']);
// Destruction de plusieurs variables
unset ($foo1, $foo2, $foo3);
?>
```

Le comportement de unset à l'intérieur d'une fonction peut varier suivant le type de variable que vous voulez détruire.

Si une variable globale est détruite avec unset depuis une fonction, seule la variable locale sera détruite. Le variable globale gardera la valeur acquise avant l'appel à unset .

Exemple avec unset dans un contexte de fonction

```
<?php
function destroy_foo() {
    global $foo;
    unset($foo);
}
$foo = 'bar';
destroy_foo();
echo $foo;
?>
```

L'exemple ci dessus affichera :

```
bar
```

Si une variable qui est passée par référence est détruite à l'intérieur d'une fonction, seule la variable locale sera détruite. La variable globale conservera la dernière valeur qu'elle avait avant l'appel de unset .

Exemple avec unset sur une référence

```
<?php
 function foo(&$bar) {
   unset($bar);
   $bar = "bla";
 }
 $bar = 'truc';
 echo "$bar\n";
 foo($bar);
 echo "$bar\n";
?>
```

L'exemple ci dessus va afficher :

```
truc
truc
```

Si une variable statique est détruite à l'intérieure d'une fonction unset détruira la référence à la variable statique, plutôt que la variable statique elle même.

Exemple avec unset avec une variable statique

```
<?php
 function foo() {
   static $a;
   $a++;
      echo "$a\n";
   unset($a);
 }
 foo();
 foo();
 foo();
?>
```

L'affichage du script ci−dessus donnera :

```
1
2
3
```

Si vous voulez détruire une variable globale, depuis une fonction, vous pouvez utiliser le tableau $globals :

Exemple avec unset avec une variable globale

```
<?php

function foo() {
    unset($GLOBALS['bar']);
}
$bar = "truc";
foo();
?>
```

Note

unset est une structure du langage et non pas une fonction.

Note

Comme ceci est une structure du langage, et non pas une fonction, il n'est pas possible de l'appeler avec les variables fonctions .

Voir aussi array_splice , isset et empty .

## 6.38.37 var_dump() : Affiche les informations d'une variable.

void **var_dump** ( mixed expression , *mixed expression* , ... )

var_dump retourne les informations structurées d'une variable, y compris son type et sa valeur. Les tableaux sont explorés recursivement, avec des indentations, pour mettre en valeur leur structure.

Array

Comparez var_dump et print_r .

Exemple avec var_dump

```
<PRE>
<?php
$a = array (1, 2, array ("a", "b", "c"));
var_dump ($a);

/* affichage :
array(3) {
  [0]=>
  int(1)
  [1]=>
  int(2)
  [2]=>
  array(3) {
    [0]=>
    string(1) "a"
    [1]=>
    string(1) "b"
```

```
    [2]=>
    string(1) "c"
  }
}

*/

$b = 3.1;
$c = TRUE;
var_dump($b,$c);

/* affichage :
float(3.1)
bool(true)

*/
?>
</PRE>
```

## 6.38.38 var_export() : Retourne le code PHP utilisé pour générer une variable

mixed **var_export** ( mixed expression , *bool return* )

var_export retourne des données structurés sur la variable expression . C'est le même principe que var_dump mais le résultat retourné est du code PHP.

Par défaut, le code sera affiché, mais vous pouvez retourner ce code dans une variable en passant TRUE pour le paramètre de return .

Comparez var_export avec var_dump .

Exemple avec var_export

```
 <?php

$a = array (1, 2, array ("a", "b", "c"));
var_export ($a);

?>
```

Cet exemple affiche :

```
array (
0 => 1,
1 => 2,
2 =>
array (
0 => 'a',
1 => 'b',
2 => 'c',
),
)
```

```
 <?php

$b = 3.1;
$v = var_export($b, TRUE);
echo $v; // 3.1

?>
```

# 6.39 WDDX

## 6.39.1 Introduction

Ces fonctions fonctionnent grâce à WDDX .

## 6.39.2 Pré−requis

Pour utiliser WDDX, vous devez installer la librairie EXPAT (qui est fournie avec la distribution d'Apache 1.3.7 ou plus récent).

## 6.39.3 Installation

Après avoir installé Expat, compilez PHP avec −−enable−wddx .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

## 6.39.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.39.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.39.6 Constantes prédefinies

Cette extension ne définit aucune constante.

## 6.39.7 Exemples

Notez bien que toutes les fonctions qui enregistrent des données, utilisent le premier élément d'un tableau pour savoir si ce tableau doit être enregistré sous la forme d'un tableau, ou d'une structure. Si le premier élément a une clé de type chaîne, le tableau sera enregistré sous la forme d'une structure, et sinon, sous la forme d'un tableau.

Enregistrer une valeur simple

```
<?php
 print wddx_serialize_value("Exemple de paquet de PHP à WDDX", "Paquet PHP");
?>
```

Cet exemple va produire le résultat suivant :

```
<wddxPacket version='1.0'><header><comment>Paquet PHP</comment></header><data><string>Exemple de paquet de PHP à WDDX</string></data></wddxPacket>
```

Utilisation de paquets incrémentaux

```
<?php
  $pi = 3.1415926;
  $packet_id = wddx_packet_start("PHP");
  wddx_add_vars($packet_id, "pi");

/* Supposons que $villes provient d'une base de données */
  $villes = array("Paris", "Montréal", "Lyon");
  wddx_add_vars($packet_id, "villes");

  $packet = wddx_packet_end($packet_id);
  print $packet;
?>
```

Cet exemple donnera :

```
<wddxPacket version='1.0'><header><comment>PHP</comment></header><data><struct><var name='pi'><number>3.1415926</number></var><var name='villes'><arr
```

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- wddx_add_vars
- wddx_deserialize
- wddx_packet_end
- wddx_packet_start
- wddx_serialize_value
- wddx_serialize_vars

## 6.39.9 wddx_deserialize() : Lire un paquet WDDX

mixed **wddx_deserialize** ( string packet )

wddx_deserialize prend la chaîne packet et la lit. Cette fonction retourne un résultat qui peut être une chaîne, un nombre ou un tableau. Notez que les structures sont lues sous la forme de tableaux associatifs.

## 6.39.10 wddx_packet_end() : Clos un paquet WDDX

string **wddx_packet_end** ( int packet_id )

wddx_packet_end clos un paquet WDDX repéré par son identifiant packet_id .

## 6.39.11 wddx_packet_start() : Commencer un nouveau paquet WDDX avec une structure

int **wddx_packet_start** ( *string comment* )

wddx_packet_start sert à créer un nouveau paquet WDDX, pour pouvoir y faire des ajouts incrémentaux de variables. Cette fonction prend un argument optionnel comment et retourne un identifiant de paquet, qui servira à d'autres fonctions. Elle va automatiquement créer une définition de structure dans le paquet, pour accueillir des variables.

## 6.39.12 wddx_serialize_value() : Enregistrer une valeur dans un paquet WDDX

string **wddx_serialize_value** ( mixed var , *string comment* )

wddx_serialize_value sert à créer un paquet WDDX à partir d'une seule valeur. Cette fonction prend la valeur de var , et un argument optionnel comment qui apparaîtra dans l'en−tête du paquet, et retourne un paquet WDDX.

## 6.39.13 wddx_serialize_vars() : Enregistrer plusieurs valeurs dans un paquet WDDX

string **wddx_serialize_vars** ( mixed var_name , *mixed ...* )

wddx_serialize_vars sert à créer un paquet WDDX avec une structure qui contient la représentation des variables passées en arguments.

wddx_serialize_vars prend un nombre variable d'arguments, chacun d'entre eux pouvant être une chaîne contenant le nom d'une variable, ou un tableau de chaîne de nom de variable, ou même d'autres tableaux.

wddx_serialize_vars

```
<?php
 $a = 1;
 $b = 5.5;
 $c = array("bleu", "orange", "violet");
 $d = "colors";
 $clvars = array("c", "d");
 print wddx_serialize_vars("a", "b", $clvars);
?>
```

L'exemple ci−dessus donnera :
<wddxPacket version='1.0'><header/><data><struct><var name='a'><number>1</number></var>
<var name='b'><number>5.5</number></var><var name='c'><array length='3'>
<string>bleu</string><string>orange</string><string>violet</string></array>
</var><var name='d'><string>colors</string></var></struct></data></wddxPacket>

# 6.40 Analyseur syntaxique XML

## 6.40.1 Introduction

Le langage XML (eXtensible Markup Language (Langage à Balises Etendu)) est un format structuré de données pour les échanges sur le web. C'est un standard défini par le consortium World Wide Web (W3C). Plus d'informations à propos du XML et des technologies afférentes sont accessibles (en anglais) http://www.w3.org/XML/ .

Cette extension PHP supporte la librairie expat de James Clark sous PHP. Cela vous permettra d'analyser mais pas de valider les documents XML. Il supporte trois types de codage différents, disponibles aussi sous PHP: US−ASCII , ISO−8859−1 et UTF−8 . UTF−16 n'est pas supporté.

Cette extension vous permet de créer des analyseurs XML puis de définir des **points d'entrée** pour chaque événement XML. Les analyseurs XML disposent de quelques paramétrages .

## 6.40.2 Pré−requis

Cette extension de PHP utilise expat , disponible à http://www.jclark.com/xml/expat.html . Le fichier Makefile livré avec expat ne construit pas par défaut de librairie : il faut utiliser la ligne suivante :

```
libexpat.a: $(OBJS)
   ar −rc $@ $(OBJS)
   ranlib $@
```

Les sources RPM de expat sont disponibles à http://sourceforge.net/projects/expat/ .

## 6.40.3 Installation

Ces fonctions sont activées par défaut, et utilisent la librairie expat fournie avec la distribution. Vous pouvez désactiver le support de XML en utilisant l'option de compilation −−disable−xml . Si vous compilez PHP comme module Apache et avec les versions 1.3.9 ou plus récent, PHP va automatiquement utiliser la librairie fournie par Apache. Si vous ne souhaitez pas utiliser la librairie expat fournie avec PHP, il faut que vous compiliez PHP avec l'option −−with−expat−dir=DIR , où DIR est le dossier d'installation de votre librairie expat.

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

## 6.40.4 Configuration à l'exécution

Cette extension ne définit aucune directive de configuration.

## 6.40.5 Types de ressources

#### 6.40.5.1 xml

La ressource xml est retournée par xml_parser_create et xml_parser_create_ns , et représente un analysueur XML à utiliser avec les autres fonctions de l'extension.

## 6.40.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*XML_ERROR_NONE (entier)*
*XML_ERROR_NO_MEMORY (entier)*
*XML_ERROR_SYNTAX (entier)*
*XML_ERROR_NO_ELEMENTS (entier)*
*XML_ERROR_INVALID_TOKEN (entier)*
*XML_ERROR_UNCLOSED_TOKEN (entier)*
*XML_ERROR_PARTIAL_CHAR (entier)*
*XML_ERROR_TAG_MISMATCH (entier)*
*XML_ERROR_DUPLICATE_ATTRIBUTE (entier)*
*XML_ERROR_JUNK_AFTER_DOC_ELEMENT (entier)*
*XML_ERROR_PARAM_ENTITY_REF (entier)*
*XML_ERROR_UNDEFINED_ENTITY (entier)*
*XML_ERROR_RECURSIVE_ENTITY_REF (entier)*
*XML_ERROR_ASYNC_ENTITY (entier)*
*XML_ERROR_BAD_CHAR_REF (entier)*
*XML_ERROR_BINARY_ENTITY_REF (entier)*
*XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF (entier)*
*XML_ERROR_MISPLACED_XML_PI (entier)*
*XML_ERROR_UNKNOWN_ENCODING (entier)*
*XML_ERROR_INCORRECT_ENCODING (entier)*
*XML_ERROR_UNCLOSED_CDATA_SECTION (entier)*
*XML_ERROR_EXTERNAL_ENTITY_HANDLING (entier)*
*XML_OPTION_CASE_FOLDING (entier)*
*XML_OPTION_TARGET_ENCODING (entier)*
*XML_OPTION_SKIP_TAGSTART (entier)*
*XML_OPTION_SKIP_WHITE (entier)*

## 6.40.7 gestionnaires d'évènements

Les gestionnaires d'évènements XML sont:

| Fonction PHP de configuration du gestionnaire | Description de l'événement |
|---|---|
| xml_set_element_handler | Un événement est généré à chaque fois que l'analyseur XML rencontre une balise de début ou de fin. Deux gestionnaires sont disponibles : un pour le début, et un pour la fin. |
| xml_set_character_data_handler | "Character data" correspond grosso modo à tout ce qui n'est pas une balise XML, y compris les espaces entre les balises. Notez bien que l'analyseur XML n'ajoute ou n'efface aucun espace, et que c'est à l'application (c'est-à-dire vous) de décider de la signification de ces espaces. |
| xml_set_processing_instruction_handler | Les programmeurs PHP sont habitués aux instructions exécutables (processing instructions ou PIs). <?php ?> est une instruction exécutable où php est appelé programme cible. Ces instructions sont gérées de manière spécifiques, (sauf le programme cible, qui est réservé à XML). |
| xml_set_default_handler | Tout ce qui n'a pas trouvé de gestionnaire est transmis au gestionnaire par défaut. Vous retrouverez par exemple, les déclarations de type de document dans ce gestionnaire. |
| xml_set_unparsed_entity_decl_handler | Ce gestionnaire est appelé pour gérer les déclaration des entités non analysés. |
| xml_set_notation_decl_handler | Ce gestionnaire est appelé pour gérer les notations. |

| | |
|---|---|
| xml_set_external_entity_ref_handler | Ce gestionnaire est appelé lorsque l'analyseur XML trouve une référence à un fichier externe. Cela peut être un fichier, ou une URL. Reportez−vous à entité externe pour un exemple. |

## 6.40.8 Problèmes de casse

Les fonctions de gestion des balises peuvent rencontrer des balises en minuscule, majuscule ou encore dans un mélange des deux. En XML, la procédure standard est d' "identifier les séquences de caractère qui ne sont pas reconnues comme majuscule, et de les remplacer par leur équivalent majuscule". En d'autres termes, XML met toutes lettres en majuscules.

Par défaut, tous les noms des éléments qui sont transmis aux fonctions de gestion sont mises en majuscule. Ce comportement est contrôlé par l'analyseur XML, et peut être lu et modifié avec les fonctions respectives xml_parser_get_option et xml_parser_set_option , respectivement.

## 6.40.9 Codes d'erreurs

Les constantes suivantes sont définies comme des codes d'erreurs XML : (retournée par xml_parse )

- XML_ERROR_NONE
- XML_ERROR_NO_MEMORY
- XML_ERROR_SYNTAX
- XML_ERROR_NO_ELEMENTS
- XML_ERROR_INVALID_TOKEN
- XML_ERROR_UNCLOSED_TOKEN
- XML_ERROR_PARTIAL_CHAR
- XML_ERROR_TAG_MISMATCH
- XML_ERROR_DUPLICATE_ATTRIBUTE
- XML_ERROR_JUNK_AFTER_DOC_ELEMENT
- XML_ERROR_PARAM_ENTITY_REF
- XML_ERROR_UNDEFINED_ENTITY
- XML_ERROR_RECURSIVE_ENTITY_REF
- XML_ERROR_ASYNC_ENTITY
- XML_ERROR_BAD_CHAR_REF
- XML_ERROR_BINARY_ENTITY_REF
- XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF
- XML_ERROR_MISPLACED_XML_PI
- XML_ERROR_UNKNOWN_ENCODING
- XML_ERROR_INCORRECT_ENCODING
- XML_ERROR_UNCLOSED_CDATA_SECTION
- XML_ERROR_EXTERNAL_ENTITY_HANDLING

## 6.40.10 Codage des caractères

L'extension XML de PHP supporte les caractères Unicode grâce à différents codages. Il y a deux types de codages de caractères : le codage à la source et le codage à la cible. PHP utilise le UTF−8 comme représentation interne.

L'encodage à la source est effectué lors de l'analyse du fichier par XML. Lors de la création d'un analyseur XML ), un type de codage à la source doit être spécifié (et il ne pourra plus être modifié jusqu'à la destruction de l'analyseur). Les codages supportés sont : ISO−8859−1 , US−ASCII et

UTF−8 . Les deux derniers sont des codages à un seul octet, c'est−à−dire que les caractères sont représentés sur un seul octet. UTF−8 peut représenter des caractères composés par un nombre variable de bits (jusqu'à 21), allant de 1 à quatre octets. Le codage par défaut utilisé par PHP ISO−8859−1 .

Le codage à la cible est effectué lorsque PHP transfert les données aux gestionnaires XML. Lorsqu'un analyseur est créé, le codage à la cible est spécifié de la même façon que le codage à la source, mais il peut être modifié à tout moment. Le codage à la cible affectera les balises, tout comme les données brutes, et les noms des instructions exécutables.

Si l'analyseur XML rencontre un caractère qu'il ne connaît pas (hors limite, par exemple), il retournera une erreur.

Si PHP rencontre un caractère dans le document XML analysé, qu'il ne peut pas représenter dans le codage à la cible choisi, le caractère sera remplacé par un point d'interrogation (cette attitude est susceptible de changer ultérieurement).

## 6.40.11 Exemples

Voici une liste d'exemple de code PHP qui analyse un document XML.

### 6.40.11.1 Exemple de structure XML

Ce premier exemple affiche la structure de l'élément de début dans un document avec indentation.

Afficher une structure XML

```
 <?php
$file = "donnees.xml";
$depth = array();

function debutElement($parser, $name, $attrs) {
    global $depth;
    for ($i = 0; $i < $depth[$parser]; $i++) {
        print "  ";
    }
    print "$name\n";
    $depth[$parser]++;
}

function finElement($parser, $name) {
    global $depth;
    $depth[$parser]--;
}

$xml_parser = xml_parser_create();
xml_set_element_handler($xml_parser, "debutElement", "finElement");
if (!($fp = fopen($file, "r"))) {
    die("Impossible d'ouvrir le fichier XML");
}

while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("erreur XML : %s à la ligne %d",
                    xml_error_string(xml_get_error_code($xml_parser)),
                    xml_get_current_line_number($xml_parser)));
    }
}
xml_parser_free($xml_parser);
?>
```

### 6.40.11.2 Transtypage XML −> HTML

Transtypage XML −> HTML

Cet exemple remplace les balises XML d'un document par des balises HTML. Les éléments inconnus seront ignorés. Bien entendu, cet exemple sera appliqué à un type précis de fichiers

XML.

```
 <?php
$file = "data.xml";
$map_array = array(
    "BOLD"     => "B",
    "EMPHASIS" => "I",
    "LITERAL"  => "TT"
);

function startElement($parser, $name, $attrs) {
    global $map_array;
    if ($htmltag == $map_array[$name]) {
        print "<$htmltag>";
    }
}

function endElement($parser, $name) {
    global $map_array;
    if ($htmltag == $map_array[$name]) {
        print "</$htmltag>";
    }
}

function characterData($parser, $data) {
    print $data;
}

$xml_parser = xml_parser_create();
// Utilisons la gestion de casse, de manière à être sûrs de trouver la balise dans $map_array
xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, true);
xml_set_element_handler($xml_parser, "startElement", "endElement");
xml_set_character_data_handler($xml_parser, "characterData");
if (!($fp = fopen($file, "r"))) {
    die("Impossible de trouver le fichier XML");
}

while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("erreur XML : %s à la ligne %d",
                    xml_error_string(xml_get_error_code($xml_parser)),
                    xml_get_current_line_number($xml_parser)));
    }
}
xml_parser_free($xml_parser);
?>
```

### 6.40.11.3 Entité externe

Cet exemple exploite les références externes de XML : il est possible d'utiliser un gestionnaire d'entité externe pour inclure et analyser les documents, tous comme les instructions exécutables peuvent servir à inclure et analyser d'autres documents, et aussi fournir une indication de confiance (voir plus bas).

Le document XML qui est utilisé dans cet exemple est fourni plus loin dans l'exemple ( xmltest.xml et xmltest2.xml ).

Entité externe

```
 <?php
$file = "xmltest.xml";

function trustedFile($file) {
    // only trust local files owned by ourselves
    if (!eregi("^([a-z]+)://", $file)
         && fileowner($file) == getmyuid()) {
            return true;
    }
    return false;
}

function startElement($parser, $name, $attribs) {
    print "<<font color=\"#0000cc\">$name</font>";
    if (sizeof($attribs)) {
        while (list($k, $v) = each($attribs)) {
            print " <font color=\"#009900\">$k</font>=\"<font
                    color=\"#990000\">$v</font>\"";
        }
    }
    print ">";
}
```

```
function endElement($parser, $name) {
    print "</<font color=\"#0000cc\">$name</font>>";
}

function characterData($parser, $data) {
    print "<b>$data</b>";
}

function PIHandler($parser, $target, $data) {
    switch (strtolower($target)) {
        case "php":
            global $parser_file;
            // If the parsed document is "trusted", we say it is safe
            // to execute PHP code inside it.  If not, display the code
            // instead.
            if (trustedFile($parser_file[$parser])) {
                eval($data);
            } else {
                printf("Untrusted PHP code: <i>%s</i>",
                         htmlspecialchars($data));
            }
            break;
    }
}

function defaultHandler($parser, $data) {
    if (substr($data, 0, 1) == "&" && substr($data, -1, 1) == ";") {
        printf('<font color="#aa00aa">%s</font>',
                 htmlspecialchars($data));
    } else {
        printf('<font size="-1">%s</font>',
                 htmlspecialchars($data));
    }
}

function externalEntityRefHandler($parser, $openEntityNames, $base, $systemId,
                                  $publicId) {
    if ($systemId) {
        if (!list($parser, $fp) = new_xml_parser($systemId)) {
            printf("Could not open entity %s at %s\n", $openEntityNames,
                   $systemId);
            return false;
        }
        while ($data = fread($fp, 4096)) {
            if (!xml_parse($parser, $data, feof($fp))) {
                printf("erreur XML : %s à la ligne %d lors de l'analyse de l'entité %s\n",
                       xml_error_string(xml_get_error_code($parser)),
                       xml_get_current_line_number($parser), $openEntityNames);
                xml_parser_free($parser);
                return false;
            }
        }
        xml_parser_free($parser);
        return true;
    }
    return false;
}

function new_xml_parser($file) {
    global $parser_file;

    $xml_parser = xml_parser_create();
    xml_parser_set_option($xml_parser, XML_OPTION_CASE_FOLDING, 1);
    xml_set_element_handler($xml_parser, "startElement", "endElement");
    xml_set_character_data_handler($xml_parser, "characterData");
    xml_set_processing_instruction_handler($xml_parser, "PIHandler");
    xml_set_default_handler($xml_parser, "defaultHandler");
    xml_set_external_entity_ref_handler($xml_parser, "externalEntityRefHandler");

    if (!($fp = @fopen($file, "r"))) {
        return false;
    }
    if (!is_array($parser_file)) {
        settype($parser_file, "array");
    }
    $parser_file[$xml_parser] = $file;
    return array($xml_parser, $fp);
}

if (!(list($xml_parser, $fp) = new_xml_parser($file))) {
    die("Impossible d'ouvrir le fichier XML");
}

print "<pre>";
while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
        die(sprintf("Erreur XML : %s à la ligne %d\n",
                    xml_error_string(xml_get_error_code($xml_parser)),
                    xml_get_current_line_number($xml_parser)));
    }
}
print "</pre>";
```

```
print "parse complete\n";
xml_parser_free($xml_parser);

?>
```

```
xmltest.xml
 <chapter>
 <TITLE>Title </TITLE>
 <para>
  <informaltable>
   <tgroup cols="3">
    <tbody>
     <row><entry>a1</entry><entry morerows="1">b1</entry><entry>c1</entry></row>
     <row><entry>a2</entry><entry>c2</entry></row>
     <row><entry>a3</entry><entry>b3</entry><entry>c3</entry></row>
    </tbody>
   </tgroup>
  </informaltable>
 </para>

 <section id="about">
  <title>About this Document</title>
  <para>

   <?php print 'Salut! Ceci est PHP version '.phpversion(); ?>
  </para>
 </section>
</chapter>
```

Ce fichier est inclus depuis xmltest.xml :

```
xmltest2.xml
 <foo>
   <element attrib="value"/>

   <?php print "Ceci est du code PHP qui est exécuté."; ?>
</foo>
```

- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- gestionnaires d'évènements
- Problèmes de casse
- Codes d'erreurs
- Codage des caractères
- Exemples
- utf8_decode
- utf8_encode
- xml_error_string
- xml_get_current_byte_index
- xml_get_current_column_number
- xml_get_current_line_number
- xml_get_error_code
- xml_parse_into_struct
- xml_parse
- xml_parser_create_ns
- xml_parser_create
- xml_parser_free
- xml_parser_get_option
- xml_parser_set_option
- xml_set_character_data_handler
- xml_set_default_handler
- xml_set_element_handler
- xml_set_end_namespace_decl_handler

- xml_set_external_entity_ref_handler
- xml_set_notation_decl_handler
- xml_set_object
- xml_set_processing_instruction_handler
- xml_set_start_namespace_decl_handler
- xml_set_unparsed_entity_decl_handler

## 6.40.13 utf8_encode() : Convertit une chaîne ISO−8859−1 en UTF−8

string **utf8_encode** ( string data )

utf8_encode code la chaîne data au format UTF−8 , et retourne la version codée. UTF−8 est un mécanisme standardisé utilisé par Unicode pour coder les caractère de grande taille dans des flux d'octets. UTF−8 est transparent pour les caractères *ASCII* , il est auto−synchronisé (c'est à dire qu'un programme peut toujours savoir dans un flux d'octet où un caractère commence), et peut être utilisé pour faire des comparaisons de chaînes standard, comme pour le tri. PHP utilise l' UTF−8 pour coder les caractères jusqu'à 4 octets comme ceci :

| octets | bits | représentation |
|---|---|---|
| 1 | 7 | 0bbbbbbb |
| 2 | 11 | 110bbbbb 10bbbbbb |
| 3 | 16 | 1110bbbb 10bbbbbb 10bbbbbb |
| 4 | 21 | 11110bbb 10bbbbbb 10bbbbbb 10bbbbbb |

Chaque b représente un bit qui peut être utilisé pour enregistrer un caractère.

## 6.40.14 xml_error_string() : Lit le message d'erreur de l'analyseur XML

string **xml_error_string** ( int code )

*code*
Un message d'erreur, issu de xml_get_error_code .

xml_error_string retourne la chaîne avec un message textuel, décrivant l'erreur code , ou FALSE si aucune description n'a été trouvée.

## 6.40.15 xml_get_current_byte_index() : Retourne l'index de l'octet courant d'un analyseur XML

int **xml_get_current_byte_index** ( resource parser )

*parser*
Une référence sur un analyseur XML valide.

xml_get_current_byte_index retourne FALSE si parser n'est pas valide, ou sinon, retourne l'index de l'octet d'analyse courante de l'analyseur XML.

## 6.40.16 xml_get_current_column_number() : Retourne le nombre courant de colonne d'un analyseur XML.

int **xml_get_current_column_number** ( resource parser )

*parser*
Une référence sur un analyseur XML valide.

xml_get_current_column_number retourne FALSE si parser n'est pas valide, ou sinon, retourne le numéro de colonne courante de la ligne courante de l'analyseur, qui correspond à la position d'analyse courante de l'analyseur XML.

## 6.40.17 xml_get_current_line_number() : Retourne le numéro de ligne courant d'un analyseur XML.

int **xml_get_current_line_number** ( resource parser )

*parser*
Une référence sur un analyseur XML valide.

xml_get_current_line_number retourne FALSE si parser n'est pas valide, ou sinon, retourne le numéro de la ligne en cours d'analyse.

## 6.40.18 xml_get_error_code() : Retourne le nombre courant de colonne d'un analyseur XML

int **xml_get_error_code** ( resource parser )

xml_get_error_code retourne FALSE si parser n'est pas valide, ou sinon, retourne le numéro de colonne courante de la ligne courante de l'analyseur, qui correspond à la position d'analyse courante de l'analyseur XML.

## 6.40.19 xml_parse_into_struct() : Analyse une structure XML

int **xml_parse_into_struct** ( resource parser , string data , array &values , *array &index* )

xml_parse_into_struct analyse le fichier XML data , et le place dans deux tableaux : le premier index contient des pointeurs sur la position des valeurs correspondantes dans le tableau values . Ces deux paramètres sont passés par références.

Ci−dessous, vous trouverez un exemple qui illustre la structure des deux tableaux générés par la fonction. On utilise une balise simple note , placée dans une autre balise para . On analyse le tout, et on affiche la structure générée :

Exemple avec xml_parse_into_struct

```
<?php
 $simple = "<para><note>Simple Note</note></para>";
 $p = xml_parser_create();
 xml_parse_into_struct($p, $simple, $vals, $index);
 xml_parser_free($p);
 echo "Tableau d'index \n";
 print_r($index);
 echo "\nTableau de valeurs \n";
 print_r($vals);
?>
```

Lors de l'exécution du code, l'affichage sera :

```
Tableau d'index
Array
(
[PARA] => Array
(
[0] => 0
[1] => 2
)

[NOTE] => Array
(
[0] => 1
)

)

Tableau de valeurs
Array
(
[0] => Array
(
[tag] => PARA
[type] => open
[level] => 1
)

[1] => Array
(
[tag] => NOTE
[type] => complete
[level] => 2
[value] => Simple Note
)

[2] => Array
(
[tag] => PARA
[type] => close
[level] => 1
)

)
```

L'analyse événementielle (comme celle de expat), peut se révéler complexe lorsque le document XML est complexe. xml_parse_into_struct ne génère pas d'objet de type DOM, mais il génère plutôt des structures qui peuvent être parcourues à la façon d'un arbre. Considérons le fichier suivant, qui représente une petite base de données XML :

moldb.xml – Petite base de données moléculaire

```
<moldb>

    <molecule>
        <name>Alanine</name>
        <symbol>ala</symbol>
        <code>A</code>
        <type>hydrophobic</type>
    </molecule>

    <molecule>
        <name>Lysine</name>
        <symbol>lys</symbol>
        <code>K</code>
        <type>charged</type>
    </molecule>

</moldb>
```

Et maintenant, un code qui analyse le document, et génère les objets ad hoc :

parsemoldb.php : Analyse moldb.xml et crée un tableau d'objets moléculaires

```
<?php

class AminoAcid {
    var $name;  // nom aa
    var $symbol;    // symbole à trois lettres
    var $code;  // code à une lettre
    var $type;  // hydrophobique, chargé ou neutre

    function AminoAcid ($aa) {
        foreach ($aa as $k=>$v)
            $this->$k = $aa[$k];
    }
}

function readDatabase($filename) {
    // read the xml database of aminoacids
    $data = implode("",file($filename));
```

```
    $parser = xml_parser_create();
    xml_parser_set_option($parser,XML_OPTION_CASE_FOLDING,0);
    xml_parser_set_option($parser,XML_OPTION_SKIP_WHITE,1);
    xml_parse_into_struct($parser,$data,$values,$tags);
    xml_parser_free($parser);

    // loop through the structures
    foreach ($tags as $key=>$val) {
        if ($key == "molecule") {
            $molranges = $val;
            // each contiguous pair of array entries are the
             // lower and upper range for each molecule definition
            for ($i=0; $i < count($molranges); $i+=2) {
                    $offset = $molranges[$i] + 1;
                $len = $molranges[$i + 1] - $offset;
                $tdb[] = parseMol(array_slice($values, $offset, $len));
            }
        } else {
            continue;
        }
    }
    return $tdb;
}

function parseMol($mvalues) {
    for ($i=0; $i < count($mvalues); $i++)
        $mol[$mvalues[$i]["tag"]] = $mvalues[$i]["value"];
    return new AminoAcid($mol);
}

$db = readDatabase("moldb.xml");
echo "** Base d'objets AminoAcid :\n";
print_r($db);

?>
```

Après exécution de parsemoldb.php , la variable $db contient un tableau d'objets AminoAcid , et l'affichage le confirme :

```
** Base d'objets AminoAcid :
Array
(
[0] => aminoacid Object
(
[name] => Alanine
[symbol] => ala
[code] => A
[type] => hydrophobic
)

[1] => aminoacid Object
(
[name] => Lysine
[symbol] => lys
[code] => K
[type] => charged
)

)
```

## 6.40.20 xml_parse() : Commence l'analyse d'un fichier XML

bool **xml_parse** ( resource parser , string data , *bool is_final* )

*parser*
: une référence sur l'analyseur XML à utiliser.

*data*
: Une partie des données à analyser. Un document peut être analysé morceau par morceau, en appelant xml_parse plusieurs fois, tant que le paramètre isfinal est mis à TRUE pour le dernier morceau.

*isfinal (optional)*
: S'il vaut TRUE , data est la dernière partie à analyser.

Lorsqu'un document XML est analysé, les gestionnaires d'événements sont appelés aussi souvent que nécessaire, et retournent TRUE ou FALSE .

TRUE est retourné lorsque l'analyse a été concluante, et FALSE en cas d'échec, ou si parser n'est pas un analyseur valide. Lors d'un échec d'analyse, la cause de l'erreur peut être obtenue grâce aux fonctions xml_get_error_code , xml_error_string , xml_get_current_line_number , xml_get_current_column_number et xml_get_current_byte_index .

## 6.40.21 xml_parser_create_ns() : Crée un analyseur XML

resource **xml_parser_create_ns** ( *string encoding* , *string separator* )

xml_parser_create_ns crée un nouvel analyseur XML avec le support des espaces de noms, et retourne une ressource, qui pourra être utilisée avec les autres fonctions XML.

Avec un analyseur qui supporte les espaces de noms, les balises qui sont passées aux différentes fonctions de gestion seront constituées du nom de l'espace et du nom de balise, séparé par la chaîne seperator ou par ':' , par défaut.

Le paramètre optionnel encoding spécifie le jeu de caractères que le XML doit traiter. Les jeux de caractères supportés sont "ISO−8859−1" , qui est aussi la valeur par défaut si encoding est omis, "UTF−8" et "US−ASCII" .

Voir aussi xml_parser_create et xml_parser_free .

## 6.40.22 xml_parser_create() : Création d'un analyseur XML

resource **xml_parser_create** ( *string encoding* )

xml_parser_create crée un analyseur XML et retourne une référence sur cet analyseur pour qu'il puisse être utilisé ultérieurement par d'autres fonctions XML. xml_parser_create retourne FALSE en cas d'erreur.

*encoding (optionnel)*
: Le codage de caractère de l'analyseur. Les codages suivants sont supportés :
  - ISO−8859−1 (par défaut)
  - US−ASCII
  - UTF−8

Voir aussi xml_parser_create_ns et xml_parser_free .

## 6.40.23 xml_parser_free() : Détruit un analyseur XML

bool **xml_parser_free** ( resource parser )

*parser*
: Une référence sur un analyseur XML.

xml_parser_free retourne FALSE si parser n'est pas une référence valide, ou sinon, détruit l'analyseur et retourne TRUE .

## 6.40.24 xml_parser_get_option() : Lit les options d'un analyseur XML

mixed **xml_parser_get_option** ( resource parser , int option )

*parser*
Une référence sur un analyseur XML valide.
*option*
L'option demandée. Reportez−vous à xml_parser_set_option pour avoir la liste des options disponibles.

xml_parser_get_option retourne FALSE si parser n'est pas valide, ou sinon, retourne la valeur de l'option demandée.

Reportez−vous à xml_parser_set_option pour avoir la liste des options disponibles.

## 6.40.25 xml_parser_set_option() : Affecte les options d'un analyseur XML

bool **xml_parser_set_option** ( resource parser , int option , mixed value )

*parser*
Une référence vers un analyseur XML.
*option*
L'option à modifier. Voir ci−dessous :
*value*
La nouvelle valeur de l'option.

xml_parser_set_option retourne FALSE si parser n'est pas une référence valide sur un analyseur XML, ou si l'option n'a pas pu être modifiée. Sinon, l'option est effectivement modifiée, et la fonction retourne TRUE .

Les options suivantes sont disponibles :

| Option | Type de données | Description |
|---|---|---|
| XML_OPTION_CASE_FOLDING | entier | Contrôle la gestion de la casse des balises de cet analyseur XML. Par défaut, activé. |
| XML_OPTION_TARGET_ENCODING | string | Modifie le codage à la cible utilisé par cet analyseur XML. Par défaut, c'est celui qui a été spécifié lors de l'appel de xml_parser_create . Les codages supportés sont ISO−8859−1 , US−ASCII et UTF−8 . |

## 6.40.26 xml_set_character_data_handler() : Affecte les gestionnaires de texte littéral

bool **xml_set_character_data_handler** ( resource parser , callback handler )

xml_set_character_data_handler affecte les gestionnaires de début et de fin de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque xml_parse est appelé pour créer parser .

La fonction handler doit accepter deux paramètres : **handler** resource parser ( string data )

*parser*
Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle

cette fonction.

*data*

Le second paramètre, data , contient les caractères sous la forme d'une chaîne.

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_character_data_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.27 xml_set_default_handler() : Affecte le gestionnaire XML par défaut

bool **xml_set_default_handler** ( resource parser , callback handler )

xml_set_default_handler affecte le gestionnaire par défaut de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque xml_parse est appelé pour créer parser .

La fonction handler doit accepter deux paramètres: **handler** resource parser ( string data )

*parser*

Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

*data*

Le second paramètre, data , contient les caractères sous la forme d'une chaîne. Cela peut être une déclaration XML, un type de document, une entité ou d'autre données pour qui aucun gestionnaire n'est prévu.

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_default_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

Il n'est pas pour l'instant possible d'utiliser des objets pour servir de gestionnaire. Reportez−vous à xml_set_object pour utiliser l'analyseur XML depuis un objet.

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.28 xml_set_element_handler() : Affecte les gestionnaires de début et de fin de balise XML

bool **xml_set_element_handler** ( resource parser , callback start_element_handler , callback end_element_handler )

xml_set_element_handler affecte les gestionnaires de début et de fin de l'analyseur XML parser .

startelementhandler et endelementhandler sont des chaînes qui contiennent les noms de fonctions qui existent lorsque xml_parse est appelé pour créer parser .

La fonction startelementhandler doit accepter trois paramètres: **start_element_handler** resource parser ( string name , array attribs )

*parser*
Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

*name*
Le deuxième paramètre, name , contient le nom de l'élément qui a provoqué l'appel du gestionnaire. Si l'analyseur gère la casse , cet élément sera en majuscule.

*attribs*
Le troisième paramètre, attribs , contient un tableau associatif avec les attributs de l'éléments (s'il en existe). Les clés de ce tableau seront les noms des attributs, et les valeurs seront les valeurs correspondantes des attributs. Les noms des attributs seront mis en majuscule si l'analyseur gère la casse . Les valeurs des attributs seront intouchées.

L'ordre original des attributs peut être retrouvé en passant en revue le tableau attribs , avec la fonction each . La première clé sera la première clé du tableau.

La fonction endelementhandler doit accepter deux paramètres: **end_element_handler** resource parser ( string name )

*parser*
Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

*name*
Le second paramètre, name , contient le nom de l'élément qui a provoqué l'appel du gestionnaire. Si l'analyseur gère la casse , cet élément sera en majuscule.

Si un gestionnaire reçoit une chaîne vide, ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_element_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.29 xml_set_end_namespace_decl_handler() : Configure le gestionnaire XML de données

bool **xml_set_end_namespace_decl_handler** ( resource pind , callback handler )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.30 xml_set_external_entity_ref_handler() : Configure le gestionnaire XML de références externes

bool **xml_set_external_entity_ref_handler** ( resource parser , callback handler )

xml_set_external_entity_ref_handler affecte le gestionnaire d'entité externe de l'analyseur XML parser . handler et endelementhandler sont des chaînes qui contiennent les noms de fonction qui existent lorsque xml_parse est appelé pour créer le parser .

La fonction handler doit accepter 5 paramètres, et retourner un entier. Si la valeur retourné par le gestionnaire est FALSE (comme par exemple si aucune valeur n'est retournée), l'analyseur XML s'arrêtera, et la fonction xml_get_error_code retournera XML_ERROR_EXTERNAL_ENTITY_HANDLING . **handler** resource parser ( string open_entity_names , string base , string system_id , string public_id )

*parser*
: Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

*openentitynames*
: Le deuxième paramètre, openentitynames , est la liste de noms d'entité, séparés par des espaces. Ces entités sont accessibles à l'analyse par cet entité (y compris le nom de l'entité referencé).

*base*
: La meilleure base de résolution de l'identifiant système de cet entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

*systemid*
: Identifiant système pour cet entité externe.

*publicid*
: Le cinquième paramètre, publicid , est l'identifiant public, comme spécifié dans la déclaration d'entité, ou un chaîne vide, si aucune déclaration n'a été spécifiée. L'espace dans l'identifiant public sera normalisé comme spécifié dans les spécifications XML.

Si un gestionnaire reçoit une chaîne vide, ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_external_entity_ref_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon ou si parser n'est pas un analyseur.

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.31 xml_set_notation_decl_handler() : Configure le gestionnaire XML de notations

bool **xml_set_notation_decl_handler** ( resource parser , callback handler )

xml_set_notation_decl_handler affecte les gestionnaires de début et de fin de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque xml_parse est appelé pour créer parser .

Une notation est une partie du DTD du document, qui a le format suivant :

```
 <!NOTATION <parameter>name</parameter>
{ <parameter>systemid</parameter> | <parameter>publicid</parameter>?>
```

Reportez−vous à la section des spécifications XML 1.0 pour connaître les notations des entités externes.

La fonction handler doit accepter cinq paramètres : **handler** resource parser ( string notation_name , string base , string system_id , string public_id )

*parser*
: Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

*notationname*
: Le nom de la notation, name , comme précisé dans le format de notation ci−dessus.

*base*
: La meilleure base de résolution de l'identifiant système de cette entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

*systemid*
: Identifiant système pour cet entité externe.

*publicid*
: Identifiant public pour cet entité externe.

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_notation_decl_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon ou si parser n'est pas un analyseur.

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.32 xml_set_object() : Utilise un analyseur XML à l'intérieur d'un objet.

void **xml_set_object** ( resource parser , object object )

xml_set_object rend l'analyseur parser utilisable depuis un objet. Toutes les méthodes de callback, affectées par xml_set_element_handler , seront les méthodes de cet objet.

```
 <?php
class xml  {
    var $parser;

    function xml()
    {
        $this−>parser = xml_parser_create();

        xml_set_object($this−>parser, &$this);
        xml_set_element_handler($this−>parser, "tag_open", "tag_close");
        xml_set_character_data_handler($this−>parser, "cdata");
    }

    function parse($data)
    {
         xml_parse($this−>parser, $data);
    }

    function tag_open($parser, $tag, $attributes)
    {
         var_dump($parser, $tag, $attributes);
     }

    function cdata($parser, $cdata)
    {
        var_dump($parser, $cdata);
```

```
    }

    function tag_close($parser, $tag)
    {
        var_dump($parser, $tag);
    }

} // end of class xml

$xml_parser = new xml();
$xml_parser->parse("<A ID='hallo'>PHP</A>");
?>
```

## 6.40.33 xml_set_processing_instruction_handler() : Affecte les gestionnaires d'instructions exécutables.

bool **xml_set_processing_instruction_handler** ( resource parser , callback handler )

xml_set_processing_instruction_handler affecte le gestionnaire d'instructions exécutables de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque xml_parse est appelé pour créer parser .

Une instruction exécutable a la forme suivante :

| Structure d'instruction exécutable |
| --- |
| <? <br><br> <font face="courrier">target</font> <br><br> <font face="courrier">data</font> |

Vous pouvez mettre du code PHP entre ces balises, mais soyez conscient d'une des limitations des instructions exécutables de XML : la balise de fin d'instruction exécutable ( ?> ) ne peut être échappée, ce qui fait que cette séquence NE DOIT JAMAIS apparaître dans le code PHP placé dans le document PHP. Si un tel texte apparaît, la balise de fin d'instruction exécutable sera reconnue, et le reste du code sera considéré comme des données brutes (et donc, pas exécutées).

La fonction handler doit accepter trois paramètres: **handler** resource parser ( string target , string data )

*parser*
: Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

*target*
: Le second paramètre, target , contient l'application cible.

*data*
: Le troisième paramètre, data , contient le code sous la forme d'une chaîne.

Si un gestionnaire reçoit une chaîne vide, ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_processing_instruction_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.34 xml_set_start_namespace_decl_handler() : Configure le gestionnaire de caractères

bool **xml_set_start_namespace_decl_handler** ( resource pind , callback hdl )

| Attention |
| --- |
| Cette fonction n'est pas documentée et seule la liste des arguments est disponible. |

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

## 6.40.35 xml_set_unparsed_entity_decl_handler() : Affecte les gestionnaires d'entités non déclarés

bool **xml_set_unparsed_entity_decl_handler** ( resource parser , callback handler )

xml_set_unparsed_entity_decl_handler affecte les gestionnaires d'entité non déclaré de l'analyseur XML parser . handler est une chaîne qui contient le nom d'une fonction qui existe lorsque xml_parse est appelé pour créer parser .

Ce gestionnaire sera appelé si l'analyseur XML rencontre une déclaration d'entité externe avec une déclaration de NDATA, comme suit :

```
<!ENTITY <parameter>name</parameter> {<parameter>publicid</parameter> | <parameter>systemid</parameter>}
       NDATA <parameter>notationname</parameter>
```

Reportez−vous à la section des spécifications XML 1.0 pour connaître les notations des entités externes.

La fonction handler doit accepter six paramètres: **handler** resource parser ( string entity_name , string base , string system_id , string public_id , string notation_name )

*parser*
: Le premier paramètre, parser , est une référence sur l'analyseur XML qui appelle cette fonction.

*entityname*
: Le nom de l'entité qui va être définie

*base*
: La meilleure base de résolution de l'identifiant système de cette entité externe. Actuellement, ce paramètre est toujours une chaîne vide.

*systemid*
: Identifiant système pour cet entité externe.

*publicid*
: Identifiant public pour cet entité externe.

*notationname*
: Nom de la notation de cette entité. (Voir xml_set_notation_decl_handler ).

Si un gestionnaire reçoit une chaîne vide ou FALSE , c'est qu'il est en train d'être désactivé.

xml_set_unparsed_entity_decl_handler retourne TRUE si le gestionnaire est actif, et FALSE sinon, ou si parser n'est pas un analyseur.

| Note |
| --- |
| A la place d'un nom de fonction, un tableau contenant une référence d'objet et un nom de méthode peut aussi être utilisé. |

# 6.41 Compression Zlib

## 6.41.1 Introduction

Ce module donne accès aux fichiers compressés avec l'utilitaire gzip, grâce à des fonctions semblables aux fonctions d'accès aux fichiers , qui fonctionnent avec les fichiers compressés ou non (mais pas avec les sockets).

| Note |
| --- |
| La version 4.0.4 a introduit la version étendue de fopen pour les fichiesr compressés avec GZ. Vous pouvez utiliser le préfixe spécial zlib: dans les URL pour accéder directement et de manière transparente aux fichiers compressés par GZ, avec toutes les fonctions f* habituelles.<br><br>En version 4.3.0, ce préfixe spécial est remplacé par zlib:// pour éviter les ambiguités avec les fichiers contenant des deux points : .<br><br>Cette fonctionnalité requiert une librarie C qui fournisse la fonction fopencookie() . A ma connaissance, la librairie GNU libc est la seule à fournir cette fonctionnalité. |

## 6.41.2 Pré−requis

Ce module utilise les fonctions de la librairie zlib de Jean−loup Gailly et Mark Adler. Vous devez utiliser zlib version >= 1.0.9 avec ce module.

## 6.41.3 Installation

Le support de Zlib dans PHP n'est pas activé par défaut. Vous devez compiler PHP avec l'option −−with−zlib[=DIR] .

La version Windows de PHP dispose du support automatique de cette extension. Vous n'avez pas à ajouter de librairie supplémentaire pour disposer de ces fonctions.

| Note |
| --- |
| Le support intégré de zlib est disponible depuis PHP 4.3.0. |

## 6.41.4 Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini .

L'extension zlib offre l'option de compréssser de manière transparente les pages PHP à la volée, si le navigateur du visiteur le support. Voici donc les trois options à utiliser dans le php.ini .

| Nom | Par défaut | Modifiable |
| --- | --- | --- |
| zlib.output_compression | "Off" | PHP_INI_ALL |
| zlib.output_compression_level | "−1" | PHP_INI_ALL |
| zlib.output_handler | "" | PHP_INI_ALL |

Pour plus de détails sur les constantes PHP_INI_*, reportez−vous à ini_set .

Voici un éclaircissement sur l'utilisation des directives de configuration.

*zlib.output_compression booléen / entier*

Active ou pas la compression transparente des pages. Si cette option est mise à "On" dans php.ini ou dans la configuration Apache, les pages sont compressées si le navigateur envoie un en−tête "Accept−Encoding: gzip" ou "deflate". Les en−têtes "Content−Encoding: gzip" (respectivement "deflate") et "Vary: Accept−Encoding" sont ajoutés dans la page envoyée au navigateur.

Vous pouvez utiliser ini_set pour désactiver cette option dans votre script, si les en−têtes n'ont pas encore été envoyé. Si vous émettez un en−tête de type "Content−Type: image/", la compression sera désactivée aussi (afin de parer à un bug de Netscape). Vous pouvez le réactiver en ajoutant la ligne "ini_set('zlib.output_compression', 'On')" après que les en−têtes aient été envoyés.

Cette option accepte aussi des valeurs entières au lieu des booléens, "On"/"Off", ce qui vous permet de configurer la taille du buffer de sortie (par défaut, il vaut 4ko).

| Note |
| --- |
| output_handler doit être laissé à vide si cette option est activée. Sinon, vous devez utiliser zlib.output_handler. |

*zlib.output_compression_level integer*

Niveau de compression utilisé pour la compression de sortie.

*zlib.output_handler string*

Vous ne pouvez pas spécifier de gestionnaire de sortie supplémentaire si zlib.output_compression est activée. Cette configuration est la même que output_handler mais dans un ordre différent.

## 6.41.5 Types de ressources

Cette extension ne définit aucune ressource.

## 6.41.6 Constantes prédefinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

*FORCE_GZIP (entier)*
*FORCE_DEFLATE (entier)*

## 6.41.7 Exemples

Cet exemple ouvre un fichier temporaire, écrit une page de test dedans, puis affiche le contenu du fichier deux fois.

Exemple de l'extension Zlib

```
 <?php

$filename = tempnam ('/tmp', 'zlibtest').'.gz';
print "<html>\n<head></head>\n<body>\n<pre>\n";
$s = "C'est juste un test, test, test, test, test, test, test, test!\n";

// ouvre le fichier en écriture avec la compression maximale
$zp = gzopen($filename, "w9");

// écrit uen chaîne dans le fichier
gzwrite($zp, $s);
```

```
// ferme le fichier
gzclose($zp);

// ouvre le fichier en lecture
$zp = gzopen($filename, "r");

// lit 3 caractères
print gzread($zp, 3);

// affiche le reste du fichier, puis le ferme
gzpassthru($zp);
gzclose($zp);

print "\n";

// ouvre le fichier et affiche le fichier une seconde fois
if (readgzfile($filename) != strlen($s)) {
        echo "Erreur avec les fonctions zlib!";
}
unlink($filename);
print "</pre>\n</h1></body>\n</html>\n";

?>
```


- Introduction
- Pré−requis
- Installation
- Configuration à l'exécution
- Types de ressources
- Constantes prédefinies
- Exemples
- gzclose
- gzcompress
- gzdeflate
- gzencode
- gzeof
- gzfile
- gzgetc
- gzgets
- gzgetss
- gzinflate
- gzopen
- gzpassthru
- gzputs
- gzread
- gzrewind
- gzseek
- gztell
- gzuncompress
- gzwrite
- readgzfile
- zlib_get_coding_type


## 6.41.9 gzcompress() : Compresse une chaîne

string **gzcompress** ( string data , *int level* )

gzcompress compresse la chaîne data , avec le niveau de compression level , en utilisant le format de compression zlib. La fonction retourne FALSE si une erreur survient. Le paramètre optionnel level peut prendre des valeurs comprises entre 0 et 9.

Pour plus de détails sur l'algorithme de compression ZLIB, reportez−vous à la RFC " ZLIB

Compressed Data Format Specification version 3.3 " (RFC 1950).

| Note |
| --- |
| Ce **n'est pas** la même chose qu'une compression gzip, qui inclut aussi des en−têtes. Voyez la fonction gzencode pour cela. |

Voir aussi gzdeflate , gzinflate , gzuncompress et gzencode .

## 6.41.10 gzdeflate() : Compresse une chaîne avec la méthode DEFLATE

string **gzdeflate** ( string data , *int level* )

gzdeflate compresse la chaîne data avec la méthode DEFLATE, ou FALSE si une erreur survient. Le paramètre optionnel level peut prendre des valeurs comprises entre 0 (pas de compression) et 9 (compression maximale). Par défaut, le niveau de compression est le niveau de la librairie Zlib.

Pour plus de détails sur l'algorithme de compression DEFALTE, reportez−vous à la RFC DEFLATE Compressed Data Format Specification version 1.3 (RFC 1951).

Voir aussi gzinflate , gzcompress , gzuncompress et gzencode .

## 6.41.11 gzencode() : Crée une chaîne compressée

string **gzencode** ( string data , *int level* , *int encoding_mode* )

gzencode retourne la version compressée de la chaîne data , compatible avec le résultat de la commande gzip . Cette fonction retourne FALSE si une erreur survient. Le paramètre optionnel level peut prendre des valeurs comprises entre 0 (pas de compression) et 9 (compression maximale). Par défaut, le niveau de compression est le niveau de la librairie Zlib.

Vous pouvez aussi utiliser les constantes FORCE_GZIP (la valeur par défaut) ou FORCE_DEFLATE comme troisième paramètre optionnel encoding_mode . Si vous utilisez FORCE_DEFLATE , vous obtiendrez une chaîne réduite (deflated) avec les en−têtes Zlib ad hoc, mais sans la somme de contrôle finale CRC32.

| Note |
| --- |
| level a été ajouté en PHP 4.2. Avant cette version gzencode n'utilisait que le paramètre data et optionnellement encoding_mode . |

La chaîne résultante contient les en−têtes et la structure de données appropriée pour en faire un fichier .gz standard, c'est à dire :

Créer un fichier gzip

```
<?php
   $data = implode("", file("bigfile.txt"));
   $gzdata = gzencode($data, 9);
   $fp = fopen("bigfile.txt.gz", "w");
   fwrite($fp, $gzdata);
   fclose($fp);
?>
```

Pour plus de détails sur le format de fichier GZIP, reportez−vous à la RFC GZIP file format specification version 4.3 (RFC 1952).

Voir aussi gzcompress gzuncompress gzdeflate et gzinflate .

## 6.41.12 gzeof() : Indique si la fin d'un fichier compressé est atteinte

int **gzeof** ( resource zp )

gzeof retourne TRUE si le pointeur du fichier compressé zp a atteint la fin du fichier, ou si une erreur s'est produite. La fonction retourne FALSE sinon.

zp doit être un fichier compressé valide, retourné par la fonction gzopen .

## 6.41.13 gzfile() : Lit la totalité d'un fichier compressé

array **gzfile** ( string filename , *int use_include_path* )

gzfile est identique à la fonction readgzfile , hormis le fait que gzfile retourne le fichier sous forme de tableau.

Vous pouvez utiliser le second paramètre optionnel en lui donnant la valeur de 1 , si vous voulez rechercher le fichier dans l' include_path .

Voir aussi readgzfile et gzopen .

## 6.41.14 gzgetc() : Lit un caractère dans un fichier compressé

string **gzgetc** ( resource zp )

gzgetc retourne une chaîne contenant un caractère unique (décompressé), lu dans le fichier représenté par zp . gzgetc retourne FALSE si la fin du fichier est atteinte (contrairement à gzeof ).

zp doit être un fichier compressé valide, retourné par la fonction gzopen .

Voir aussi gzopen et gzgets .

## 6.41.15 gzgets() : Lit une ligne dans un fichier compressé

string **gzgets** ( resource zp , int length )

gzgets retourne une chaîne non compressée, de taille maximale length − 1 octets, lue dans le fichier représenté par zp . La lecture s'arrête lorsque length − 1 octets ont été lus, ou que la fonction a rencontré une nouvelle ligne ou la fin du fichier (le premier des trois qui survient).

Si une erreur survient, gzgets retourne FALSE .

zp doit être un fichier compressé valide, retourné par la fonction gzopen .

Voir aussi gzopen , gzgetc et fgets .

## 6.41.16 gzgetss() : Lit une ligne dans un fichier compressé, et supprime les balises

string **gzgetss** ( resource zp , int length , *string allowable_tags* )

gzgetss est identique à la fonction gzgets , hormis le fait que gzgetss supprime toutes les balises HTML et PHP dans le texte lu.

Vous pouvez utiliser le troisième paramètre optionnel allowable_tags pour indiquer les balises que vous souhaitez conserver.

| Note |
| --- |
| allowable_tags a été ajouté en PHP 3.0.13, et PHP 4.0b3. |

Voir aussi gzgets , gzopen et strip_tags .

## 6.41.17 gzinflate() : Décompresse une chaîne

string **gzinflate** ( string data , *int length* )

gzinflate prend la chaîne data compressée avec gzdeflate et retourne la chaîne originale, décompressée, ou bien FALSE en cas d'erreur. gzinflate retourne une erreur si la chaîne décompressée est plus de 256 fois plus longue que la chaîne compressée data ou plus grande que la taille de length octets, passé comme paramètre optionnel.

Voir aussi gzcompress . gzuncompress , gzdeflate et gzencode .

## 6.41.18 gzopen() : Ouvre un fichier compressé avec gzip

resource **gzopen** ( string filename , string mode , *int use_include_path* )

gzopen ouvre le fichier filename , compressé avec l'utilitaire gzip, en écriture ou en lecture. Le paramètre de mode mode prend les mêmes valeurs que dans la fonction fopen ("rb" ou "wb"), mais il accepte aussi un entier, qui représente un niveau de compression ("wb9") ou une heuristique : "f" pour filtrer les données "wb6f", 'h' pour la compression Huffman seulement "wb1h". (voir aussi la description de deflateInit2 dans zlib.h pour plus d'informations sur le paramètre d'heuristique).

gzopen peut aussi servir pour lire des fichiers qui ne sont pas compressés. Dans ce cas, gzread lira le fichier, sans décompression.

gzopen retourne une ressource de fichier. Tout ce que vous lirez à partir de ce pointeur sera décompressé à la volée de manière transparente.

Si l'ouverture du fichier échoue, gzopen retourne FALSE .

Vous pouvez utiliser le troisième paramètre optionnel en lui donnant la valeur de 1 , si vous voulez rechercher le fichier dans l' include_path .

Exemple avec gzopen

```
<?php
 $fp = gzopen ("/tmp/file.gz", "r");
?>
```

Voir aussi gzclose .

## 6.41.19 gzpassthru() : Affiche le contenu d'un fichier en une seule fois

int **gzpassthru** ( resource zp )

gzpassthru lit toutes les données du fichier zp jusqu'à la fin, puis affiche le résultat (décompressé).

Si une erreur survient, la fonction retourne FALSE , sinon, elle retourne TRUE .

zp doit être un fichier compressé valide, retourné par la fonction gzopen .

## 6.41.20 gzputs() : Ecrit une chaîne dans un fichier compressé Gzip

int **gzputs** ( resource zp , string str , *int length* )

gzputs est un alias de la fonction gzwrite , et lui est identique en tous points.

## 6.41.21 gzread() : Lecture de fichier compressé binaire

string **gzread** ( resource zp , int length )

gzread lit jusqu'à length octets dans le fichier compressé gzip, représenté par zp . La lecture s'arrête lorsque length octets (décompressé) ont été lus, ou bien que la fin du fichier a été atteinte.

Exemple avec gzread

```
<?php
// Lit le contenu d'un fichier dans une chaîne
  $filename = "/usr/local/something.txt.gz";
  $zd = gzopen($filename, "r");
  $contents = gzread($zd, 10000);
  gzclose($zd);
?>
```

Voir aussi gzwrite , gzopen , gzgets , gzgetss , gzfile et gzpassthru .

## 6.41.22 gzrewind() : Recommence la lecture du fichier

int **gzrewind** ( resource zp )

gzrewind replace le pointeur de position du fichier zp au début du fichier.

Si une erreur survient, la fonction retourne 0.

zp doit être un fichier compressé valide, retourné par la fonction gzopen .

Voir aussi gzseek et gztell .

## 6.41.23 gzseek() : Déplace le pointeur de lecture

int **gzseek** ( resource zp , int offset )

gzseek place le pointeur de lecture du fichier zp à la position offset , compté en octets depuis le début du fichier. C'est l'équivalent de la fonction gzseek( zp, offset, SEEK_SET ) , en langage C.

Si le fichier est ouvert en lecture, cette fonction est alors émulée, et se révèle extrêment lente. Si le fichier est ouvert en lecture, seuls les déplacements vers l'avant sont supportés : gzseek compresse alors une série de zéros jusqu'à la position indiquée.

gzseek retourne 0 en cas de réussite, sinon, elle retourne −1. Notez que placer le pointeur au delà de la fin du fichier n'est pas une erreur.

Voir aussi gztell et gzrewind .

## 6.41.24 gztell() : Lit la position courante du pointeur de lecture

int **gztell** ( resource zp )

gztell retourne la position du pointeur de lecture dans le fichier zp , c'est à dire sa position dans le fichier.

Si une erreur survient, gztell retourne FALSE .

zp doit être un fichier compressé valide, retourné par la fonction gzopen .

Voir aussi gzopen , gzseek et gzrewind .

## 6.41.25 gzuncompress() : Décompresse une chaîne compressée

string **gzuncompress** ( string data , *int length* )

gzuncompress prend la chaîne data , compressée avec la fonction gzcompress et retourne la chaîne originale, ou bien FALSE en cas d'erreur. gzuncompress retourne une erreur si la chaîne décompressée est plus de 256 fois plus longue que la chaîne compressée data ou plus grande que la taille de length octets, passé comme paramètre optionnel.

Voir aussi gzdeflate , gzinflate , gzcompress et gzencode .

## 6.41.26 gzwrite() : Ecrit dans un fichier compressé, en gzip

int **gzwrite** ( resource zp , string string , *int length* )

gzwrite écrit le contenu de la chaîne string dans le fichier compressé zp . Si le paramètre length est fourni, l'écriture s'arrêtera lorsque length octets décompressés on été écrit, ou à la fin de la chaîne string : le premier qui arrive.

Notez que si length est donné, la directive magic_quotes_runtime sera ignorée, et les slashs ne seront pas supprimés de la chaîne string .

Voir aussi gzread , gzopen et gzputs .

## 6.41.27 readgzfile() : Lit tout le fichier compressé

int **readgzfile** ( string filename , *int use_include_path* )

readgzfile lit le fichier filename , le décompresse et affiche le résultat.

readgzfile peut aussi servir à lire un fichier qui n'est pas compressé : dans ce cas, readgzfile va lire le fichier sans le décompresser.

readgzfile retourne le nombre d'octets décompressés qui ont été lus dans le fichier. Si une erreur survient, readgzfile retourne FALSE . Un message d'erreur sera alors affiché, à moins que la fonction n'ait été appelée avec @readgzfile .

Le fichier filename sera ouvert dans le système de fichier, et son contenu sera affiché.

Vous pouvez utiliser le troisième paramètre optionnel use_include_path en lui donnant la valeur de 1 , si vous voulez rechercher le fichier dans l' include_path .

Voir aussi gzpassthru , gzfile et gzopen .

## 6.41.28 zlib_get_coding_type() : Retourne la méhode de compression utilisée avec Gzip

string **zlib_get_coding_type** ( )

zlib_get_coding_type retourne la méhode de compression utilisée avec Gzip. Les valeurs possibles sont gzip , deflate et FALSE .

Voir aussi la directive zlib.output_compression .

# 7 Appendices

## 7.1 Histoire de PHP

L'évolution de PHP s'est faite en quelques années. Devenir un des langages les plus importants du web ne fut pas une évolution simple. Pour ceux que ça intéresse, voici comment PHP a évolué jusqu'à aujourd'hui. Les vieilles versions de PHP sont disponibles dans le musée PHP .

### 7.1.1 Histoire de PHP

#### 7.1.1.1 PHP /FI

PHP a pris la suite d'un langage plus ancien, appelé PHP /FI. PHP /FI a été créé par Rasmus Lerdorf, en 1995. C'était initialement une librairie de scripts Perl, dont il se servait pour noter les accès à CV en ligne. Il donna le nom de 'Personnal Home Page Tools' à cette librairie. Au fur et à mesure qu'il ajoutait de nouvelles fonctionnalités, Rasmus a transformé la librairie en une implémentation en C, capable de communiquer avec les bases de données, et de créer des applications dynamiques et simples pour le web. Rasmus décida alors de publier son code , pour que tout le monde puisse l'utiliser et en profiter. Cela appela aussi aux contributions et aux améliorations du code.

PHP /FI, qui signifie Personal Home Page / Forms Interpreter, (Home Page personnelle, Interpreteur de Formulaire), incluait plusieurs fonctionnalités de base que nous connaissons encore aujourd'hui. Il avait ces variables qui ressemblent au Perl, un système d'interprétation automatique des variables de formulaires, et une syntaxe qui s'intègre facilement dans HTML.La syntaxe elle−même était similaire à celle du Perl mais beaucoup plus limitée. Elle était simple et un peu incohérente.

En 1997, PHP /FI 2.0,la seconde version en langage C, avait déjà une audience estimée de plusieurs milliers d'utilisateurs dans le monde, et environs 50,000 noms de domaine indiquaient qu'ils avaient installé PHP . Cela représentait environs 1% des noms de domaines sur l'Internet. Même si le nombre de contributeur était plutôt élevé, PHP était toujours le projet d'un seul homme.

PHP /FI 2.0 fut publié officiellement en novembre 1997, après avoir passé l'essentiel de sa vie en version beta. Peu de temps après, une version alpha de PHP 3.0 était publié.

#### 7.1.1.2 PHP 3

PHP 3.0 fut la première version du langage tel que nous le connaissons actuellement. Il fut créé par Andi Gutmans et Zeev Suraski en 1997, sous forme de réécriture complète de PHP /FI, lorsqu'ils s'aperçurent que PHP /FI était sous performant pour leur application de commerce électronique. Dans un effort de coopération, et de compatibilité avec les anciennes versions de PHP /FI, Andi, Rasmus et Zeev décidèrent de coopérer et d'annoncer PHP 3.0 comme le successeur officiel de PHP /FI. Le développement de PHP /FI 2.0 fut complètement arrêté.

Une des améliorations notables de PHP 3.0 fut ses capacités d'extensions. En plus de fournir une solide infrastructure aux utilisateurs finaux, des accès à de nombreuses bases de données et protocoles, PHP 3.0 proposait une API modulaire, qui attira des douzaines de développeurs. Ceux−ci réalisèrent et partagèrent de nouvelles extensions. Sans doute, ce fut la clé du succès

retentissant de PHP 3.0. Les autres améliorations de PHP 3.0 furent le support de la syntaxe objet, et une syntaxe de langage plus robuste et cohérente.

Le nouveau langage fut publié sous un nouveau nom, qui indiquait clairement que le projet n'était plus un projet personnel, comme l'était PHP /FI 2.0. Il fut nommé ' PHP ' avec une nouvelle signification : ' PHP : Hypertext Preprocessor'. C'est un acronyme récursif, c'est à dire qu'il se définit lui−même. En français, cela donne : 'Le préprocesseur Hypertexte, c'est PHP '.

A la fin de 1998, PHP avait conquis une base de plusieurs dizaines de milliers d'utilisateurs, et des centaines de milliers de sites indiquaient qu'ils l'utilisaient. Au plus fort de son utilisation, PHP 3.0 était installé sur 10% du parc mondial de serveurs web mondial.

PHP 3.0 fut officiellement publié en Juin 1998, après 9 mois de tests.

#### 7.1.1.3 PHP 4

Durant l'hiver 1998, juste après la publication de PHP 3.0, Andi Gutmans et Zeev Suraski commencèrent la réécriture du moteur interne de PHP à la base. L'objectif était d'améliorer les performances de PHP avec les applications complexes, et améliorer la modularité du code. Ces applications étaient rendues possibles par la syntaxe de PHP 3.0, mais le logiciel n'était pas conçu pour supporter efficacement ces applications.

Le nouveau moteur, appelé 'Zend Engine' (combinaison des noms de Zeev et Andi), atteint ces objectifs avec succès, et la première version fut publiée vers la mi−1999. PHP 4.0, s'appuyant sur ce moteur et amélioré par un grand nombre de nouvelles fonctionnalités fut publié officiellement en mai 2000, presque 2 ans après son prédécesseur. En plus de performances nettement plus élevées, PHP 4.0 apportait le support de nombreux serveurs web, les sessions HTTP, la bufferisation de sortie, une sécurité accrue des informations visiteurs et plusieurs nouvelles structures de langage.

PHP 4 est actuellement la dernière version de PHP . Le projet de PHP 5.0 a déjà débuté, afin d'améliorer le moteur Zend, et y intégrer des capacités objets améliorées.

Actuellement, PHP est utilisé par des centaines de milliers de développeurs, et plusieurs millions de sites web indiquent qu'ils sont configurés avec PHP , ce qui représente environs 20% des noms de domaines sur Internet.

L'équipe de développement de PHP inclue des douzaines de développeurs, et d'autres équipes travaillent à des projets liés tels que PEAR ou la documentation.

#### 7.1.1.4 PHP 5

Le futur de PHP est essentiellement constitué par son coeur, le Zend Engine. PHP 5 incluera la nouvelle version du Zend Engine 2.0. Pour plus d'informations sur ce moteur, voyez sa page web .

### 7.1.2 Quelques projets liés à PHP

#### 7.1.2.1 PEAR

PEAR, signifie ' PHP Extension and Application Repository' (initialement, PHP Extension and Add−on Repository) est la première version de classe de base en PHP , et deviendra à terme le moyen privilégié de distribuer PHP et les modules C.

PEAR a été conçu durant les débats tenus lors des PHP Developers' Meeting (PDM) (Rencontres de développeurs PHP ), qui ont eu lieu en Janvier 2000 à Tel Aviv. Il a été crée à l'initiative de Stig S. Bakken, et dédié à sa première fille, Malin Bakken.

Depuis l'an 2000, PEAR a grossi jusqu'à devenir un projet significatif avec un groupe de développeur compétents, qui travaillent à mettre en place une librairie complète, réutilisable et commune à la communauté PHP entière. PEAR inclus une vaste sélection de classes de bases pour réaliser une couche d'abstraction de bases de données, du cache, des calculs mathématiques et des transactions de commerce électronique.

#### 7.1.2.2 Equipe d'assurance Qualité

L'équipe d'assurance Qualité de PHP a été mise en place durant l'été 2000, en réponse aux critiques reçues par PHP 3.0, qui n'était pas suffisamment testé sur des environnements de productions. L'équipe est constitué de d'un groupe de développeurs de haut niveau, qui ont une bonne connaissance des sources PHP . Ces développeurs passent le plus clair de leur temps à localiser et supprimer les bugs trouvés dans PHP . De plus, il y a bien d'autres membres du groupe PHP qui testent et fournissent un compte rendu fidèle de l'utilisation de PHP sur différentes plate formes.

#### 7.1.2.3 PHP −GTK

PHP −GTK est la solution PHP pour écrire des applications avec interface, coté client. Andrei Zmievski se rappelle de la création de PHP −GTK:

La programmation d'interface GUI a toujours été une passion pour moi, et je pensais que Gtk+ était une excellente librairie, hormis le fait qu'elle était écrite en C, ce qui était plutôt laborieux. Après avoir assisté aux implémentations de PyGtk et GTK−Perl, j'ai décidé de voir si PHP pouvait disposer de sa propre interface avec Gtk+, même minimale. J'ai commencé au mois d'août 2000, lorsque j'avais un peu plus de temps libre, et j'ai réalisé les premières expérimentations. Mon guide principal fut l'implémentation de PyGtk, qui était plutôt complète et avait une interface orientée objet. James Henstridge, l'auteur de PyGtk, fut d'une aide précieuse au démarrage du projet.

Réécrire toutes les interfaces de Gtk+ à la main était hors de question et j'ai pensé à réaliser un générateur de code, exactement comme PyGtk l'avait fait. Le générateur de code était un programme PHP qui lisait un ensemble de fichiers .defs, contenant les classes, constantes et méthodes Gtk+, puis générait le code d'interface de PHP . Ce qui ne pouvait pas être généré automatiquement était traités à la main, dans le fichier .overrides.

Travailler sur le générateur de code et sur l'infrastructure pris un peu de temps, car je n'avais pas beaucoup de temps à consacrer à PHP −GTK durant l'automne 2000. Après avoir montré PHP −GTK à Frank Kromann, il s'y intéressa, et commença à m'aider sur le générateur et l'implémentation Win32. Lorsque nous écrivîmes le premier programme 'Hello World', ce fut extrêmement excitant. Cela a pris encore quelques mois pour avoir une version initiale. Elle fut publiée en mars 2001. Cela a encore pris quelques mois pour que le projet soit dans une forme

présentable. Ce projet fut rapidement présenté sur SlashDot.

Anticipant que PHP serait un projet à part entière, j'ai installé des listes de diffusion et un serveur CVS indépendants, de même que le site gtk.php.net, avec l'aide de Colin Viebrock. La documentation demandait aussi de l'aide, et James Moore vient me porter une aide précieuse.

Depuis sa publication PHP −GTK a gagné en popularité. Nous avons notre propre équipe de documentation, et le manuel s'améliore de plus en plus. Des extensions ont été écrites pour PHP −GTK et des applications de plus en plus importantes sont concues.

### 7.1.3 Livres traitant de PHP

Comme PHP a évolué, il a été reconnu comme une plate forme de développement populaire. Un des signes qui ne trompe pas est le nombre de livres dédié à PHP a évolué paralellement.

Autant que nous le sachions, le premier livre dédié à PHP a été 'php− dynamische webauftritte professionell realisieren' − un livre en allemand, écrit par Egon Schmid, Christian Cartus et Richard Blume. Le premier livre en anglais sur PHP fut publié juste après : 'Core PHP Programming' par Leon Atkinson. Ces deux livres couvraient PHP 3.0.

Même si ces deux livres restent uniques dans leur genre, ils furent rapidement suivis par un grand nombre d'autres livres, de différents éditeurs. Il y a plus de 40 livres en anglais, 50 en allemand et plus de 20 en français. De plus, on peut maintenant trouver des livres sur PHP en espagnol, coréen, japonais et hébreux.

Clairement, ce nombre grandissant de livres, écrit par différents auteurs, publiés par différents éditeurs et leur disponibilité en différentes langues est un témoingnage du succès planétaire de PHP .

### 7.1.4 Publications sur PHP

Autant que nous le sachions, le premier article consacré à PHP dans un magasine papier fut publié dans un magazine Francais, vers la fin 1998 et couvrait PHP 3.0. Comme tous les livres, ce fut le premier d'une longue série d'articles publiés dans différents magazines.

Des articles sur PHP ont été publiés dans Dr. Dobbs, Linux Enterprise, Linux Magazine et bien d'autres. Des articles sur la migration d'applications PHP vers un environnement Windows existe dans la librarie MSDN de Microsoft .

## 7.2 Migration de PHP 3.0 à PHP 4.0

### 7.2.1 Ce qui a changé en PHP 4.0

PHP 4.0 et le moteur Zend ont significativement amélioré les performances et les possibilités de PHP, tout en assurant une compatibilité ascendante maximale. Le maximum de codes existants sous PHP 3.0 fonctionneront sous PHP 4.0. La migration de votre code de PHP 3.0 vers PHP 4.0 sera beaucoup plus facile que celle de PHP/FI 2.0 vers 3.0. Un grand nombre de scripts seront prêts sans modifications, mais il est bon que vous connaissiez les quelques différences, et que vous testiez vos applications avant d'effecteur le changement de cadre de production. Les

indications suivantes vous mettront sur la voie.

## 7.2.2 Utiliser PHP 3 et PHP 4 simultanément

Les systèmes d'exploitation récents disposent de capacités de versioning et de scoping. Ces fonctionnalités rendent possible l'installation de PHP 3 et PHP 4 comme modules Apache, simultanément.

Ceci a été fait sur les plate−formes suivantes :

- Linux avec les binutils récents (testé avec binutils 2.9.1.0.25)
- Solaris 2.5 ou plus récent
- FreeBSD (testé avec 3.2, 4.0)

Pour l'activer, configurez PHP 3 et PHP 4 pour qu'ils utilisent APXS ( −−with−apxs ) et les extensions nécessaires ( −−enable−versioning ). En dehors de cela, toutes les instructions d'installation habituelles s'appliquent. Par exemple :

Configuration pour utiliser simultanément PHP 3 et 4

```
$ ./configure \
 --with-apxs=/apache/bin/apxs \
 --enable-versioning \
 --with-mysql \
 --enable-track-vars
```

## 7.2.3 Migration des fichiers de configuration

Le fichier de configuration global, php3.ini , a été renommé en php.ini .

Pour les fichiers de configuration Apache, il y a eu des modifications plus importantes. Les types MIME reconnus par le module PHP ont été modifiés.

Activation des types Mimes PHP 3 et 4

```
 application/x-httpd-php3        -->    application/x-httpd-php
application/x-httpd-php3-source -->    application/x-httpd-php-source
```

Vous pouvez faire fonctionner vos deux versions de PHP avec le même fichier de configuration Apache (suivant la version qui est déjà compilée sur le serveur), en utilisant la syntaxe suivante :

Utilisation du même fichier de configuration Apache pour PHP 3 et 4

```
 AddType  application/x-httpd-php3        .php3
AddType  application/x-httpd-php3-source .php3s
AddType  application/x-httpd-php         .php
AddType  application/x-httpd-php-source  .phps
```

De plus, les directives de nom de PHP pour Apache ont aussi été modifiées.

Depuis PHP 4.0, il n'y a que 4 directives Apache qui se rapportent à PHP :

Directives Apache pour PHP 4

```
 php_value [PHP directive name] [value]
php_flag [PHP directive name] [On|Off]
php_admin_value [PHP directive name] [value]
php_admin_flag [PHP directive name] [On|Off]
```

Il y a deux différences entre les options Admin et les autres valeurs :

- Les options Admin ne peuvent être placées que des le fichier de configuration général (i.e.,

httpd.conf ).

- Les valeurs Standard ne peuvent pas contrôler certaines directives PHP. Par exemple, le safe mode (si vous pouviez modifier les configurations dans le fichier .htaccess , cela annulerait toute la sécurité du safe mode . A l'inverse, les valeurs Admin peuvent modifier n'importe quelle directive PHP.

Pour rendre le processus de transition plus agréable, PHP 4.0 est distribué avec des scripts qui convertissent automatiquement vos configuration Apache et vos fichiers .htaccess pour qu'ils puissent fonctionner aussi bien avec PHP 3 que PHP 4. Ces scripts ne convertissent PAS les lignes concernant les types MIME. Vous devez le faire vous−même.

Pour convertir votre fichier de configuration Apache, exécutez le script apconf−conv.sh (disponible dans le dossier scripts/apache/ ). Par exemple :

```
Conversion du fichier de configuration Apache pour PHP 4
~/php4/scripts/apache:#  ./apconf−conv.sh /usr/local/apache/conf/httpd.conf
```

Votre configuration originale sera sauvée dans le fichier httpd.conf.orig .

Pour convertir vos fichiers .htaccess , exécutez le script aphtaccess−conv.sh (disponible dans le dossier scripts/apache/ ). Par exemple :

```
Conversion des fichiers .htaccess pour PHP 4
~/php4/scripts/apache:#  find / −name .htaccess −exec ./aphtaccess−conv.sh {} \;
```

De la même façon, votre vieux fichier .htaccess sera sauvé sous le nom .htaccess.orig .

Les scripts de conversion requièrent l'installation préalable de awk.

## 7.2.4 Comportement de l'analyseur

L'analyse et l'exécution sont désormais deux étapes complètement dissociées, et l'exécution intervient lorsque le code, ainsi que tous ses inclusions et pré−requis, ont été complètement analysés et validés.

Une des nouvelles conditions introduites est que les fichiers inclus et requis ( include et require ) doivent être syntaxiquement complets. Vous ne pouvez plus répartir différents cas de votre code dans plusieurs fichiers. Vous ne pouvez plus commencer une boucle for ou while , une condition if ou un cas switch dans un fichier, et finir la boucle ou placer les cas else , endif , case ou break dans un autre fichier.

Il est toujours valable d'inclure du code supplémentaire depuis une boucle ou dans une condition, mais les accolades de bloc {...} , et les éléments de la boucle doivent être dans le même fichier ou chaîne évaluée avec eval .

Cela ne devrait pas perturber trop de monde, car étaler son code de cette façon est plutôt un style à éviter.

Une autre nouveauté est qu'il est plus possible de faire retourner une valeur avec un fichier requis ( require ) (mais c'est plutôt rare en PHP 3.0). Retourner une valeur avec un fichier inclus ( include ) est toujours possible.

## 7.2.5 Rapport d'erreur

### 7.2.5.1 Changement de configuration

Avec PHP 3.0, le niveau de rapport d'erreur était obtenu en ajoutant les constantes numériques de chaque niveau de rapport. Généralement, on utilisait 15 pour afficher toutes les erreurs, et 7 pour afficher toutes les erreurs hormis les alertes simples.

PHP 4.0 dispose d'un nombre significativement plus grand de niveaux de rapport d'erreur, et l'analyseur comprend désormais les constantes, lors des modifications.

Le niveau de rapport d'erreur doit désormais être explicitement configuré en supprimant les niveaux dont vous ne voulez pas du niveau maximal, grâce à la fonction de OU exclusif. Ça a l'air compliqué? Supposons que vous souhaitiez afficher toutes les erreurs, hormis les alertes de style, qui sont repérées par la constante : E_NOTICE. Il suffit d'ajouter la valeur suivante dans le fichier php.ini : error_reporting = E_ALL &~ ( E_NOTICE ) . Si vous voulez supprimer en plus les alertes, vous pouvez ajouter la constante appropriée, en la combinant avec l'opérateur OU logique '|': error_reporting= E_ALL &~ ( E_NOTICE | E_WARNING ) .

| Attention |
| --- |
| L'utilisation des vieilles valeurs de 7 et 15 est une très mauvaise idée, car elles ne prennent pas en compte les nouvelles classes d'erreurs, y compris certaines erreurs d'analyse. Cela peut conduire à de très étranges résultats, où le script n'affiche plus rien, malgré une erreur d'analyse.<br><br>Cela a conduit à un grand nombre de rapport d'erreur dans le passé, alors que les programmeurs n'étaient tout simplement pas capables de repérer l'accolade manquante, car l'analyseur avait la consigne de cacher ces erreurs.<br><br>Vérifier votre niveau d'erreur doit être le premier réflexe lorsque vos scripts meurent silencieusement. Le moteur Zend est considéré actuellement comme suffisamment mature pour ne plus causer ce genre de problème aujourd'hui. |

### 7.2.5.2 Nouveaux messages d'erreurs

Un grand nombre de scripts PHP PHP 3.0 utilisent des structures qui doivent être considérées comme un très mauvais style, même s'il effectue bien la tâche qui lui est affectée, car ils ne sont pas robustes. PHP 4.0 affichera de nombreux messages d'erreurs dans des situations où PHP 3.0 restera coi. La solution de facilité consiste à supprimer les messages de niveau E_NOTICE, mais c'est une meilleure idée de corriger le code à la place.

Le cas le plus courant qui génèrera des messages d'alertes est l'utilisation de constantes sans guillemets comme index de tableaux. PHP 3.0, comme PHP 4.0, finiront par les interpréter littéralement comme des chaînes, si aucune constante n'est définie à la place. Mais si jamais une telle constante est définie dans une autre partie du code, cela risque de produire des résultats étonnants. Cela peut devenir un trou de sécurité si un pirate arrive à redéfinir les constantes de telle manière que le script lui donne accès à un niveau de droits supérieur. PHP 4.0 vous signalera tout oubli de guillemets comme par exemple dans : $_SERVER[REQUEST_METHOD] . Modifier ce code en $_SERVER['REQUEST_METHOD'] rendra l'analyseur heureux, et améliorera grandement votre style et la sécurité du code.

PHP 4.0 vous signalera les variables ou les éléments de tableaux non initialisés.

## 7.2.6 Initialiseur

Les variables statiques et les membres de classes n'acceptent plus que des initialiseurs scalaires, tandis que PHP 3.0 acceptait aussi les expressions. Cela est dû, encore une fois, à la séparation de l'analyse et de l'exécution : aucun code ne peut être exécuté tant que l'analyse n'est pas terminée.

Pour les classes, il vaut mieux initialiser les membres dans le constructeur. Pour les variables statiques, une valeur fixe et simple est la seule chose qui viennent à l'esprit.

## 7.2.7 empty("0")

L'évolution la plus polémique est celle de empty . Une chaîne contenant seulement le caractère '0' (zéro) est maintenant considérée comme vide, alors qu'elle ne l'était pas en PHP 3.0.

Ce nouveau comportement prend tout son sens dans les applications web, puisque tous les résultats de champs de type input sont de type chaîne de caractères, même si un nombre est demandé, et ce, grâce aux capacités de conversion automatique de PHP. D'un autre côté, cela peut casser votre code d'une manière très subtile, menant droit au comportement erratique, difficilement repérable si vous ne savez pas ce qui vous attend.

## 7.2.8 Fonctions manquantes

Bien que PHP 4.0 dispose de nombreuses nouvelles fonctionnalités fonctions et extensions, vous vous rencontrer des fonctions PHP 3.0 qui manquent. Un petit nombre de fonctions de base n'ont pu être portées en PHP 4.0, maintenant que l'analyse et l'exécution ont été séparées. D'autres fonctions, et mêmes des extensions entières sont maintenant obsolètes, remplacées par de nouvelles fonctions plus puissantes ou plus efficaces. Certaines fonctions n'ont tout simplement pas été portées pour le moment ou pour des raisons de licences.

### 7.2.8.1 Fonctions manquantes pour des raisons de structure

Comme PHP 4.0 sépare l'analyse et l'exécution, il n'est plus possible de modifier le comportement de l'analyseur (intégré dans le moteur Zend) durant l'exécution, puisque toute l'analyse a eu lieu, et est terminée. La fonction short_tags() a cessé d'exister. Vous pouvez toujours modifier le comportement de l'analyseur avec le fichier php.ini .

Une autre fonctionnalité qui ne fait pas partie de PHP est le débogeur. Des débogueurs réalisés par des tiers sont disponibles, mais ne sont pas distribués avec PHP D'autres extensions du Zend Engine sont aussi disponibles auprès de divers sites et groupes.

### 7.2.8.2 Fonctions et extensions obsolètes

Les extensions Adabas et Solid n'existent plus. Elles sont intégrées dans les fonctions ODBC Unifié .

### 7.2.8.3 Nouveau statut pour unset

unset , bient que toujours disponible, a été implémenté légèrement différemment en PHP 4.0, et elle n'est plus vraiment une 'fonction'.

Cela n'a pas de conséquence directe sur le comportement de unset , mais utiliser cette fonction pour faire un test avec function_exists retournera FALSE comme il se doit avec les fonction bas niveau comme echo .

Une autre application pratique disparue est qu'il n'est plus possible d'appeler unset indirectement, c'est−à−dire que $func="unset"; $func($somevar) ne fonctionne plus.

## 7.2.9 Extensions PHP 3.0

Les extensions écrites pour PHP 3.0 ne fonctionnent plus avec PHP 4.0, ni les exécutables, ni les codes sources. Il n'est pas difficile de porter les extensions de PHP 3.0 à 4.0 si vous avez accès aux sources originales. Une description détaillée du processus de portage ne fait pas partie de cet appendice (pour le moment).

## 7.2.10 Substitution de variables dans les chaînes

PHP 4.0 dispose d'un nouveau mécanisme de substitution des variables dans les chaînes. Vous pouvez désormais accéder aux membres d'objets et aux tableaux multidimensionnels dans une chaîne.

Pour cela, il suffit de placer la variable entre accolades, le signe $ suivant immédiatement la première accolade : {$variable['a']}

Pour utiliser la valeur d'un membre d'objet dans une chaîne, il suffit d'écrire : "text {$obj−>member} text" ; alors qu'en PHP 3.0, il fallait faire comme ceci : "texte".$objet−>membre." texte" .

Cette technique rend le code beaucoup plus lisible, mais risque de poser des problèmes dans certains scripts PHP 3.0. Vous pouvez facilement traquer ce problème en recherche les séquences {$ dans votre code, et en les remplaçant par \{$ avec votre outil de remplacement habituel.

## 7.2.11 Cookies

PHP 3.0 avait la mauvaise habitude d'envoyer les cookies dans l'ordre inverse de celui du code (l'ordre des appels à setcookie ). PHP 4.0 rétablit l'ordre naturel en les envoyant dans le même ordre que vous même.

Cela peut aussi prendre à contre−pied certains programmes, mais ce comportement était tellement étrange qu'il méritait un tel traitement un jour ou l'autre, pour éviter d'autres problèmes ultérieurs.

## 7.2.12 Gestion des variables globales

Alors que la gestion des variables globales a été surtout étudiée pour être simple en PHP 3 et aux premières versions de PHP 4, la sécurité est devenue la priorité avec les dernières versions. Alors qu'en PHP 3, le code suivant fonctionnait sans problème, en PHP 4, il faut utiliser unset($GLOBALS["id"]); . Ce n'est qu'un des problème lié à la gestion des variables. Il faut toujours utiliser la variable $GLOBALS avec les nouvelles versions de PHP . Apprenez−en plus sur ce sujet dans la section Références globales .

```
Migration des variables globales
```

```
<?php
 $id = 1;
 function test() {
   global $id;
   unset($id);
 }
 test();
 echo($id);
  // Ceci va afficher 1 en PHP 4
?>
```

# 7.3 Migration de PHP/FI 2.0 à PHP 3.0

## 7.3.1 A propos des incompatibilités en 3.0

PHP 3.0 a été entièrement réécrit. Le nouvel analyseur syntaxique est beaucoup plus robuste et cohérent qu'en version 2.0. Il est aussi nettement plus rapide et utilise encore moins de mémoire. Cependant, ces améliorations n'ont pu être possibles qu'au prix de modifications parfois importantes, tant au niveau des syntaxes, qu'au niveau des fonctionnalités.

De plus, l'équipe de développement PHP a essayé de nettoyer la syntaxe et les sémantiques, ce qui a aussi causé quelques incompatibilités. A long terme, nous pensons que ces modifications seront pour le bien de tous.

Ce chapitre va tenter de vous montrer les incompatibilités que vous pourriez rencontrer lors de votre migration de PHP/FI 2.0 à PHP 3.0 et de vous aider à les résoudre. Les nouvelles fonctionnalités ne sont pas signalées, à moins que cela ne soit nécessaire.

Un programme de conversion automatique de vos vieux script PHP/FI 2.0 existe. Il est disponible dans le dossier de convertisseur de la distribution PHP 3.0. Ce programme ne fait que repérer les modifications de syntaxe et ne vous épargnera pas une relecture attentive du script.

## 7.3.2 old_function

L'instruction old_function vous permet de déclarer une fonction en utilisant une syntaxe identique à celle de PHP/FI2 (hormis le fait que vous devez remplacer ' function ' par ' old_function ').

C'est une fonctionnalité obsolète, qui ne doit être utilisée que par les convertisseurs PHP/FI2−>PHP 3.

| Attention |
| --- |
| Les fonctions déclarées comme old_function ne peuvent pas être appelées depuis le code interne PHP. Entre autres choses, cela signifique que vous ne pouvez pas les utiliser avec des fonctions comme usort , array_walk et register_shutdown_function . Vous pouvez contourner cette limitation en écrivant une fonction moderne, qui appelle cette vieille syntaxe old_function . |

## 7.3.3 Balises PHP

La première chose que vous remarquerez probablement est que les balises de PHP start et end ont changé. L'ancienne forme <? > a été remplacée par trois nouvelles balises possibles :

Migration: Migration: balises start/end

```
<?
 echo "Ceci est du code PHP/FI 2.0.\n";
```

```
>
```

Comme en version 2.0, PHP/FI accepte aussi cette variante :

```
Migration: premières nouvelles balises PHP
 <?
  echo "Ceci est du code PHP 3.0!\n";
?>
```

Notez bien que la balise de fin contient désormais un point d'interrogation et un signe supérieur ">". Cependant, si vous souhaitez utiliser XML sur votre serveur, vous aurez sûrement des problèmes avec cette variante, car PHP risque d'essayer d'exécuter des balises XML. A cause de ceci, la notation suivante a été ajoutée :

```
Migration: Nouvelles balises PHP
 <?php
  echo "Ceci est du code PHP 3.0!\n";
?>
```

Certains d'entre vous rencontrent des problèmes avec les éditeurs qui ne comprennent pas ce type de balises d'instruction : Microsoft FrontPage est l'un de ces éditeurs, et, pour contourner le problème, la variation suivante a été introduite :

```
Nouvelles balises PHP
 <script language="php">
  echo "Ceci est du code PHP 3.0!\n";
</script>
```

## 7.3.4 Syntaxe if..endif

La syntaxe alternative pour écrire des instructions if/elseif/else, avec if(); elseif(); else; endif; ne pouvait pas être conservée sans ajouter beaucoup de complexité à l'analyseur syntaxique. De ce fait, cette syntaxe à changée :

```
Migration: ancienne syntaxe if..endif
 <?php
  if ($foo);
    echo "oui\n";
  elseif ($bar);
    echo "presque\n";
  else;
    echo "non\n";
  endif;
?>
```

```
Migration: nouvelle syntaxe if..endif
 <?php
  if ($foo):
    echo "oui\n";
  elseif ($bar):
    echo "presque\n";
  else:
    echo "non\n";
  endif;
?>
```

Notez que les points virgules ont été remplacée par des points dans toutes les commandes, sauf pour la dernière expression (endif).

## 7.3.5 Syntaxe while

Tout comme pour if..endif, la syntaxe des boucles while..endwhile a changée :

```
Migration: ancienne syntaxe while..endwhile
 <?php
  while ($more_to_come);
    ...
  endwhile;
?>
```

Migration: nouvelle syntaxe while..endwhile

```
<?php
 while ($more_to_come):
   ...
 endwhile;
?>
```

Attention

Attention : si vous utilisez la vieille syntaxe while..endwhile en PHP 3.0, vous obtiendrez une boucle sans fin !

## 7.3.6 Types d'expression

PHP/FI 2.0 utilisait le membre à gauche dans les expressions, pour déterminer le type de résultat attendu. PHP 3.0 prend en compte les deux côtés de l'expression et cela peut produire des résultats inattendus avec les scripts 2.0.

Considérez les lignes suivantes:

```
<?php
 $a[0]=5;
 $a[1]=7;
 $key = key($a);
 while ("" != $key) {
   echo "$keyn";
   next($a);
 }
?>
```

En PHP/FI 2.0, cet exemple va afficher les indices des $a. En PHP 3.0, l'exemple ne va rien afficher du tout. La raison est qu'en PHP 2.0, puisque l'argument de gauche est de type chaîne, une comparaison de chaîne était effectuée et, effectivement, "" n'est pas "" , ce qui conduit la boucle à continuer. En PHP 3, lorsqu'une chaîne est comparée avec un entier, la comparaison est de type chaîne (la chaîne est convertie en entier). Ce qui revient à faire la comparaison entre ( atoi("") ) qui vaut 0 et la variable qui vaut aussi 0 et comme 0==0, la boucle ne commence même pas.

La correction de ceci est simple : il suffit de remplacer les commandes while par:

```
<?php
 while ((string)$key != "") {
?>
```

## 7.3.7 Les messages d'erreur ont changé

Les messages d'erreur en PHP 3.0 sont généralement plus précis que ceux de la version 2.0., mais vous ne verrez plus la portion de code qui a causé l'erreur. A la place, un numéro de ligne et un nom de fichier sera retourné.

## 7.3.8 Evaluation rapide des booléens

En PHP 3., l'évaluation des est court−circuité. Cela signifie dans une expression telle que ( (1 || test_me()) ), la fonction test_me() ne sera pas exécutée, car cela ne changera pas le résultat.

C'est une amélioration mineure, mais qui peut avoir des effets secondaires importants.

## 7.3.9 La valeur TRUE / FALSE comme retour de fonctions

La plupart des fonctions internes de PHP ont été réécrite pour qu'elle retourne TRUE en cas de succès, et FALSE en cas d'erreur, au contraire des fonctions qui retournaient 0 et −1 en PHP/FI 2.0. Le nouveau comportement est beaucoup plus logique, comme par exemple $fp = fopen("/your/file") or fail("fichier non trouvé!");. Etant donné que PHP/FI 2.0 n'a pas de règle claire à propos de ce que les fonctions doivent retourner en cas d'échec, la plupart des scripts devront probablement être vérifié manuellement, après avoir utilisé le convertisseur 2.0 à 3.0.

Migration depuis 2.0: valeur retournées, ancienne façon

```
<?php
 $fp = fopen($file, "r");
 if ($fp == -1);
   echo("Impossible d'ouvrir le fichier $file en lecture <br>\n");
 endif;
?>
```

Migration depuis 2.0: valeur retournées, nouvelle façon

```
<?php
 $fp = @fopen($file, "r") or
    print("Impossible d'ouvrir le fichier $file en lecture<br>\n");
?>
```

## 7.3.10 Diverses incompatibilités

- Le module PHP 3.0 pour Apache n'accepte plus les versions d'Apache antérieure à la version 1.2. Apache 1.2 ou plus récent est nécessaire.
- echo n'utilise plus de chaîne de formatage. Il faut utiliser printf à la place.
- En PHP/FI 2.0, un effet secondaire de l'implémentation faisait que $foo[0] était la même chose que $foo . Ce n'est plus vrai en PHP 3.0.
- Lire un tableau avec $array[] n'est plus valable.

  Ainsi, il n'est plus possible de passer en revue un tableau avec des boucles telles que $data = $array[] . Utilisez current et next à la place.

  Ainsi, $array1[] = $array2 n'ajoute pas les valeurs de $array2 à $array1 , mais crée un nouvel élément dans $array1 et y affecte $array2 comme dernier élément. Voir aussi les tableaux multidimensionnels.
- "+" n'est plus utilisable comme opérateur de concaténation de chaîne. A la place, il convertit les arguments en nombres et effectue une addition numérique. Utilisez "." à la place.

Migration depuis 2.0: concaténation de chaînes

```
<?php
 echo "1" + "1";
?>
```

En PHP 2.0 cela retournerait 11, en PHP 3.0 cela va retourner 2. A la place, faites :

```
<?php
 echo "1"."1";
?>
```

```
<?php
 $a = 1;
 $b = 1;
 echo $a + $b;
?>
```

Cela va afficher 2, tant en PHP 2.0 qu'en 3.0.

```
<?php
 $a = 1;
 $b = 1;
```

```
  echo $a.$b;
?>


Cela va afficher 11 en PHP 3.0.
```

# 7.4 Débuggeur PHP

## 7.4.1 A propos du débuggeur

PHP 3 inclut le support d'un serveur de débuggage.

PHP 4 n'inclut aucun support de débuggage. Vous pouvez toutefois utiliser un des débogeur externe. Le Studio Zend inclut un débogeur et il y a aussi des débogeurs gratuits comme http://dd.cron.ru/dbg/ , le Advanced PHP Debugger (APD) ou Xdebug

## 7.4.2 Utiliser le débuggeur PHP

Le débuggeur PHP sert à repérer les bugs récalcitrants. Le débuggeur fonctionne en se connectant à un port TCP à chaque démarrage de PHP. Tous les messages d'erreur seront envoyés sur cette connexion. Cette page est faite pour un "serveur de débuggage", qui peut fonctionner avec un IDE ou un éditeur programmable (tel que Emacs).

Comment paramétrer le débuggeur :

1. Réservez un port TCP pour le débuggeur dans le fichier de configuration ( debugger.port ) et activez−le ( debugger.enabled ).
2. Configurer un client TCP sur ce port (par exemple socket −l −s 1400 sous UNIX).
3. Dans votre code, placez la ligne "debugger_on( host )", où host est l'IP ou le nom de l'hôte qui supporte le client TCP .

Désormais, toutes les alertes, notes, ... seront envoyées sur la socket client, **même si vous avez désactivé le rapport d'erreur avec error_reporting** .

## 7.4.3 Protocole du débuggeur

Le protocole de débugage PHP 3 fonctionne ligne par ligne. Chaque ligne a un type **type** et plusieurs lignes composent un message Chaque message commence avec une ligne du type start et se termine avec une ligne de type end . PHP peut envoyer des lignes de plusieurs messages simultanément.

Voici un exemple de ligne à ce format :

```
date

time

host
(
pid
)

type
:
```

| |
|---|
| message−data |

*date*

Les dates sont au format ISO 8601 ( yyyy − mm − dd )

*time*

Les heures, y compris les micro−secondes: hh : mm : uuuuuu

*host*

Le nom DNS ou adresse IP de l'hôte qui a généré l'erreur.

*pid*

PID (process id) sur l'hôte host , qui a généré l'erreur.

*type*

Type de la ligne. Indique au programme client comment traiter les données suivantes :

| Nom | Signification |
|---|---|
| start | Indique au programme client que le message du débuggeur commence ici. Le contenu de data sera un type d'erreur, comme listé ci−dessous. |
| message | Le message d'erreur PHP 3. |
| location | Nom du fichier et numéro de ligne, où l'erreur est survenue. La première occurrence de location contiendra toujours la localisation générale. data contiendra : file : line . Il y a toujours une indication de location après un message et après chaque function . |
| frames | Nombre de frames dans le dump de la pile. S'il y a 4 frames, attendez vous à des informations sur 4 niveaux de fonctions. Si la ligne "frame" n'existe pas, la profondeur doit être 0 (une erreur est survenue au niveau général). |
| function | Nom de la fonction qui a généré l'erreur. Elle sera répétée à chaque niveau de la pile d'appel. |
| end | Indique au client que le message du débuggeur se termine ici. |

*data*

Ligne de données.

| Débuggeur | Interne PHP 3 |
|---|---|
| alerte (warning) | E_WARNING |
| erreur | E_ERROR |
| analyse (parse) | E_PARSE |
| note (notice) | E_NOTICE |
| core−error | E_CORE_ERROR |
| core−warning | E_CORE_WARNING |
| inconnue | (toutes les autres) |

Exemple de message du débuggeur

```
1998−04−05 23:27:400966 lucifer.guardian.no(20481) start: notice
1998−04−05 23:27:400966 lucifer.guardian.no(20481) message: Uninitialized variable
1998−04−05 23:27:400966 lucifer.guardian.no(20481) location: (
NULL
):7
1998−04−05 23:27:400966 lucifer.guardian.no(20481) frames: 1
1998−04−05 23:27:400966 lucifer.guardian.no(20481) function: display
1998−04−05 23:27:400966 lucifer.guardian.no(20481) location: /home/ssb/public_html/test.php3:10
1998−04−05 23:27:400966 lucifer.guardian.no(20481) end: notice
```

# 7.5 Développement PHP

## 7.5.1 Créer une fonction PHP 3

### 7.5.1.1 Prototypes de fonctions

Toutes les fonctions suivent le schéma suivant :

```
 void php3_foo(INTERNAL_FUNCTION_PARAMETERS) {
}
```

Même si votre fonction ne prend aucun argument, c'est comme cela qu'elle doit être appelée.

### 7.5.1.2 Arguments de fonctions

Les arguments sont toujours de type val. Ce type contient un membre de type union, qui indique le type reél d'argument. De cette façon, si votre fonction prend deux arguments, elle ressemble à ceci :

```
Argument de fonction de lecture
 pval *arg1, *arg2;
if (ARG_COUNT(ht) != 2 || getParameters(ht,2,&arg1,&arg2)==FAILURE) {
   WRONG_PARAM_COUNT;
}
```

NOTE: Les arguments peuvent être passé par valeur ou par référence. Dans les deux cas, vous devez passer *) à getParameters. Si vous voulez vérifier que le n−ième paramètre a été passé par référence ou par valeur, vous devez utiliser la fonction ParameterPassedByReference(ht,n). Elle retournera 1 ou 0.

Lorsque vous modifiez l'un des paramètres, qu'ils soient envoyés par référence ou par valeur, vous pouvez le passer à pval_destructor pour le réinitialiser, ou, s'il s'agit d'un tableau et que vous voulez ajouter des valeurs, vous pouvez utiliser des fonctions similaires à celles qui sont dans internal_functions.h, qui manipule return_value comme tableau.

Par ailleurs, si vous modifiez un paramètre en IS_STRING, assurez−vous que vous avez bien assigné un nouvelle chaîne avec estrdup() et une nouvelle longueur de chaîne. Seulement après, vous pouvez modifier le type en IS_STRING. Si vous modifiez une chaîne en IS_STRING ou IS_ARRAY vous devez d'abord appeler le destructeur pval_destructor.

### 7.5.1.3 Fonctions à nombre d'arguments variable

Une fonction peut prendre un nombre variable d'arguments. Si votre fonction peut prendre deux ou trois arguments, utiliser la syntaxe suivante :

```
Fonctions à nombre d'arguments variable
 pval *arg1, *arg2, *arg3;
int arg_count = ARG_COUNT(ht);
if (arg_count < 2 || arg_count > 3 ||
    getParameters(ht,arg_count,&arg1,&arg2,&arg3)==FAILURE) {
    WRONG_PARAM_COUNT;
}
```

### 7.5.1.4 Utiliser les arguments d'une fonction

Le type de chaque argument est stocké dans le champ pval. Ce champ peut prendre les valeurs suivantes :

| | |
|---|---|
| IS_STRING | Chaîne de caractères |
| IS_DOUBLE | Nombre à virgule flottante, en précision double |
| IS_LONG | Entier long |
| IS_ARRAY | Tableau |
| IS_EMPTY | Aucune |
| IS_USER_FUNCTION | ?? |
| IS_INTERNAL_FUNCTION | ?? (Si ce type ne peut pas être passé à une fonction, effacez−le) |
| IS_CLASS | ?? |
| IS_OBJECT | ?? |

Si vous recevez un argument d'un type, et que vous voulez l'utiliser avec un autre type, ou si vous voulez simplement forcer le type, vous pouvez utiliser l'une des fonctions de conversion suivantes :

```
 convert_to_long(arg1);
convert_to_double(arg1);
convert_to_string(arg1);
convert_to_boolean_long(arg1);
/* Si la chaîne est "" ou "0" elle devient 0, 1 sinon */
convert_string_to_number(arg1);
/* Convertit une chaîne en LONG ou DOUBLE suivant la chaîne */
```

Ces fonctions convertissent sur place : elles ne retournent aucune valeur.

La valeur de l'argument est enregistrée dans une union. Les membres sont :

- IS_STRING: arg1−>value.str.val
- IS_LONG: arg1−>value.lval
- IS_DOUBLE: arg1−>value.dval

#### 7.5.1.5 Gestion de la mémoire dans une fonction

Toute la mémoire nécessaire à une fonction doit être allouée avec emalloc() ou estrdup(). Ces fonctions ont le goût et l'odeur des fonctions C classiques malloc() et strdup(). La mémoire doit être libérée avec efree().

Il y a deux types de mémoire dans ce programme : la mémoire qui est retournée à l'analyseur, et la mémoire qui nécessaire pour le stockage temporaire dans la fonction. Lorsque vous assignez une chaîne dans une variable qui est retournée à l'analyseur, assurez−vous de bien allouer la mémoire avec emalloc() ou estrdup(). Cette mémoire ne doit JAMAIS être libérée, sauf si vous réécrivez votre original plus loin, dans la même fonction (mais ce n'est pas de la programmation propre).

Pour tous vos besoins en mémoire temporaire/permanante dont vous avez besoin dans vos fonctions/librairies, vous devez utiliser les fonctions emalloc(), estrdup() et efree(). Elles se comportent EXACTEMENT comme leurs homologues. Tout ce qui est créé avec emalloc() ou estrdup() doit être libéré avec efree() à un moment ou un autre, à moins que ce ne soit utile ailleurs dans le programme; sinon, il va y avoir une fuite de mémoire. La signification de "Elles se comportent EXACTEMENT comme leurs homologues" est que si vous libérez une variable qui n'a pas été créée avec emalloc() ou estrdup(), vous courez droit à au crash ("segmentation fault"). Soyez alors extrêmement prudent, et libérez toute votre mémoire inutilisée.

Si vous compilez avec "−DDEBUG", PHP 3 affichera la liste de tous les appels à emalloc() et estrdup() mais jamais à efree() lorsque celui−ci intervient dans un script spécifié.

#### 7.5.1.6 Affecter une variable dans la table des symboles

Un grand nombre de macros sont disponibles pour rendre plus facile l'insertion de variables dans la table des symboles :

- SET_VAR_STRING(name,value)
- SET_VAR_DOUBLE(name,value)
- SET_VAR_LONG(name,value)

Soyez prudent avec la constante SET_VAR_STRING. La valeur doit être placée dans une portion de mémoire créée avec malloc(), sinon le gestionnaire de mémoire essayera de libérer le pointeur plus tard. Ne passez aucune mémoire allouée statiquement à SET_VAR_STRING.

Les tables des symboles de PHP est une table de hash. A n'importe quel moment, &symbol_table est un pointeur sur la table principale, et active_symbol_table pointe sur la table actuellement utilisée. (ces deux tables peuvent être identiques au démarrage, ou différent, suivant que vous êtes dans une fonction ou non).

Les exemples suivants utilisent 'active_symbol_table'. Vous devriez la remplacer par &symbol_table si vous voulez travailler sur la table principale. De plus, les mêmes fonctions peuvent être appliquées à des tableaux, comme expliqué ci−dessous.

```
Vérification de l'existence de $foo dans la table des symboles
 if (hash_exists(active_symbol_table,"foo",sizeof("foo"))) {
 // existe...
} else {
 // n'existe pas
}
```

```
Rechercher la taille d'une variable dans la table des symboles
 hash_find(active_symbol_table,"foo",sizeof("foo"),&pvalue);
check(pvalue.type);
```

En PHP 3.0, les tableaux sont implémentés en utilisant les mêmes tables de hash que les variables. Cela signifie que les deux fonctions ci−dessus peuvent être appelées pour vérifier la présence de variables dans un tableau.

Si vous voulez définir un nouveau tableau dans la table des symboles, utilisez le code suivant.

D'abord, vous devez vérifier qu'il n'existe pas, avec hash_exists() ou hash_find().

Puis, initialisez le tableau :

```
Initialisation d'un tableau
 pval arr;
if (array_init(&arr) == FAILURE) { /*Initialiation échouée*/ };
hash_update(active_symbol_table,"foo",sizeof("foo"),&arr,sizeof(pval),NULL);
```

Ce code déclare un nouveau tableau, appelé $foo, dans la table de symbole. Ce tableau est vide.

Voici comment ajouter deux nouvelles entrées dans ce tableau :

```
Ajout d'entrées dans un tableau.
 pval entry;
entry.type = IS_LONG;
entry.value.lval = 5;
/* définit $foo["bar"] = 5 */
hash_update(arr.value.ht,"bar",sizeof("bar"),&entry,sizeof(pval),NULL);
/* définit $foo[7] = 5 */
hash_index_update(arr.value.ht,7,&entry,sizeof(pval),NULL);
/* définit la prochaine place libre dans $foo[],
 * $foo[8], qui sera 5 (comme en php2)
 */
hash_next_index_insert(arr.value.ht,&entry,sizeof(pval),NULL);
```

Si vous voulez modifier une valeur que vous avez inséré dans une table de hash, vous devez d'abord la lire dans la table. Pour éviter cette recherche, vous pouvez fournir une pval ** à la fonction d'ajout dans la table de hash, et elle modifiera la valeur à l'adresse pval *, avec la valeur donnée. Si cette valeur est NULL , (comme dans tous les exemples ci dessus), ce paramètre sera ignoré.

hash_next_index_insert() utiliser plus ou moins la même logique que "$foo[] = bar;" in PHP 2.0.

Si vous construisez un tableau, pour le retourner, vous pouvez l'initialiser comme ceci :

```
 if (array_init(return_value) == FAILURE) { échec...; }
```

puis ajouter les valeurs grâces aux macros:

```
 add_next_index_long(return_value,long_value);
add_next_index_double(return_value,double_value);
add_next_index_string(return_value,estrdup(string_value));
```

Bien sûr, si l'ajout n'est pas fait juste après l'initialisation, vous devrez d'abord rechercher le tableau :

```
 pval *arr;
if (hash_find(active_symbol_table,"foo",sizeof("foo"),(void **)&arr)==FAILURE)
{ introuvable... }
else
{ utilisez arr->value.ht... }
```

Notez que hash_find reçoit un pointeur sur un pointeur sur pval, et pas un pointeur sur pval.

Toutes les fonctions d'accès aux hash retourne TRUE (SUCCES) ou FALSE (FAILURE), excepté hash_exists(), qui retourne un booléen.

#### 7.5.1.7 Retourne une valeur simple

Un grand nombre de macros sont disponible pour simplifier le retour des valeurs.

La macro RETURN_* fixe la valeur de retour, et termine la fonction :

- RETURN
- RETURN_FALSE
- RETURN_TRUE
- RETURN_LONG(l)
- RETURN_STRING(s,dup) Si dup est TRUE , duplique la chaîne.
- RETURN_STRINGL(s,l,dup) retourne la chaîne (s) en spécifiant la longueur (l).
- RETURN_DOUBLE(d)

La macro RETVAL_* macros fixe la valeur de retour, mais ne termine pas la fonction.

- RETVAL_FALSE
- RETVAL_TRUE
- RETVAL_LONG(l)
- RETVAL_STRING(s,dup) Si dup est TRUE , duplique la chaîne
- RETVAL_STRINGL(s,l,dup) retourne la chaîne (s) en spécifiant la longueur (l).
- RETVAL_DOUBLE(d)

Les macros ci−dessus vont utiliser estrdup() sur les arguments passés. Cela vous permet de libérer tranquillement les arguments après avoir appelé cette fonction, ou bien, utiliser de la mémoire allouée statiquement.

Si votre fonction retourne un booléen de succès/erreur, utilisez toujours RETURN_TRUE et RETURN_FALSE respectivement.

#### 7.5.1.8 Retourner des valeurs complexes

Votre fonction peut aussi retourner des valeurs complexes, tels que des objets ou tableaux.

Retourner un objet:

1. Appeler object_init(return_value).
2. Remplissez les valeurs. Les fonctions utilisables sont listées ci dessous.
3. Eventuellement, enregistrez les fonctions pour cet objet. Afin de lire des valeurs de cet objet, la fonction doit lire dans "this", dans la table de symbole active active_symbol_table. Son type doit être IS_OBJECT, et c'est une table de hash basique. (i.e., vous pouvez utiliser les fonctions habituelles de .value.ht). L'enregistrement reél peut être fait comme suit :

```
add_method( return_value, function_name, function_ptr );
```

Les fonctions d'accès aux objets sont :

- add_property_long( return_value, property_name, l ) − Ajoute un membre nommé 'property_name', de type long, égal à 'l'
- add_property_double( return_value, property_name, d ) − Idem, ajoute un double
- add_property_string( return_value, property_name, str ) − Idem, ajoute une chaîne
- add_property_stringl( return_value, property_name, str, l ) − Idem, ajoute une chaîne de longueur 'l'.

Retournez un tableau :

1. Appelez array_init(return_value).
2. Remplissez les valeurs. Les fonctions disponibles sont listées ci−dessous.

Les fonctions utilisées pour accéder à un tableau sont :

- add_assoc_long(return_value,key,l) − Ajoute une entrée associative avec la clé 'key' et la valeur 'l', de type long
- add_assoc_double(return_value,key,d) − Ajoute une entrée associative avec la clé 'key' et la valeur 'l', de type double
- add_assoc_string(return_value,key,str,duplicate)
- add_assoc_stringl(return_value,key,str,length,duplicate) spécifie la taille d'une chaîne
- add_index_long(return_value,index,l) − Ajoute une entrée d'index index' avec la valeur 'l', de type long
- add_index_double(return_value,index,d)
- add_index_string(return_value,index,str)
- add_index_stringl(return_value,index,str,length) − spécifie la longueur de la chaîne.
- add_next_index_long(return_value,l) − ajoute une entrée tableau, dans le prochain offset libre, de longueur 'l', de type long
- add_next_index_double(return_value,d)
- add_next_index_string(return_value,str)
- add_next_index_stringl(return_value,str,length) − spécifie la taille d'une chaîne

#### 7.5.1.9 Utiliser la liste des ressources

PHP 3.0 dispose de standards pour traiter un certains nombre de ressources. Ils remplacent tous les listes de PHP 2.0.

Fonctions accessibles :

- php3_list_insert(ptr, type) − retourne l'identifiant 'id' de la nouvelle ressource insérée.
- php3_list_delete(id) − efface la ressource d'identifiant id
- php3_list_find(id,*type) − retourne le pointeur de la ressource d'identifiant id, et modifie le type 'type'

Typiquement, ces fonctions sont utilisées pour les pilotes SQL, mais elles peuvent servir n'importe quoi d'autre. Par exemple, conserver un pointeur de fichier.

La liste standard de code ressemble à ceci :

```
Ajouter une nouvelle ressource
 RESOURCE *resource;
/* ...alloue de la mémoire pour la ressource, et l'acquiert ... */
/* Ajoute la nouvelle ressource dans la liste */
return_value->value.lval = php3_list_insert((void *) resource, LE_RESOURCE_TYPE);
return_value->type = IS_LONG;
```

```
Utiliser une ressource existante
 pval *resource_id;
RESOURCE *resource;
int type;
convert_to_long(resource_id);
resource = php3_list_find(resource_id->value.lval, &type);
if (type != LE_RESOURCE_TYPE) {
    php3_error(E_WARNING,"resource index %d has the wrong type",resource_id->value.lval);
    RETURN_FALSE;
}
/* ...utiliser la ressource... */
```

```
Effacer une ressource existante
 pval *resource_id;
RESOURCE *resource;
int type;
convert_to_long(resource_id);
php3_list_delete(resource_id->value.lval);
```

Les types de ressources doivent être enregistré dans le fichier php3_list.h, dans l'énumération list_entry_type. En plus, il faut penser à ajouter une fonction de terminaison, pour chaque type de ressource défini, dans le fichier list.c, pour la fonction list_entry_destructor() (même si vous n'avez rien de particulier à faire lors de la terminaison, vous devez au moins ajouter un cas vide.

#### 7.5.1.10 Utiliser la table des ressources persistantes.

PHP 3.0 dispose d'une lieu de stockage des ressources persistantes (i.e., les ressources qui doivent être conservées d'un hit à l'autre). Le premier module a utiliser cette capacité a été MySQL, et mSQL suivi, ce qui fait que l'on peut se faire une impression du fonctionnement de cette fonction avec mysql.c. Les fonctions ressemblent à ceci :

- php3_mysql_do_connect
- php3_mysql_connect()
- php3_mysql_pconnect()

L'idée conductrice de ces modules est la suivante :

1. Programmez tout votre module pour qu'il travaille avec les ressources standard, comme mentionné dans la section (9).
2. Ajoutez une autre fonction de connexion, qui vérifie d'abord que la ressource existe dans la liste des ressources persistantes. Si c'est le cas, enregistrez cette ressource comme pour les ressources standard (et grâce à la première étape, cela va fonctionner immédiatement). Si la ressource n'existe pas, créez la, ajoutez la à la liste de ressources persistantes, et ajoutez la à la liste de ressources, ce qui fait que le code va fonctionner, et que le prochain appel renverra une ressource existante. Vous devez enregistrer ces fonctions avec un type différent (LE_MYSQL_LINK pour les liens non persistants, et LE_MYSQL_PLINK pour les liens persistants).

Si vous jetez un oeil dans mysql.c, vous verrez que, hormis la fonction de connexion complexe, rien n'a du être changé dans le module.

La même interface existe pour la liste des ressources standard, et pour la liste des ressources persistantes, seule la 'list' est remplacée par 'plist':

- php3_plist_insert(ptr, type) − retourne l'identifiant 'id' de la nouvelle ressource insérée.
- php3_plist_delete(id) − efface la ressource d'identifiant id
- php3_plist_find(id,*type) − retourne le pointeur de la ressource d'identifiant id, et modifie le type 'type'

Cependant, il est probable que ces fonctions seront inutiles pour vous, lorsque vous essayerez d'implémentez un module persistant. Typiquement, on utiliser le fait que la liste de ressources persistantes est une table de hash. Par exemple, dans les modules MySQL/mSQL, lors d'un appel à pconnect(), la fonction construit une chaîne avec l'hôte/utilisateur/mot_de_passe, et l'utilise pour enregistrer dans la table de hash. Au prochain appel, avec les mêmes hôte/utilisateur/mot_de_passe, la même clé sera générée, et la ressource associée sera retrouvée.

Jusqu'à ce que la documentation s'étoffe, jetez un oeil aux fichiers mysql.c ou msql.c pour voir comment implémentez vos accès aux ressources persistantes.

Une chose importante à noter : les ressources qui sont enregistrées dans la liste de ressource persistante ne DOIVENT PAS être allouée avec le gestionnaire de mémoire PHP, c'est−à−dire qu'elles ne doivent pas être créée avec emalloc(), estrdup(), etc. Au contraire, il faut utiliser les fonctions standard malloc(), strdup(), etc. La raison est for simple : à la fin de la requête, la mémoire sera supprimée par le gestionnaire. Etant donné que les liens persistants doivent être conservés, il ne faut pas utiliser le gestionnaire de mémoire.

Lorsque vous enregistrez une ressource qui sera placé dans la liste de ressources persistantes, il faut ajouter les destructeurs dans les deux listes de ressources, persistantes ou pas. Le destructeur de la liste de ressources non persistantes ne doit rien faire du tout, tandis que celui de la liste de ressources persistantes doit libérer proprement toutes les ressources acquises (mémoire, lien SQL...). Commep pour les ressources non persistantes vous DEVEZ ajouter un destructeur, même s'il ne fait rien. N'oubliez pas que emalloc() et compagnie ne doivent pas être utilisé en conjonction avec la liste de ressources persistantes, et donc, vous ne devez pas utiliser efree() non plus.

#### 7.5.1.11 Ajouter des directives de configuration à l'exécution

De nombreuses caractéristiques de PHP 3 peuvent être configurée à l'exécution. Ces directives peuvent apparaître dans le fichier php3.ini , ou, dans le cas du module Apache, dans le fichier .conf . L'avantage de l'avoir dans le fichier .conf , est que ces caractéristiques peuvent être configurées dossier par dossier. Cela signifie qu'un dossier peut avoir un safe mode exec dir, tandis qu'un autre en aura un autre. Cette granularité de la configuration peut être extrêmement pratique lorsque le serveur supporte plusieurs serveurs virtuels.

Les étapes de configuration d'une nouvelle directive sont :

1. Ajouter la directive à la structure php3_ini_structure dans le fichier mod_php3.h .
2. Dans main.c, éditez la fonction php3_module_startup et ajoutez l'appel approprié à cfg_get_string() ou cfg_get_long().
3. Ajoutez la directive, ses restrictions et un commentaire dans la structure php3_commands du fichier mod_php3.c. Notez la partie restrictions RSRC_CONF sont des directives qui ne peuvent être disponibles que dans le fichier de configuration Apache. Toutes les directives OR_OPTIONS peuvent être placées n'importe où, y compris dans un fichier .htaccess .
4. Soit dans php3take1handler(), soit dans php3flaghandler(), ajoutez l'entrée appropriée pour votre directive.

5. Dans la section de configuration, de _php3_info(), dans le fichier functions/info.c, vous devez ajouter votre configuration.
6. Finalement, vous devez utiliser votre configuration quelque part. Elle sera accessible par php3_ini.directive.

## 7.5.2 Appeler des fonctions utilisateurs

Pour appeler des fonctions utilisateurs depuis une fonction interne, vous devez utiliser la fonction call_user_function().

call_user_function() retourne SUCCESS en cas de succès, et FAILURE en cas d'échec, ou si la fonction n'a pas été trouvée. Vous devez vérifier cette valeur. Si la réponse est SUCCESS, vous êtes responsable de la destruction de retval (ou alors, retournez la comme valeur de réponse de votre fonction). Si la réponse est FAILURE, la valeur de retval est indéfinie, et vous ne devez pas y toucher.

Toutes les fonctions internes qui appellent une fonction utilisateur, **DOIVENT** être réentrante. En particulier, elles ne doivent pas utiliser de valeurs globales, ou de variables statiques.

call_user_function() prend 6 arguments :

### 7.5.2.1 HashTable *function_table

La table de hash dans laquelle le fonction doit être recherchée.

### 7.5.2.2 pval *object

Un pointeur sur un objet sur lequel la fonction est invoquée. Il devrait être à NULL , si on invoque une fonction globale. Si il n'est pas à NULL (ie, il pointe sur un objet), l'argument function_table est ignorée, et la liste des fonctions sera lue dans l'objet, plutôt que dans l'argument. L'objet PEUT être modifié par la fonction qui est appelée (la fonction y aura accès via $this). Si, vous quelque raison, vous ne le voulez pas, envoyez une copie de l'objet à la place.

### 7.5.2.3 pval *function_name

Le nom de la fonction à appeler. Elle doit être de type pval, IS_STRING, avec les valeurs de function_name.str.val et function_name.str.len correctes. function_name est modifié par call_user_function() − il est converti en minuscule. Si vous voulez préserver la casse, envoyez une copie du nom de la fonction.

### 7.5.2.4 pval *retval

Un pointeur sur une structure pval, dans laquelle la valeur de retour de la fonction sera placée. La structure doit avoir été allouée au préalable, − call_user_function() ne l'allouera pas.

#### 7.5.2.5 int param_count

Le nombre de paramètre passé à la fonction.

#### 7.5.2.6 pval *params[]

Un tableau de pointeur sur les valeurs qui vont être passées comme arguments à la fonction. Le premier argument est à l'offset 0, le second à l'offset 1,... Le tableau est un tableau de pointeurs sur pval; Les pointeurs sont envoyés tels quels à la fonction, ce qui signifie que si la fonction modifie les arguments, les valeurs originales seront modifiées. Si vous voulez l'éviter, passez une copie à la place.

### 7.5.3 Rapport d'erreurs

Pour signaler les erreurs d'une fonction interne, vous devez appelez la fonction php3_error(). Cette fonction prend deux arguments au moins : le niveau de l'erreur, et le message d'erreur, sous forme de chaîne de caractères. Tous les arguments suivants sont des paramètres de formats de chaîne. Les niveaux d'erreurs sont :

#### 7.5.3.1 E_NOTICE

Les notes ne sont pas affichées par défaut, et indique que le script a rencontré quelque chose qui peut être une erreur, mais peut aussi être un événement normal dans la vie du script. Par exemple, essayer d'accéder à une valeur qui n'a pas été déclarée, ou appeler stat sur un fichier qui n'existe pas.

#### 7.5.3.2 E_WARNING

Les alertes sont affichées par défaut, mais n'interrompent pas l'exécution du script. Elles indiquent un problème qui doit être intercepté par le script avant que l'appel. Par exemple, appeler ereg avec une regex invalide.

#### 7.5.3.3 E_ERROR

Les erreurs sont aussi affichées par défaut, et l'exécution du script est interrompue. Elles indiquent des erreurs qui ne peuvent pas être ignorées, comme des problèmes d'allocation de mémoire, par exemple.

#### 7.5.3.4 E_PARSE

Les erreurs d'analyse de doivent être générées que par l'analyseur. Elles ne sont citées ici que dans le but d'être exhaustif.

#### 7.5.3.5 E_CORE_ERROR

Elles sont similaires aux erreurs E_ERROR , mais elles sont générées par le code de PHP. Les fonctions ne doivent pas générer ce genre d'erreur.

#### 7.5.3.6 E_CORE_WARNING

Elles sont similaires à E_WARNING , mais elles sont générées par le code de PHP. Les fonctions ne doivent pas générer ce genre d'erreur.

#### 7.5.3.7 E_COMPILE_ERROR

Elles sont similaires à E_ERROR , mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.

#### 7.5.3.8 E_COMPILE_WARNING

Elles sont similaires à E_WARNING , mais elles sont générées par Zend Scripting Engine. Les fonctions ne doivent pas générer ce genre d'erreur.

#### 7.5.3.9 E_USER_ERROR

E_USER_ERROR est comparable à E_ERROR . Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur.

#### 7.5.3.10 E_USER_WARNING

E_USER_WARNING est comparable à E_WARNING . Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur.

#### 7.5.3.11 E_USER_NOTICE

E_USER_WARNING est comparable à E_NOTICE . Elle est générée en PHP par l'utilisation de la fonction trigger_error . Les fonctions ne doivent pas générer ce genre d'erreur.

#### 7.5.3.12 E_ALL

Toutes les erreurs ci−dessus. Utiliser ce niveau de rapport d'erreur provoque l'affichage de toutes les erreurs.

## 7.6 Utiliser PHP en ligne de commande

Depuis la version 4.3, PHP supporte un nouveau type de SAPI (Server Application Programming Interface, c'est à dire Interface de Programmation d'Applications Serveur) appelé CLI , ce qui signifie **Command Line Interface** et se traduit par Interface de Ligne de Commande . Comme son nom l'indique, ce type SAPI cible les applications shell (ou desktop), écrites en PHP. Il y a pas mal de différences enter le type CLI SAPI et les autres SAPI , qui sont expliqués dans ce chapitre.

Le CLI SAPI a été publié pour la première fois avec la version PHP 4.2.0 , mais il est resté expérimental, et doit être explicitement activé avec la commande −−enable−cli , lorsque vous exécutez le script ./configure . Depuis PHP 4.3.0, le CLI SAPI n'est plus expérimental, et est donc **toujours** compilé et installé comme binaire PHP (appelé aussi php.exe sous Windows).

Les différences les plus notables entre le CLI SAPI et les SAPI sont :

- Contrairement au CGI SAPI , aucun en−tête HTTP n'est écrit dans le résultat.

  Bien que le CGI SAPI fournisse un moyen de supprimer les en−têtes HTTP, il n'y a pas moyen d'activer les en−têtes HTTP dans le CLI SAPI .
- Il y a plusieurs directives du php.ini qui sont ignorées par le CLI SAPI , car elles n'ont pas de sens, en environnement shell :

| Directive | Valeur par défaut pour CLI SAPI | Commentaire |
|---|---|---|
| html_errors | FALSE | Il peut être bien difficile de lire les messages d'erreur sur un terminal lorsqu'ils sont noyés dans des balises HTML sans grand intérêts. Par conséquent, cette directive est forcée à FALSE . |
| implicit_flush | TRUE | Il est souhaitable que tout affichage en provenance de print , echo et consorts soit immédiatement affiché dans le terminal, et non pas placé dans un buffer quelconque. Vous pouvez toujours utiliser la bufferisation de sortie si vous voulez retarder un affichage, ou bien en manipuler ultimement le contenu. |
| max_execution_time | 0 (unlimited) | Etant donné les possibilités infinies de PHP en environnement shell, le temps d'exécution maximal d'un script PHP a été rendu illimité. Alors que les scripts destinés au web doivent s'accomplir en une fraction de seconde, il arrive que les scripts shell requièrent bien plus de temps. |
| register_argc_argv | TRUE | En donnant la valeur de TRUE à cette directive, vous aurez toujours accès à la variable **argc** (représentant le nombre d'arguments passés à l'application) et **argv** (le tableau contenant les arguments passés) dans le CLI SAPI .<br><br>Depuis PHP 4.3.0, les variables $argc et $argv sont définies et remplies avec les valeurs appropriées, en utilisant CLI SAPI . Avant cette version, la création de ces variables était liée au comportement des versions CGI et MODULE , qui requièrent l'activation de la directive PHP register_globals . Indépendamment de la version ou de la valeur de register_globals, vous pouvez toujour accéder à $_SERVER et $HTTP_SERVER_VARS . Par exemple: $_SERVER['argv'] |
| Note | | |
| Ces directives ne peuvent pas être initialisées avec d'autres valeurs dans le fichier php.ini ou par une autre méthode. C'est une limitation, car ces valeurs par défaut s'appliquent une fois que tous les autres fichiers de configuration ont été analysés. Cependant, ces valeurs peuvent être modifiées durant l'exécution (ce qui n'est pas logique pour certaines directives, comme par exemple register_argc_argv ). | | |

- Pour faciliter le travail en environnement shell, les constantes suivantes sont définies :

| Constante | Déscription |
|---|---|
| STDIN | Un descripteur de fichier déjà disponible vers stdin . Cela évite de l'ouvrir avec<br><br>`$stdin = fopen('php://stdin', 'r');` |
| STDOUT | Un descripteur de fichier déjà disponible vers stdout . Cela évite de l'ouvrir avec<br><br>`$stdout = fopen('php://stdout', 'w');` |
| STDERR | Un descripteur de fichier déjà disponible vers stderr . Cela évite de l'ouvrir avec<br><br>`$stderr = fopen('php://stderr', 'w');` |

  Etant donné ce qui précède, vous n'avez pas besoin d'ouvrir un fichier vers stderr par vous−même, mais vous pouvez utiliser cette constante directement, comme un descripteur de fichier :

```
php -r 'fwrite(STDERR, "stderr\n");'
```

  Vous n'avez pas non plus à fermer explicitement ces fichiers, PHP s'en chargera automatiquement.
- CLI SAPI **ne transforme pas** le dossier courant en dossier d'exécution du script!

Exemple de différence entre CGI SAPI et CLI SAPI :

```
<?php
   /* Un test simple : affiche le dossier d'exécution */
   echo getcwd(), "\n";
?>
```

Lorsque vous utilisez la version CGI , l'affichage pourra être
$ pwd
/tmp

$ php−cgi −f autre_dossier/test.php
/tmp/autre_dossier
Cela monter clairement que PHP modifie le dossier courant, et utilise le dossier du script exécuté.

En utilisant le CLI SAPI , on obtient :
$ pwd
/tmp

$ php −f autre_dossier/test.php
/tmp
Cela donne beaucoup plus de souplesse lorsque vous rédigez des scripts shell avec PHP .

| Note |
| --- |
| CGI SAPI supporte le même comportement de CLI SAPI , en lui passant l'option −C , lorsque vous l'invoquez en ligne de commmande. |

La liste des options de ligne de commande fournies par PHP est disponible en exécutant PHP avec l'option −h :
Usage: php [options] [−f] <file> [args...]
php [options] −r <code> [args...]
php [options] [−− args...]
−s Display colour syntax highlighted source.
−w Display source with stripped comments and whitespace.
−f <file> Parse <file>.
−v Version number
−c <path>|<file> Look for php.ini file in this directory
−a Run interactively
−d foo[=bar] Define INI entry foo with value 'bar'
−e Generate extended information for debugger/profiler
−z <file> Load Zend extension <file>.
−l Syntax check only (lint)
−m Show compiled in modules
−i PHP information
−r <code> Run PHP <code> without using script tags <?..?>
−h This help

args... Arguments passed to script. Use −− args when first argument
starts with − or script is read from stdin

Le CLI SAPI dispose de trois moyens pour lire le code du script PHP que vous voulez exécuter :

1. Indiquer à PHP d'exécuter un fichier :

php mon_script.php

php −f mon_script.php
Les deux méthodes (en utilisant −f ou pas) exécute le script contenu dans le fichier mon_script.php . Vous pouvez choisir n'importe quel fichier, et ces fichiers ne sont pas tenus d'utiliser l'extension .php . N'importe quelle extension peut faire l'affaire.

2. Donner du code PHP à exécuter directement en ligne de commande.

   php −r 'print_r(get_defined_constants());'
   Une attention particulière doit alors être apportée aux variables d'environnement, qui seront remplacées, et aux guillements, qui ont des significations spéciales en ligne de commande.

   | Note |
   | --- |
   | Lisez l'exemple attentivement, il n'y a ni balise d'ouverture, ni balise de fermeture! L'option −r fonctionne sans ces balises, et les ajouter conduira alors à une erreur d'analyse. |

3. Donner du code PHP dans l'entrée standard ( stdin ).

   Cela donne la possibilité de créer dynamiquement du code PHP , puis de le fournir à PHP, et enfin, de le traiter à nouveau en shell. Voici un exemple fictif :
   $ some_application | some_filter | php | sort −u >final_output.txt

Il n'est pas possible de combiner ces trois modes d'exécution.

Comme toute application shell, l'exécutable PHP accepte des arguments, et votre script peut aussi les recevoirs. Le nombre d'arguments n'est pas limité par PHP, mais par le shell, qui a une limite en terme de nombre de caractères qui peuvent être passés. Généralement, vous n'atteidrez pas cette limite, mais cela reste possible. Les arguments passés au script seront transmis via la variable tableau $argv . L,index zéro contiendra toujours le nom du script appelé. Si le code a été fourni en ligne de commande, ou via stdin , ce premier élément vaudra − . L'autre variable globale fournie est $argc qui contient le nombre d'éléments dans le tableau $argv : ce nombre est **différent** du nombre d'arguments passés au script.

Tant que les arguments que vous passez à votre script ne commencent pas par le caractère − , il n'y a rien de spécial à surveiller. Si vous passez des arguments à voter script qui commencent par − , cela posera des problèmes car PHP va penser qu'il doit les interpréter. Pour éviter cela, utilisez le séparateur −− . Après cet argument, tous les arguments suivants seront passés à votre script sans être modifié ou analysé par PHP.

```
# Cela ne va pas exécuter le code, mais afficher l'aide de PHP
$ php −r 'var_dump($argv);' −h
Usage: php [options] [−f] <file> [args...]
[...]

# Cela va passer l'argument '−h' à votre script, et éviter que PHP ne le traite
$ php −r 'var_dump($argv);' −− −h
array(2) {
[0]=>
string(1) "−"
[1]=>
string(2) "−h"
}
```

Cependant, il y a une autre méthode pour utiliser PHP en script shell. Vous pouvez aussi utiliser la ligne #!/usr/bin/php en tout début de votre script, suivi de code PHP compris entre balise ouvrantes/fermantes. Vous pouvez aussi ajouter après #!/usr/bin/php les options nécessaires. Vous pouvez aussi le faire en ligne de commande. De cette manière, le script peut être exécuté comme un script shell habituel :

```
 #!/usr/bin/php
<?php
    var_dump($argv);
?>
```

En supposant que ce fichier s'appelle test , dans le dossier courant, nous pouvons alors faire ceci :
$ chmod 755 test
$ ./test −h −− foo
array(4) {
[0]=>
string(6) "./test"
[1]=>
string(2) "−h"
[2]=>
string(2) "−−"
[3]=>
string(3) "foo"
}
Comme vous le voyez, aucune précaution n'est nécessaire pour passer des paramètres qui commencent par − à votre script.

| Option | Description |
|---|---|
| −s | Affiche le code avec la colorisation syntaxique.<br><br>Cette option utilise le mécanisme interne pour analyser le fichier, et produire une version colorisée du code source, au format HTML. Notez que cette option ne fait que générer un bloc HTML , sans les en−têtes. <code> [...] </code><br><br>Note<br>Cette option ne fonctionne pas avec l'option −r . |
| −w | Affiche la source sans les commentaires et les espaces.<br><br>Note<br>Cette option ne fonctionne pas avec l'option −r . |
| −f | Analyse et exécute le fichier donné après l'option −f . Cette option est optionnelle, et peut être omise. Le seul nom du fichier est suffisant. |
| −v | Affiche les versions de PHP, PHP SAPI, et Zend sur le terminal. Par exemple :<br>$ php −v<br>PHP 4.3.0−dev (cli), Copyright (c) 1997−2002 The PHP Group<br>Zend Engine v1.2.1, Copyright (c) 1998−2002 Zend Technologies |
| −c | Avec cette option, vous pouvez spécifier le nom du dossier dans lequel PHP va recherche le fichier php.ini , ou encore vous pouvez spécifier un fichier de configuration directement (qui n'a pas obligatoirement à être appelé php.ini ) :<br>$ php −c /custom/directory/ mon_script.php<br><br>$ php −c /custom/directory/custom−file.ini mon_script.php |
| −a | Exécute PHP de manière interactive. |
| −d | Cette option permet de modifier n'importe quelle directive de configuration du fichier php.ini . La syntaxe est : |

| | |
|---|---|
| | −d configuration_directive[=value]<br><br>Exemples:<br># L'omission de la valeur conduit à donner la valeur de "1"<br>$ php −d max_execution_time −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(1) "1"<br><br># Passer une valeur vide conduit à donner la valeur de ""<br>php −d max_execution_time= −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(0) ""<br><br># La directive de configuration sera n'importe quelle valeur passée après le caractère '='<br>$ php −d max_execution_time=20 −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(2) "20"<br>$ php −d max_execution_time=doesntmakesense −r '$foo = ini_get("max_execution_time"); var_dump($foo);'<br>string(15) "doesntmakesense" |
| −e | Génére des informations étendues pour le profilage et le débogage. |
| −z | Charge une extension Zend. Si seulement un fichier est fourni, PHP essaie de charger cette extension dans le dossier courant par défaut des librairie sur votre système (généralement spécifié avec /etc/ld.so.conf sous Linux). Passer un nom de fichier avec le chemin complet fera que PHP utilisera ce fichier, sans recherche dans les dossiers classiques. Un chemin de dossier relatif indiquera à PHP qu'il doit chercher uniquement les extensions dans ce dossier. |
| −l | Cette option permet de faire une vérification syntaxique sur le code PHP fourni. En cas de réussite, le message No syntax errors detected in <filename> ( Aucune d'erreur de syntaxe n'ont été détectées dans <nom_de_fichier> est affiché sur la sortie standard, et le script shell retourne 0 . En cas d'erreur, le message Errors parsing <filename> ( Erreurs d'analyse dans le fichier <filename> ) est affiché, en plus des messages d'erreurs détectés par l'analyseur lui même. Le script shell retourne le code 255 .<br><br>Cette option ne détecte pas les erreurs fatales (comme par exemple les fonctions non définies). Utilisez −f si vous voulez tester aussi ces erreurs.<br><br>Note<br>Cette option ne fonctionne pas avec l'option −r . |
| −m | Cette option liste les extensions PHP et Zend compilées et chargées :<br>$ php −m<br>[PHP Modules]<br>xml<br>tokenizer<br>standard<br>session<br>posix<br>pcre<br>overload<br>mysql<br>mbstring<br>ctype<br><br>[Zend Modules] |
| −i | Cette option appelle la fonction phpinfo , et affiche le résultat. Si PHP ne fonctionne pas correctement, il est recommandé d'utiliser la commande php −i et de voir si il n'y a pas d'erreurs affichées avant ou après la table d'information. N'oubliez pas que le résutltat de cette option est au format HTML , et donc grand et illisible directement. |
| −r | Cette option permet l'exécution de PHP directement dans la ligne de commande. Les balises |

| | |
|---|---|
| | de PHP ( <?php et ?> ) **ne sont pas** demandées, et coauseront une erreur d'analyse.<br><br>Note<br>De l'attention doit être portée lors de l'utilisation de cette option de PHP , pour qu'il n'y ait pas de collision avec les substitutions de variables en lignes de commande, réalisées par le shell.<br><br>Exemple conduisant à une erreur d'analyse :<br>$ php −r "$foo = get_defined_constants();"<br>Command line code(1) : Parse error − parse error, unexpected '='<br>Le problème ici est que le shell (sh/bash) effectue une substitution de variables, grâce aux guillemets doubles " . puisque la variable $foo n'est probablement pas définie dans le shell, elle est remplacée par rien, ce qui fait que le code passé à PHP pour l'exécution est :<br>$ php −r " = get_defined_constants();"<br>La solution de ce problème est d'utiliser les guillemets simples ' . Les variables de ces chaînes ne seront pas substituées par leurs valeurs par le shell.<br>$ php −r '$foo = get_defined_constants(); var_dump($foo);'<br>array(370) {<br>["E_ERROR"]=><br>int(1)<br>["E_WARNING"]=><br>int(2)<br>["E_PARSE"]=><br>int(4)<br>["E_NOTICE"]=><br>int(8)<br>["E_CORE_ERROR"]=><br>[...]<br>Si vous utilisez un shell différent de sh/bash, vous pouvez rencontrer d'autres problèmes. N'hésitez pas à ouvrir un rapport de bugs, ou d'envoyer un mail à phpdoc@lists.php.net. Il est toujours très facile d'avoir des problèmes lorsque vous essayez d'inclure des variables shell dans le code, ou d'utiliser les anti−slash pour l'échappement. Vous aurez été prévenu. |
| −h | Avec cette option, vous pouvez obtenir des informations sur la liste des options supporté par votre PHP, avec des explications succintes. |

L'exécutable PHP peut être utilisé pour exécuter des scripts indépendants du serveur web. Si vous êtes sur un système Unix, il est recommandé d'ajouter la ligne de configuration en début de scripts, de le rendre exécutable de manière à ce que le système sache quel programme doit exécuter le script. Sous Windows, vous pouvez associer l'exécutable php.exe avec le double−clic sur les fichiers d'extension .php , ou bien vous pouvez faire un fichier batch pour exécuter le script grâce à PHP. La première ligne utilisée dans le monde Unix ne perturbera pas PHP sous Windows, ce qui rend les scripts facilements portables. Un exemple complet est disponible ci−dessous :

Script prévu pour être exécuté en ligne de commande (script.php)

```
 #!/usr/bin/php
<?php

if ($argc != 2 || in_array($argv[1], array('−−help', '−help', '−h', '−?'))) {
?>

C'est une ligne de commande à une option.

  Utilisation :
  <?php echo $argv[0]; ?> <option>

  <option> peut être un mot que vous souhaitez afficher.
  Avec les options −−help, −help, −h,
  et −?, vous obtiendrez cette aide.

<?php
} else {
    echo $argv[1];
}
?>
```

Dans le script ci−dessus, nous utilisons la première ligne pour indiquer que le fichier doit être exécuté par PHP. Nous travaillons avec une version CLI, donc il n'y a pas d'en−têtes HTTP qui soient affichés. Il y a deux variables que vous pouvez utiliser avec les applications de ligne de commande : $argc et $argv . La première est le nombre d'arguments plus un (le nom du script qui est exécuté). La seconde est un tableau contenant les arguments, commençant avec le nom du script en élément 0 ( $argv[0] ).

Dans notre exemple, nous avons vérifié qu'il y a plus ou moins d'un argument. De plus, si cet argument est −−help , −help , −h ou −? , nous avons affiché un message d'aide, ainsi que le nom du script. Nous nous recevons un autre argument, cet argument est affiché dans le terminal.

Pour exécuter le script ci−dessus sous Unix, vous devez le rendre exécuétable, puis l'appeler avec une commande comme : script.php echothis ou script.php −h . Sous Windows, vous pouvez faire un fichier batch pour cela :

| Fichier batch pour exécuter un script PHP en ligne de commande (script.bat) |
|---|
| @c:\php\php.exe script.php %1 %2 %3 %4 |

Si vous avez nommé le programme ci−dessus script.php , et que vous avez votre exécutable php.exe situé à c:\php\php.exe , ce fichier batch l'exécutera avec les options que vous lui passez : script.bat echothis ou script.bat −h .

Voir aussi l'extension Readline , qui dispose de nombreuses fonctions pour améliorer la convivialité de vos applications en ligne de commande.

## 7.7 Liste d'alias

Voici la liste des aliases. Tous les alias sont listés ci−dessous. C'est une très mauvaise habitude d'utiliser ces alias, car ils risquent à tous moment de disparaître, rendus obsolète sans préavis, ou bien par un simple changement de nom, ce qui rend votre script inutilisable avec des versions plus récentes de PHP. Préférez toujours les versions officielles. En fait, cette liste est surtout destinées à ceux qui doivent mettre à jour leur scripts avec les syntaxes plus récentes.

Cette liste est en phase avec PHP 4.0.6. Pour une liste d'alias qui soit mise à jour quotidiennement, voyez le site de Zend .

| Alias | Fonction principale | Extension mère |
|---|---|---|
| _ | gettext | Gettext |
| add | swfmovie_add | Ming (flash) |
| add | swfsprite_add | Ming (flash) |
| add_root | domxml_add_root | DOM XML |
| addaction | swfbutton_addaction | Ming (flash) |
| addcolor | swfdisplayitem_addcolor | Ming (flash) |
| addentry | swfgradient_addentry | Ming (flash) |
| addfill | swfshape_addfill | Ming (flash) |
| addshape | swfbutton_addshape | Ming (flash) |
| addstring | swftext_addstring | Ming (flash) |
| addstring | swftextfield_addstring | Ming (flash) |
| align | swftextfield_align | Ming (flash) |
| attributes | domxml_attributes | DOM XML |
| children | domxml_children | DOM XML |
| chop | rtrim | Base syntax |
| close | closedir | Base syntax |
| com_get | com_propget | COM |
| com_propset | com_propput | COM |
| com_set | com_propput | COM |
| cv_add | ccvs_add | CCVS |
| cv_auth | ccvs_auth | CCVS |

| | | |
|---|---|---|
| cv_command | ccvs_command | CCVS |
| cv_count | ccvs_count | CCVS |
| cv_delete | ccvs_delete | CCVS |
| cv_done | ccvs_done | CCVS |
| cv_init | ccvs_init | CCVS |
| cv_lookup | ccvs_lookup | CCVS |
| cv_new | ccvs_new | CCVS |
| cv_report | ccvs_report | CCVS |
| cv_return | ccvs_return | CCVS |
| cv_reverse | ccvs_reverse | CCVS |
| cv_sale | ccvs_sale | CCVS |
| cv_status | ccvs_status | CCVS |
| cv_textvalue | ccvs_textvalue | CCVS |
| cv_void | ccvs_void | CCVS |
| die | exit | Miscellaneous functions |
| dir | getdir | Base syntax |
| diskfreespace | disk_free_space | Filesystem |
| domxml_getattr | domxml_get_attribute | DOM XML |
| domxml_setattr | domxml_set_attribute | DOM XML |
| doubleval | floatval | Base syntax |
| drawarc | swfshape_drawarc | Ming (flash) |
| drawcircle | swfshape_drawcircle | Ming (flash) |
| drawcubic | swfshape_drawcubic | Ming (flash) |
| drawcubicto | swfshape_drawcubicto | Ming (flash) |
| drawcurve | swfshape_drawcurve | Ming (flash) |
| drawcurveto | swfshape_drawcurveto | Ming (flash) |
| drawglyph | swfshape_drawglyph | Ming (flash) |
| drawline | swfshape_drawline | Ming (flash) |
| drawlineto | swfshape_drawlineto | Ming (flash) |
| dtd | domxml_intdtd | DOM XML |
| dumpmem | domxml_dumpmem | DOM XML |
| fbsql | fbsql_db_query | FrontBase |
| fputs | fwrite | Base syntax |
| get_attribute | domxml_get_attribute | DOM XML |
| getascent | swffont_getascent | Ming (flash) |
| getascent | swftext_getascent | Ming (flash) |
| getattr | domxml_get_attribute | DOM XML |
| getdescent | swffont_getdescent | Ming (flash) |
| getdescent | swftext_getdescent | Ming (flash) |
| getheight | swfbitmap_getheight | Ming (flash) |
| getleading | swffont_getleading | Ming (flash) |
| getleading | swftext_getleading | Ming (flash) |
| getshape1 | swfmorph_getshape1 | Ming (flash) |
| getshape2 | swfmorph_getshape2 | Ming (flash) |
| getwidth | swfbitmap_getwidth | Ming (flash) |
| getwidth | swffont_getwidth | Ming (flash) |
| getwidth | swftext_getwidth | Ming (flash) |
| gzputs | gzwrite | Zlib |
| i18n_convert | mb_convert_encoding | Multi-bytes Strings |
| i18n_discover_encoding | mb_detect_encoding | Multi-bytes Strings |
| i18n_http_input | mb_http_input | Multi-bytes Strings |
| i18n_http_output | mb_http_output | Multi-bytes Strings |
| i18n_internal_encoding | mb_internal_encoding | Multi-bytes Strings |
| i18n_ja_jp_hantozen | mb_convert_kana | Multi-bytes Strings |
| i18n_mime_header_decode | mb_decode_mimeheader | Multi-bytes Strings |
| i18n_mime_header_encode | mb_encode_mimeheader | Multi-bytes Strings |
| imap_create | imap_createmailbox | IMAP |
| imap_fetchtext | imap_body | IMAP |
| imap_getmailboxes | imap_list_full | IMAP |
| imap_getsubscribed | imap_lsub_full | IMAP |
| imap_header | imap_headerinfo | IMAP |
| imap_listmailbox | imap_list | IMAP |
| imap_listsubscribed | imap_lsub | IMAP |
| imap_rename | imap_renamemailbox | IMAP |
| imap_scan | imap_listscan | IMAP |
| imap_scanmailbox | imap_listscan | IMAP |
| ini_alter | ini_set | Base syntax |
| is_double | is_float | Base syntax |
| is_integer | is_int | Base syntax |
| is_long | is_int | Base syntax |
| is_real | is_float | Base syntax |

| | | |
|---|---|---|
| is_writeable | is_writable | Base syntax |
| join | implode | Base syntax |
| labelframe | swfmovie_labelframe | Ming (flash) |
| labelframe | swfsprite_labelframe | Ming (flash) |
| last_child | domxml_last_child | DOM XML |
| lastchild | domxml_last_child | DOM XML |
| ldap_close | ldap_unbind | LDAP |
| magic_quotes_runtime | set_magic_quotes_runtime | Base syntax |
| mbstrcut | mb_strcut | Multi-bytes Strings |
| mbstrlen | mb_strlen | Multi-bytes Strings |
| mbstrpos | mb_strpos | Multi-bytes Strings |
| mbstrrpos | mb_strrpos | Multi-bytes Strings |
| mbsubstr | mb_substr | Multi-bytes Strings |
| ming_setcubicthreshold | ming_setcubicthreshold | Ming (flash) |
| ming_setscale | ming_setscale | Ming (flash) |
| move | swfdisplayitem_move | Ming (flash) |
| movepen | swfshape_movepen | Ming (flash) |
| movepento | swfshape_movepento | Ming (flash) |
| moveto | swfdisplayitem_moveto | Ming (flash) |
| moveto | swffill_moveto | Ming (flash) |
| moveto | swftext_moveto | Ming (flash) |
| msql | msql_db_query | mSQL |
| msql_createdb | msql_create_db | mSQL |
| msql_dbname | msql_result | mSQL |
| msql_dropdb | msql_drop_db | mSQL |
| msql_fieldflags | msql_field_flags | mSQL |
| msql_fieldlen | msql_field_len | mSQL |
| msql_fieldname | msql_field_name | mSQL |
| msql_fieldtable | msql_field_table | mSQL |
| msql_fieldtype | msql_field_type | mSQL |
| msql_freeresult | msql_free_result | mSQL |
| msql_listdbs | msql_list_dbs | mSQL |
| msql_listfields | msql_list_fields | mSQL |
| msql_listtables | msql_list_tables | mSQL |
| msql_numfields | msql_num_fields | mSQL |
| msql_numrows | msql_num_rows | mSQL |
| msql_regcase | sql_regcase | mSQL |
| msql_selectdb | msql_select_db | mSQL |
| msql_tablename | msql_result | mSQL |
| mssql_affected_rows | sybase_affected_rows | Sybase |
| mssql_affected_rows | sybase_affected_rows | Sybase |
| mssql_close | sybase_close | Sybase |
| mssql_close | sybase_close | Sybase |
| mssql_connect | sybase_connect | Sybase |
| mssql_connect | sybase_connect | Sybase |
| mssql_data_seek | sybase_data_seek | Sybase |
| mssql_data_seek | sybase_data_seek | Sybase |
| mssql_fetch_array | sybase_fetch_array | Sybase |
| mssql_fetch_array | sybase_fetch_array | Sybase |
| mssql_fetch_field | sybase_fetch_field | Sybase |
| mssql_fetch_field | sybase_fetch_field | Sybase |
| mssql_fetch_object | sybase_fetch_object | Sybase |
| mssql_fetch_object | sybase_fetch_object | Sybase |
| mssql_fetch_row | sybase_fetch_row | Sybase |
| mssql_fetch_row | sybase_fetch_row | Sybase |
| mssql_field_seek | sybase_field_seek | Sybase |
| mssql_field_seek | sybase_field_seek | Sybase |
| mssql_free_result | sybase_free_result | Sybase |
| mssql_free_result | sybase_free_result | Sybase |
| mssql_get_last_message | sybase_get_last_message | Sybase |
| mssql_get_last_message | sybase_get_last_message | Sybase |
| mssql_min_client_severity | sybase_min_client_severity | Sybase |
| mssql_min_error_severity | sybase_min_error_severity | Sybase |
| mssql_min_message_severity | sybase_min_message_severity | Sybase |
| mssql_min_server_severity | sybase_min_server_severity | Sybase |
| mssql_num_fields | sybase_num_fields | Sybase |
| mssql_num_fields | sybase_num_fields | Sybase |
| mssql_num_rows | sybase_num_rows | Sybase |
| mssql_num_rows | sybase_num_rows | Sybase |
| mssql_pconnect | sybase_pconnect | Sybase |
| mssql_pconnect | sybase_pconnect | Sybase |

| | | |
|---|---|---|
| mssql_query | sybase_query | Sybase |
| mssql_query | sybase_query | Sybase |
| mssql_result | sybase_result | Sybase |
| mssql_result | sybase_result | Sybase |
| mssql_select_db | sybase_select_db | Sybase |
| mssql_select_db | sybase_select_db | Sybase |
| multcolor | swfdisplayitem_multcolor | Ming (flash) |
| mysql | mysql_db_query | MySQL |
| mysql_createdb | mysql_create_db | MySQL |
| mysql_db_name | mysql_result | MySQL |
| mysql_dbname | mysql_result | MySQL |
| mysql_dropdb | mysql_drop_db | MySQL |
| mysql_fieldflags | mysql_field_flags | MySQL |
| mysql_fieldlen | mysql_field_len | MySQL |
| mysql_fieldname | mysql_field_name | MySQL |
| mysql_fieldtable | mysql_field_table | MySQL |
| mysql_fieldtype | mysql_field_type | MySQL |
| mysql_freeresult | mysql_free_result | MySQL |
| mysql_listdbs | mysql_list_dbs | MySQL |
| mysql_listfields | mysql_list_fields | MySQL |
| mysql_listtables | mysql_list_tables | MySQL |
| mysql_numfields | mysql_num_fields | MySQL |
| mysql_numrows | mysql_num_rows | MySQL |
| mysql_selectdb | mysql_select_db | MySQL |
| mysql_tablename | mysql_result | MySQL |
| name | domxml_attrname | DOM XML |
| new_child | domxml_new_child | DOM XML |
| new_xmldoc | domxml_new_xmldoc | DOM XML |
| nextframe | swfmovie_nextframe | Ming (flash) |
| nextframe | swfsprite_nextframe | Ming (flash) |
| node | domxml_node | DOM XML |
| oci8append | ocicollappend | OCI8 |
| oci8assign | ocicollassign | OCI8 |
| oci8assignelem | ocicollassignelem | OCI8 |
| oci8close | ocicloselob | OCI8 |
| oci8free | ocifreecoll | OCI8 |
| oci8free | ocifreedesc | OCI8 |
| oci8getelem | ocicollgetelem | OCI8 |
| oci8load | ociloadlob | OCI8 |
| oci8max | ocicollmax | OCI8 |
| oci8ocifreecursor | ocifreestatement | OCI8 |
| oci8save | ocisavelob | OCI8 |
| oci8savefile | ocisavelobfile | OCI8 |
| oci8size | ocicollsize | OCI8 |
| oci8trim | ocicolltrim | OCI8 |
| oci8writetemporary | ociwritetemporarylob | OCI8 |
| oci8writetofile | ociwritelobtofile | OCI8 |
| odbc_do | odbc_exec | OCI8 |
| odbc_field_precision | odbc_field_len | OCI8 |
| output | swfmovie_output | Ming (flash) |
| parent | domxml_parent | DOM XML |
| pdf_add_outline | pdf_add_bookmark | PDF |
| pg_clientencoding | pg_client_encoding | PostgreSQL |
| pg_setclientencoding | pg_set_client_encoding | PostgreSQL |
| pos | current | Base syntax |
| recode | recode_string | Recode |
| remove | swfmovie_remove | Ming (flash) |
| remove | swfsprite_remove | Ming (flash) |
| rewind | rewinddir | Base syntax |
| root | domxml_root | DOM XML |
| rotate | swfdisplayitem_rotate | Ming (flash) |
| rotateto | swfdisplayitem_rotateto | Ming (flash) |
| rotateto | swffill_rotateto | Ming (flash) |
| save | swfmovie_save | Ming (flash) |
| savetofile | swfmovie_savetofile | Ming (flash) |
| scale | swfdisplayitem_scale | Ming (flash) |
| scaleto | swfdisplayitem_scaleto | Ming (flash) |
| scaleto | swffill_scaleto | Ming (flash) |
| set_attribute | domxml_set_attribute | DOM XML |
| set_content | domxml_set_content | DOM XML |
| setaction | swfbutton_setaction | Ming (flash) |

| | | |
|---|---|---|
| setattr | domxml_set_attribute | DOM XML |
| setbackground | swfmovie_setbackground | Ming (flash) |
| setbounds | swftextfield_setbounds | Ming (flash) |
| setcolor | swftext_setcolor | Ming (flash) |
| setcolor | swftextfield_setcolor | Ming (flash) |
| setdepth | swfdisplayitem_setdepth | Ming (flash) |
| setdimension | swfmovie_setdimension | Ming (flash) |
| setdown | swfbutton_setdown | Ming (flash) |
| setfont | swftext_setfont | Ming (flash) |
| setfont | swftextfield_setfont | Ming (flash) |
| setframes | swfmovie_setframes | Ming (flash) |
| setframes | swfsprite_setframes | Ming (flash) |
| setheight | swftext_setheight | Ming (flash) |
| setheight | swftextfield_setheight | Ming (flash) |
| sethit | swfbutton_sethit | Ming (flash) |
| setindentation | swftextfield_setindentation | Ming (flash) |
| setleftfill | swfshape_setleftfill | Ming (flash) |
| setleftmargin | swftextfield_setleftmargin | Ming (flash) |
| setline | swfshape_setline | Ming (flash) |
| setlinespacing | swftextfield_setlinespacing | Ming (flash) |
| setmargins | swftextfield_setmargins | Ming (flash) |
| setmatrix | swfdisplayitem_setmatrix | Ming (flash) |
| setname | swfdisplayitem_setname | Ming (flash) |
| setname | swftextfield_setname | Ming (flash) |
| setover | swfbutton_setover | Ming (flash) |
| setrate | swfmovie_setrate | Ming (flash) |
| setratio | swfdisplayitem_setratio | Ming (flash) |
| setrightfill | swfshape_setrightfill | Ming (flash) |
| setrightmargin | swftextfield_setrightmargin | Ming (flash) |
| setspacing | swftext_setspacing | Ming (flash) |
| setup | swfbutton_setup | Ming (flash) |
| show_source | highlight_file | Base syntax |
| sizeof | count | Base syntax |
| skewx | swfdisplayitem_skewx | Ming (flash) |
| skewxto | swfdisplayitem_skewxto | Ming (flash) |
| skewxto | swffill_skewxto | Ming (flash) |
| skewy | swfdisplayitem_skewy | Ming (flash) |
| skewyto | swfdisplayitem_skewyto | Ming (flash) |
| skewyto | swffill_skewyto | Ming (flash) |
| snmpwalkoid | snmprealwalk | SNMP |
| strchr | strstr | Base syntax |
| streammp3 | swfmovie_streammp3 | Ming (flash) |
| swfaction | swfaction_init | Ming (flash) |
| swfbitmap | swfbitmap_init | Ming (flash) |
| swfbutton | swfbutton_init | Ming (flash) |
| swffill | swffill_init | Ming (flash) |
| swffont | swffont_init | Ming (flash) |
| swfgradient | swfgradient_init | Ming (flash) |
| swfmorph | swfmorph_init | Ming (flash) |
| swfmovie | swfmovie_init | Ming (flash) |
| swfshape | swfshape_init | Ming (flash) |
| swfsprite | swfsprite_init | Ming (flash) |
| swftext | swftext_init | Ming (flash) |
| swftextfield | swftextfield_init | Ming (flash) |
| unlink | domxml_unlink_node | DOM XML |
| xptr_new_context | xpath_new_context | DOM XML |

# 7.8 Mots réservés en PHP

Cette annexes est une liste d'identifiant prédéfinis en PHP. Aucun des identifiants utilisés ici ne doit être repris comme nom de variable ou de fonction dans vos scripts. Ces identifiants incluent des mots clés, des constantes, des classes, et es variables prédéfinies. Ces listes ne sont pas complètes ou exhaustives.

## 7.8.1 Liste de mots−clé

Ces mots ont un sens spécial en PHP. Certains représentent des objets qui ressemblent à des fonctions, d'autres à des contantes, et ainsi de suite, mais ils n'en sont pas vraiment : ce sont des structures de langage. Vous ne pourrez pas les utiliser comme constantes, nom de classe ou fonctions. Vous pouvez les utiliser comme noms de variables, mais cela risque de vous mener des confusions.

| | | | | | |
|---|---|---|---|---|---|
| and | or | xor | __FILE__ | exception | php_user_filter |
| __LINE__ | array | as | break | case | |
| cfunction | class | const | continue | declare | |
| default | die | do | echo | else | |
| elseif | empty | enddeclare | endfor | endforeach | |
| endif | endswitch | endwhile | eval | exit | |
| extends | for | foreach | function | global | |
| if | include | include_once | isset | list | |
| new | old_function | print | require | require_once | |
| return | static | switch | unset | use | |
| var | while | __FUNCTION__ | __CLASS__ | __METHOD__ | |

## 7.8.2 Variables prédéfinies

Depuis PHP 4.1.0, la méthode recommandée pour accéder à des variables externes est d'utiliser les super−globales mentionnées ci−dessous. Avant cela, il fallait utiliser soit l'option de configuration register_globals , soit les tableaux prédéfinis $HTTP_*_VARS . Depuis PHP 5.0.0, les tableaux prédéfinis PHP peuvent être désactivés avec l'option de configuration register_long_arrays .

### 7.8.2.1 Variables de serveur : $_SERVER

| Note |
|---|
| Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_SERVER_VARS . |

$_SERVER est un tableau contenant des informations comme les entêtes, dossiers et chemins du script. Les entrées de ce tableau sont créées par le serveur web. Il n'y a aucune garantie que tous les serveurs les rempliront tous; certains en oublieront quelques unes et en rajouteront de nouvelles non mentionnées ici. Cependant, un grand nombre de ces variables fait partie des spécifications CGI 1.1 , et vous pouvez donc vous attendre à les retrouver.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_SERVER; pour y accéder, comme vous le faisiez avec $HTTP_SERVER_VARS .

$HTTP_SERVER_VARS contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_SERVER et $HTTP_SERVER_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

You may or may not find any of the following elements in $_SERVER. Note that few, if any, of these will be available (or indeed have any meaning) if running PHP on the command line.

*' PHP_SELF '*

Le nom du fichier du script en cours d'exécution, par rapport au document root. Par exemple, $_SERVER['PHP_SELF'] dans le script situé à l'adresse http://www.monsite.com/test.php/foo.bar sera /test.php/foo.bar .

Si PHP fonctionne en ligne de commande, cette variable n'est pas disponible.

*' argv '*
: Tableau des arguments passées au script. Lorsque le script est appelé en ligne de commande, cela donne accès aux arguments, comme en langage C. Lorsque le script est appelé avec la méthode GET, ce tableau contiendra la chaîne de requête.

*' argc '*
: Contient le nombre de paramètres de la ligne de commande passés au script (si le script fonctionne en ligne de commande).

*' GATEWAY_INTERFACE '*
: Numéro de révision de l'interface CGI du serveur : i.e. ' CGI/1.1 '.

*' SERVER_NAME '*
: Le nom du serveur hôte qui exécute le script suivant. Si le script est exécuté sur un hôte virtuel, ce sera la valeur définie pour cet hôte virtuel.

*' SERVER_SOFTWARE '*
: Chaîne d'identification du serveur, qui est donnée dans les en−têtes lors de la réponse aux requêtes.

*' SERVER_PROTOCOL '*
: Nom et révision du protocole de communication : i.e. ' HTTP/1.0 ';

*' REQUEST_METHOD '*
: Méthode de requête utilisée pour accéder à la page; i.e. ' GET ', ' HEAD ', ' POST ', ' PUT '.

*' QUERY_STRING '*
: La chaîne de requête, si elle existe, qui est utilisée pour accéder à la page.

*' DOCUMENT_ROOT '*
: La racine sous laquelle le script courant est exécuté, comme défini dans la configuration du serveur.

*' HTTP_ACCEPT '*
: Contenu de l'en−tête Accept: de la requête courante, s'il y en a une.

*' HTTP_ACCEPT_CHARSET '*
: Contenu de l'en−tête Accept−Charset: de la requête courante, s'elle existe. Par exemple : ' iso−8859−1,*,utf−8 '.

*' HTTP_ACCEPT_ENCODING '*
: Contenu de l'en−tête Accept−Encoding: de la requête courante, si elle existe. Par exemple : ' gzip '.

*' HTTP_ACCEPT_LANGUAGE '*
: Contenu de l'en−tête Accept−Language: de la requête courante, si elle existe. Par exemple : ' fr '.

*' HTTP_CONNECTION '*
: Contenu de l'en−tête Connection: de la requête courante, si elle existe. Par exemple : ' Keep−Alive '.

*' HTTP_HOST '*
: Contenu de l'en−tête Host: de la requête courante, si elle existe.

*' HTTP_REFERER '*
: L'adresse de la page (si elle existe) qui a conduit le client à la page courante. Cette valeur est affectée par le client, et tous les clients ne le font pas. Certains navigateur permettent même de modifier la valeur de HTTP_REFERER , sous forme de fonctionnalité. En bref, ce n'est pas une valeur de confiance.

*' HTTP_USER_AGENT '*
: Contenu de l'en−tête User_Agent: de la requête courante, si elle existe. C'est une chaîne qui décrit le client HTML utilisé pour voir la page courante. Par exemple :

Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586) . Entre autres choses, vous pouvez utiliser cette valeur avec get_browser pour optimiser votre page en fonction des capacités du client.

*' REMOTE_ADDR '*
: L'adresse IP du client qui demande la page courante.

*' REMOTE_HOST '*
: Le nom de l'hôte qui lit le script courant. La résolution DNS inverse est basée sur la valeur de REMOTE_ADDR .

| Note |
| --- |
| Votre serveur web doit être configuré pour créer cette variable. Par exemple, pour Apache, vous devez ajouter la directive HostnameLookups On dans le fichier httpd.conf , pour que cette variable existe. Voyez aussi gethostbyaddr . |

*' REMOTE_PORT '*
: Le port utilisé par la machine cliente pour communiquer avec le serveur web.

*' SCRIPT_FILENAME '*
: Le chemin absolu jusqu'au script courant.

*' SERVER_ADMIN '*
: La valeur donnée à la directive SERVER_ADMIN (pour Apache), dans le fichier de configuration. Si le script est exécuté par un hôte virtuel, ce sera la valeur définie par l'hôte virtuel.

*' SERVER_PORT '*
: Le port de la machine serveur utilisé pour les communications. Par défaut, c'est '80'. En utilisant SSL, par exemple, il sera remplacé par le numéro de port HTTP sécurisé.

*' SERVER_SIGNATURE '*
: Chaîne contenant le numéro de version du serveur et le nom d'hôte virtuel, qui sont ajoutés aux pages générées par le serveur, si cette option est activée.

*' PATH_TRANSLATED '*
: Chemin dans le système de fichier (pas le document−root) jusqu'au script courant, une fois que le serveur a fait une traduction chemin virtuel −> réel.

*' SCRIPT_NAME '*
: Contient le nom du script courant. Cela sert lorsque les pages doivent s'appeler elles−mêmes.

*' REQUEST_URI '*
: L'URI qui a été fourni pour accéder à cette page. Par exemple : ' /index.html '.

#### 7.8.2.2 Variables d'environnement : $_ENV

| Note |
| --- |
| Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_ENV_VARS . |

Cette variable est importée dans l'espace de nom global de PHP, depuis l'environnement dans lequel l'exécutable PHP fonctionne. De nombreuses valeur sont fournies par le shell qui exécute PHP, et différents systèmes pouvant disposer de différents shell, même un début de liste serait ici impossible. Reportez vous à la documentation de votre shell pour connaître une liste de variables prédéfinies.

Les autres variables d'environnement incluent les variables CGI, placées ici, indépendemment du fait que PHP fonctionne en tant que CGI ou bien que module du serveur.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_ENV; pour y accéder, comme vous le faisiez avec $HTTP_ENV_VARS .

$HTTP_ENV_VARS contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_ENV et $HTTP_ENV_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

#### 7.8.2.3 HTTP Cookies: $_COOKIE

| Note |
| --- |
| Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_COOKIE_VARS . |

Un tableau associatif des valeurs passées au script courant via les cookies HTTP. Elle est automatiquement global dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_COOKIE; pour y accéder, comme vous le faisiez avec $HTTP_COOKIE_VARS .

$HTTP_COOKIE_VARS contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_COOKIE et $HTTP_COOKIE_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

#### 7.8.2.4 HTTP GET variables: $_GET

| Note |
| --- |
| Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_GET_VARS . |

Un tableau associatif des valeurs passées au script courant via le protocole HTTP et la méthode GET. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_GET; pour y accéder, comme vous le faisiez avec $HTTP_GET_VARS .

$HTTP_GET_VARS contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_GET et $HTTP_GET_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

#### 7.8.2.5 HTTP POST variables: $_POST

| Note |
| --- |
| Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_POST_VARS . |

Un tableau associatif des valeurs passées au script courant via le protocole HTTP et la méthode POST. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_POST; pour y accéder, comme vous le faisiez avec $HTTP_POST_VARS .

$HTTP_POST_VARS contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_POST et $HTTP_POST_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

#### 7.8.2.6 Variable de téléchargement de fichier via HTTP : $_FILES

| Note |
| --- |
| Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_POST_FILES . |

Un tableau associatif des valeurs téléchargées au script courant via le protocole HTTP et la méthode POST. Elle est automatiquement globale dans tous les contextes d'exécution.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_FILES; pour y accéder, comme vous le faisiez avec $HTTP_POST_FILES .

$HTTP_POST_FILES contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_FILES et $HTTP_POST_FILES . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

#### 7.8.2.7 Variables de requête : $_REQUEST

| Note |
| --- |
| Introduite en 4.1.0. Dans les versions plus anciennes, elle n'avait pas d'équivalent. |

Un tableau associatif constitué du contenu des variables $_GET , $_POST , $_COOKIE , et $_FILES .

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_REQUEST; pour y accéder.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_REQUEST . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

#### 7.8.2.8 Session variables: $_SESSION

| Note |
| --- |
| Introduite en 4.1.0. Dans les versions plus anciennes, elle s'appelait $HTTP_SESSION_VARS . |

Un tableau associatif des valeurs stockées dans les sessions, et accessible au script courant. Elle est automatiquement globale dans tous les contextes d'exécution. Voyez l'extension Sessions pour plus de détails sur comment est utilisée cette variable.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $_SESSION; pour y accéder, comme vous le faisiez avec $HTTP_SESSION_VARS .

$HTTP_SESSION_VARS contient les mêmes informations, mais n'est pas autoglobale.

Si la directive register_globals est active, alors ces variables seront aussi rendues directement accessible dans le contexte d'exécution global. C'est à dire séparément des tableaux $_SESSION et $HTTP_SESSION_VARS . Pour plus d'informations sur la sécurité de cette configuration, voyez le chapitre Utiliser la directive Register Globals . Ces variables globales individuelles ne sont pas des globales.

#### 7.8.2.9 Variables globales: $GLOBALS

| Note |
| --- |
| $GLOBALS est disponible depuis PHP 3.0.0. |

Un tableau associatif contenant les références sur toutes les variables globales actuellement définies dans le contexte d'exécution global du script. Les noms des variables sont les index du tableau.

Cette variable est une 'superglobal', ou globale automatique. Cela signifie qu'elle est simplement disponible dans tous les contextes d'exécution (fonctions ou méthodes). Vous n'avez pas besoin de faire global $GLOBALS; pour y accéder.

#### 7.8.2.10 Le dernier message d'erreur : $php_errormsg

$php_errormsg est une variable qui contient le texte de la dernière erreur générée par PHP. Cette variable sera uniquement accessible dans le même contexte d'exécution que celui de la ligne qui a généré l'erreur, et uniquement si la directive de configuration track_errors est activée (elle est désactivée par défaut).

### 7.8.3 Classes prédéfinies

#### 7.8.3.1 Classes standards

Ces classes sont définies dans le jeu de classe standard de PHP, inclus dans toute version de PHP.

*Directory*
: La classe qui permet d'instancier dir .

*stdClass*
*__PHP_Incomplete_Class*

#### 7.8.3.2 Classes prédéfinies en PHP 5

Ces classe supplémentaires ont été ajoutées en PHP 5.0.0.

*exception*
*php_user_filter*

#### 7.8.3.3 Classes définies par Ming

Ces classes sont définies dans l'extension Ming et ne seront pas disponibles si PHP a été compilé pour être exécuté dynamiquement, ou bien si elles n'ont pas été compilées dans PHP.

*swfshape*
*swffill*
*swfgradient*
*swfbitmap*
*swftext*
*swftextfield*
*swffont*
*swfdisplayitem*
*swfmovie*
*swfbutton*
*swfaction*
*swfmorph*
*swfsprite*

#### 7.8.3.4 Classes définies par Oracle 8

Ces classes sont définies dans l'extension Oracle 8 et ne seront pas disponibles si PHP a été compilé pour être exécuté dynamiquement, ou bien si elles n'ont pas été compilées dans PHP.

*OCI−Lob*
*OCI−Collection*

#### 7.8.3.5 Classes définies par qtdom

Ces classes sont définies dans l'extension qtdom et ne seront pas disponibles si PHP a été compilé pour être exécuté dynamiquement, ou bien si elles n'ont pas été compilées dans PHP.

*QDomDocument*
*QDomNode*

## 7.8.4 Constantes prédéfinies en PHP

#### 7.8.4.1 constantes prédéfinies de zend() : Constantes définies dans le coeur de PHP, Zend et les modules SAPI

Ces constantes sont définies par le coeur de PHP. Cela inclut notamment PHP, le moteur Zend et les modules SAPI.

*PHP_VERSION (chaîne de caractères)*
*PHP_OS (chaîne de caractères)*
*DEFAULT_INCLUDE_PATH (chaîne de caractères)*
*PEAR_INSTALL_DIR (chaîne de caractères)*
*PEAR_EXTENSION_DIR (chaîne de caractères)*
*PHP_EXTENSION_DIR (chaîne de caractères)*
*PHP_BINDIR (chaîne de caractères)*
*PHP_LIBDIR (chaîne de caractères)*
*PHP_DATADIR (chaîne de caractères)*
*PHP_SYSCONFDIR (chaîne de caractères)*
*PHP_LOCALSTATEDIR (chaîne de caractères)*
*PHP_CONFIG_FILE_PATH (chaîne de caractères)*
*PHP_OUTPUT_HANDLER_START (entier)*
*PHP_OUTPUT_HANDLER_CONT (entier)*
*PHP_OUTPUT_HANDLER_END (entier)*
*E_ERROR (entier)*
*E_WARNING (entier)*
*E_PARSE (entier)*
*E_NOTICE (entier)*
*E_CORE_ERROR (entier)*
*E_CORE_WARNING (entier)*
*E_COMPILE_ERROR (entier)*
*E_COMPILE_WARNING (entier)*
*E_USER_ERROR (entier)*
*E_USER_WARNING (entier)*
*E_USER_NOTICE (entier)*
*E_ALL (entier)*

#### 7.8.4.2 constantes standards prédéfinies() : Constantes standards dans PHP

Ces constantes sont définies par défaut dans PHP.

*EXTR_OVERWRITE (entier)*
*EXTR_SKIP (entier)*
*EXTR_PREFIX_SAME (entier)*
*EXTR_PREFIX_ALL (entier)*
*EXTR_PREFIX_INVALID (entier)*
*EXTR_PREFIX_IF_EXISTS (entier)*
*EXTR_IF_EXISTS (entier)*
*SORT_ASC (entier)*
*SORT_DESC (entier)*
*SORT_REGULAR (entier)*
*SORT_NUMERIC (entier)*
*SORT_STRING (entier)*
*CASE_LOWER (entier)*
*CASE_UPPER (entier)*
*COUNT_NORMAL (entier)*

*COUNT_RECURSIVE (entier)*
*ASSERT_ACTIVE (entier)*
*ASSERT_CALLBACK (entier)*
*ASSERT_BAIL (entier)*
*ASSERT_WARNING (entier)*
*ASSERT_QUIET_EVAL (entier)*
*CONNECTION_ABORTED (entier)*
*CONNECTION_NORMAL (entier)*
*CONNECTION_TIMEOUT (entier)*
*INI_USER (entier)*
*INI_PERDIR (entier)*
*INI_SYSTEM (entier)*
*INI_ALL (entier)*
*M_E (nombre à virgule flottante)*
*M_LOG2E (nombre à virgule flottante)*
*M_LOG10E (nombre à virgule flottante)*
*M_LN2 (nombre à virgule flottante)*
*M_LN10 (nombre à virgule flottante)*
*M_PI (nombre à virgule flottante)*
*M_PI_2 (nombre à virgule flottante)*
*M_PI_4 (nombre à virgule flottante)*
*M_1_PI (nombre à virgule flottante)*
*M_2_PI (nombre à virgule flottante)*
*M_2_SQRTPI (nombre à virgule flottante)*
*M_SQRT2 (nombre à virgule flottante)*
*M_SQRT1_2 (nombre à virgule flottante)*
*CRYPT_SALT_LENGTH (entier)*
*CRYPT_STD_DES (entier)*
*CRYPT_EXT_DES (entier)*
*CRYPT_MD5 (entier)*
*CRYPT_BLOWFISH (entier)*
*DIRECTORY_SEPARATOR (entier)*
*SEEK_SET (entier)*
*SEEK_CUR (entier)*
*SEEK_END (entier)*
*LOCK_SH (entier)*
*LOCK_EX (entier)*
*LOCK_UN (entier)*
*LOCK_NB (entier)*
*HTML_SPECIALCHARS (entier)*
*HTML_ENTITIES (entier)*
*ENT_COMPAT (entier)*
*ENT_QUOTES (entier)*
*ENT_NOQUOTES (entier)*
*INFO_GENERAL (entier)*
*INFO_CREDITS (entier)*
*INFO_CONFIGURATION (entier)*
*INFO_MODULES (entier)*
*INFO_ENVIRONMENT (entier)*
*INFO_VARIABLES (entier)*
*INFO_LICENSE (entier)*
*INFO_ALL (entier)*
*CREDITS_GROUP (entier)*
*CREDITS_GENERAL (entier)*

*CREDITS_SAPI (entier)*
*CREDITS_MODULES (entier)*
*CREDITS_DOCS (entier)*
*CREDITS_FULLPAGE (entier)*
*CREDITS_QA (entier)*
*CREDITS_ALL (entier)*
*STR_PAD_LEFT (entier)*
*STR_PAD_RIGHT (entier)*
*STR_PAD_BOTH (entier)*
*PATHINFO_DIRNAME (entier)*
*PATHINFO_BASENAME (entier)*
*PATHINFO_EXTENSION (entier)*
*CHAR_MAX (entier)*
*LC_CTYPE (entier)*
*LC_NUMERIC (entier)*
*LC_TIME (entier)*
*LC_COLLATE (entier)*
*LC_MONETARY (entier)*
*LC_ALL (entier)*
*LC_MESSAGES (entier)*
*ABDAY_1 (entier)*
*ABDAY_2 (entier)*
*ABDAY_3 (entier)*
*ABDAY_4 (entier)*
*ABDAY_5 (entier)*
*ABDAY_6 (entier)*
*ABDAY_7 (entier)*
*DAY_1 (entier)*
*DAY_2 (entier)*
*DAY_3 (entier)*
*DAY_4 (entier)*
*DAY_5 (entier)*
*DAY_6 (entier)*
*DAY_7 (entier)*
*ABMON_1 (entier)*
*ABMON_2 (entier)*
*ABMON_3 (entier)*
*ABMON_4 (entier)*
*ABMON_5 (entier)*
*ABMON_6 (entier)*
*ABMON_7 (entier)*
*ABMON_8 (entier)*
*ABMON_9 (entier)*
*ABMON_10 (entier)*
*ABMON_11 (entier)*
*ABMON_12 (entier)*
*MON_1 (entier)*
*MON_2 (entier)*
*MON_3 (entier)*
*MON_4 (entier)*
*MON_5 (entier)*
*MON_6 (entier)*
*MON_7 (entier)*
*MON_8 (entier)*

*MON_9 (entier)*
*MON_10 (entier)*
*MON_11 (entier)*
*MON_12 (entier)*
*AM_STR (entier)*
*PM_STR (entier)*
*D_T_FMT (entier)*
*D_FMT (entier)*
*T_FMT (entier)*
*T_FMT_AMPM (entier)*
*ERA (entier)*
*ERA_YEAR (entier)*
*ERA_D_T_FMT (entier)*
*ERA_D_FMT (entier)*
*ERA_T_FMT (entier)*
*ALT_DIGITS (entier)*
*INT_CURR_SYMBOL (entier)*
*CURRENCY_SYMBOL (entier)*
*CRNCYSTR (entier)*
*MON_DECIMAL_POINT (entier)*
*MON_THOUSANDS_SEP (entier)*
*MON_GROUPING (entier)*
*POSITIVE_SIGN (entier)*
*NEGATIVE_SIGN (entier)*
*INT_FRAC_DIGITS (entier)*
*FRAC_DIGITS (entier)*
*P_CS_PRECEDES (entier)*
*P_SEP_BY_SPACE (entier)*
*N_CS_PRECEDES (entier)*
*N_SEP_BY_SPACE (entier)*
*P_SIGN_POSN (entier)*
*N_SIGN_POSN (entier)*
*DECIMAL_POINT (entier)*
*RADIXCHAR (entier)*
*THOUSANDS_SEP (entier)*
*THOUSEP (entier)*
*GROUPING (entier)*
*YESEXPR (entier)*
*NOEXPR (entier)*
*YESSTR (entier)*
*NOSTR (entier)*
*CODESET (entier)*
*LOG_EMERG (entier)*
*LOG_ALERT (entier)*
*LOG_CRIT (entier)*
*LOG_ERR (entier)*
*LOG_WARNING (entier)*
*LOG_NOTICE (entier)*
*LOG_INFO (entier)*
*LOG_DEBUG (entier)*
*LOG_KERN (entier)*
*LOG_USER (entier)*
*LOG_MAIL (entier)*
*LOG_DAEMON (entier)*

*LOG_AUTH (entier)*
*LOG_SYSLOG (entier)*
*LOG_LPR (entier)*
*LOG_NEWS (entier)*
*LOG_UUCP (entier)*
*LOG_CRON (entier)*
*LOG_AUTHPRIV (entier)*
*LOG_LOCAL0 (entier)*
*LOG_LOCAL1 (entier)*
*LOG_LOCAL2 (entier)*
*LOG_LOCAL3 (entier)*
*LOG_LOCAL4 (entier)*
*LOG_LOCAL5 (entier)*
*LOG_LOCAL6 (entier)*
*LOG_LOCAL7 (entier)*
*LOG_PID (entier)*
*LOG_CONS (entier)*
*LOG_ODELAY (entier)*
*LOG_NDELAY (entier)*
*LOG_NOWAIT (entier)*
*LOG_PERROR (entier)*

# 7.9 Types des ressources PHP

Voici la liste des fonctions qui créent, utilisent ou détruisent les ressources PHP. Vous pouvez déterminer si une variable contient une ressource avec la fonction is_resource , et le type d'une ressource que vous utilisez avec la fonction get_resource_type .

| Ressource | Construite par | Utilisé par | Détruite par | Définition |
|---|---|---|---|---|
| aspell | aspell_new | aspell_check , aspell_check_raw , aspell_suggest | None | Dictionnaire Aspell |
| bzip2 | bzopen | bzerrno , bzerror , bzerrstr , bzflush , bzread , bzwrite | bzclose | Fichier bzip2 |
| COM | com_load | com_invoke , com_propget , com_get , com_propput , com_set , com_propput | None | Référence sur un object COM |
| VARIANT | | | | |
| cpdf | cpdf_open | cpdf_page_init , cpdf_finalize_page , cpdf_finalize , cpdf_output_buffer , cpdf_save_to_file , cpdf_set_current_page , cpdf_begin_text , cpdf_end_text , cpdf_show , cpdf_show_xy , cpdf_text , cpdf_set_font , cpdf_set_leading , cpdf_set_text_rendering , cpdf_set_horiz_scaling , cpdf_set_text_rise , cpdf_set_text_matrix , cpdf_set_text_pos , cpdf_set_text_pos , cpdf_set_word_spacing , cpdf_continue_text , cpdf_stringwidth , cpdf_save , cpdf_translate , cpdf_restore , cpdf_scale , cpdf_rotate , cpdf_setflat , cpdf_setlinejoin , cpdf_setlinecap , cpdf_setmiterlimit , cpdf_setlinewidth , cpdf_setdash , cpdf_moveto , cpdf_rmoveto , cpdf_curveto , cpdf_lineto , cpdf_rlineto , cpdf_circle , cpdf_arc , cpdf_rect , cpdf_closepath , cpdf_stroke , cpdf_closepath_fill_stroke , cpdf_fill_stroke , cpdf_clip , cpdf_fill , cpdf_setgray_fill , cpdf_setgray_stroke , cpdf_setgray , cpdf_setrgbcolor_fill , cpdf_setrgbcolor_stroke , cpdf_setrgbcolor , cpdf_add_outline , cpdf_set_page_animation , cpdf_import_jpeg , cpdf_place_inline_image , cpdf_add_annotation | cpdf_close | Document PDF avec la librairie CPDF |
| cpdf outline | | | | |
| curl | curl_init | curl_init , curl_exec | curl_close | Session Curl |
| dbm | dbmopen | dbmexists , dbmfetch , dbminsert , dbmreplace , dbmdelete , dbmfirstkey , dbmnextkey | dbmclose | Lien vers une base de données DBM |
| dba | dba_popen | dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync | dba_close | Lien vers une base de données DBA |
| dba persistent | dba_open | dba_delete , dba_exists , dba_fetch , dba_firstkey , dba_insert , dba_nextkey , dba_optimize , dba_replace , dba_sync | None | Lien persistant vers une base |

| | | | | |
|---|---|---|---|---|
| | | | | de données DBA |
| dbase | dbase_open | dbase_pack , dbase_add_record , dbase_replace_record , dbase_delete_record , dbase_get_record , dbase_get_record_with_names , dbase_numfields , dbase_numrecords | dbase_close | Lien vers une base de données dbase |
| dbx_link_object | dbx_connect | dbx_query | dbx_close | Connexion dbx |
| dbx_result_object | dbx_query | | None | Résultat dbx |
| domxml attribute | | | | |
| domxml document | | | | |
| domxml node | | | | |
| xpath context | | | | |
| xpath object | | | | |
| fbsql database | fbsql_select_db | | None | Base de données fbsql |
| fbsql link | fbsql_change_user , fbsql_connect | fbsql_autocommit , fbsql_change_user , fbsql_create_db , fbsql_data_seek , fbsql_db_query , fbsql_drop_db , , fbsql_select_db , fbsql_errno , fbsql_error , fbsql_insert_id , fbsql_list_dbs | fbsql_close | Lien vers une base de données fbsql |
| fbsql plink | fbsql_change_user , fbsql_pconnect | fbsql_autocommit , fbsql_change_user , fbsql_create_db , fbsql_data_seek , fbsql_db_query , fbsql_drop_db , , fbsql_select_db , fbsql_errno , fbsql_error , fbsql_insert_id , fbsql_list_dbs | None | Lien persistant vers une base de données fbsql |
| fbsql result | fbsql_db_query , fbsql_list_dbs , fbsql_query , fbsql_list_fields , fbsql_list_tables , fbsql_tablename | fbsql_affected_rows , fbsql_fetch_array , fbsql_fetch_assoc , fbsql_fetch_field , fbsql_fetch_lengths , fbsql_fetch_object , fbsql_fetch_row , fbsql_field_flags , fbsql_field_name , fbsql_field_len , fbsql_field_seek , fbsql_field_table , fbsql_field_type , fbsql_next_result , fbsql_num_fields , fbsql_num_rows , fbsql_result , fbsql_num_rows | fbsql_free_result | Résultat fbsql |
| fdf | fdf_open | fdf_create , fdf_save , fdf_get_value , fdf_set_value , fdf_next_field_name , fdf_set_ap , fdf_set_status , fdf_get_status , fdf_set_file , fdf_get_file , fdf_set_flags , fdf_set_opt , fdf_set_submit_form_action , fdf_set_javascript_action | fdf_close | Fichier FDF |
| ftp | ftp_connect | ftp_login , ftp_pwd , ftp_cdup , ftp_chdir , ftp_mkdir , ftp_rmdir , ftp_nlist , ftp_rawlist , ftp_systype , ftp_pasv , ftp_get , ftp_fget , ftp_put , ftp_fput , ftp_size , ftp_mdtm , ftp_rename , ftp_delete , ftp_site | ftp_quit | Connexion FTP |
| gd | imagecreate , imagecreatefromgif , imagecreatefromjpeg , imagecreatefrompng , imagecreatefromwbmp , imagecreatefromstring , imagecreatetruecolor | imagearc , imagechar , imagecharup , imagecolorallocate , imagecolorat , imagecolorclosest , imagecolorexact , imagecolorresolve , imagegammacorrect , imagegammacorrect , imagecolorset , imagecolorsforindex , imagecolorstotal , imagecolortransparent , imagecopy , imagecopyresized , imagedashedline , imagefill , imagefilledpolygon , imagefilledrectangle , imagefilltoborder , imagegif , imagepng , imagejpeg , imagewbmp , imageinterlace , imageline , imagepolygon , imagepstext , imagerectangle , imagesetpixel , imagestring , imagestringup , imagesx , imagesy , imagettftext , imagefilledarc , imageellipse , imagefilledellipse , imagecolorclosestalpha , imagecolorexactalpha , imagecolorresolvealpha , imagecopymerge , imagecopymergegray , imagecopyresampled , imagetruecolortopalette , imagesetbrush , imagesettile , imagesetthickness | imagedestroy | Image GD |
| gd font | imageloadfont | imagechar , imagecharup , imagefontheight | None | Police pour GD |
| gd PS encoding | | | | |
| gd PS font | imagepsloadfont | imagepstext , imagepsslantfont , imagepsextendfont , imagepsencodefont , imagepsbbox | imagepsfreefont | Police PostScript pour GD |
| GMP integer | gmp_init | gmp_intval , gmp_strval , gmp_add , gmp_sub , gmp_mul , gmp_div_q , gmp_div_r , gmp_div_qr , gmp_div , gmp_mod , gmp_divexact , gmp_cmp , gmp_neg , gmp_abs , gmp_sign , gmp_fact , gmp_sqrt , gmp_sqrtrm , gmp_perfect_square , gmp_pow , gmp_powm , gmp_prob_prime , gmp_gcd , gmp_gcdext , gmp_invert , gmp_legendre , gmp_jacobi , gmp_random , gmp_and , gmp_or , gmp_xor , gmp_setbit , gmp_clrbit , gmp_scan0 , gmp_scan1 , gmp_popcount , gmp_hamdist | None | Nombre GMP |
| hyperwave document | hw_cp , hw_docbyanchor , hw_getremote , hw_getremotechildren | hw_children , hw_childrenobj , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getsrcbydestobj , hw_getandlock , hw_gettext , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_inscoll , hw_pipedocument , hw_unlock | hw_deleteobject | Objet Hyperwave |
| hyperwave link | hw_connect | hw_children , hw_childrenobj , hw_cp , hw_deleteobject , hw_docbyanchor , hw_docbyanchorobj , hw_errormsg , hw_edittext , hw_error , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getremotechildren , hw_getsrcbydestobj , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobj , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , | hw_close , hw_free_document | Lien vers un serveur Hyperwave |

| | | | | |
|---|---|---|---|---|
| | | hw_getanchorsobj , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdoc , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername | | |
| hyperwave link persistent | hw_pconnect | hw_children , hw_childrenobj , hw_cp , hw_deleteobject , hw_docbyanchor , hw_docbyanchorobj , hw_errormsg , hw_edittext , hw_error , hw_getparents , hw_getparentsobj , hw_getchildcoll , hw_getchildcollobj , hw_getremote , hw_getremotechildren , hw_getsrcbydestobj , hw_getobject , hw_getandlock , hw_gettext , hw_getobjectbyquery , hw_getobjectbyqueryobj , hw_getobjectbyquerycoll , hw_getobjectbyquerycollobj , hw_getchilddoccoll , hw_getchilddoccollobj , hw_getanchors , hw_getanchorsobj , hw_mv , hw_incollections , hw_info , hw_inscoll , hw_insdoc , hw_insertdocument , hw_insertobject , hw_mapid , hw_modifyobject , hw_pipedocument , hw_unlock , hw_who , hw_getusername | None | Lien persistant vers un serveur Hyperwave |
| icap | icap_open | icap_fetch_event , icap_list_events , icap_store_event , icap_snooze , icap_list_alarms , icap_delete_event | icap_close | Lien vers un serveur icap |
| imap | imap_open | imap_append , imap_body , imap_check , imap_createmailbox , imap_delete , imap_deletemailbox , imap_expunge , imap_fetchbody , imap_fetchstructure , imap_headerinfo , imap_header , imap_headers , imap_listmailbox , imap_getmailboxes , imap_get_quota , imap_status , imap_listsubscribed , imap_set_quota , imap_set_quota , imap_getsubscribed , imap_mail_copy , imap_mail_move , imap_num_msg , imap_num_recent , imap_ping , imap_renamemailbox , imap_reopen , imap_subscribe , imap_undelete , imap_unsubscribe , imap_scanmailbox , imap_mailboxmsginfo , imap_fetchheader , imap_uid , imap_msgno , imap_search , imap_fetch_overview | imap_close | Lien vers un serveur mail (IMAP, POP3) |
| imap chain persistent | | | | |
| imap persistent | | | | |
| ingres | ingres_connect | ingres_query , ingres_num_rows , ingres_num_fields , ingres_field_name , ingres_field_type , ingres_field_nullable , ingres_field_length , ingres_field_precision , ingres_field_scale , ingres_fetch_array , ingres_fetch_row , ingres_fetch_object , ingres_rollback , ingres_commit , ingres_autocommit | ingres_close | Lien vers une base de données ingresII |
| ingres persistent | ingres_pconnect | ingres_query , ingres_num_rows , ingres_num_fields , ingres_field_name , ingres_field_type , ingres_field_nullable , ingres_field_length , ingres_field_precision , ingres_field_scale , ingres_fetch_array , ingres_fetch_row , ingres_fetch_object , ingres_rollback , ingres_commit , ingres_autocommit | None | Lien persistant vers une base de données ingresII |
| interbase blob | | | | |
| interbase link | ibase_connect | ibase_query , ibase_prepare , ibase_trans | ibase_close | Lien vers une base de données Interbase |
| interbase link persistent | ibase_pconnect | ibase_query , ibase_prepare , ibase_trans | None | Lien persistant vers une base de données Interbase |
| interbase query | ibase_prepare | ibase_execute | ibase_free_query | Requête Interbase |
| interbase result | ibase_query | ibase_fetch_row , ibase_fetch_object , ibase_field_info , ibase_num_fields | ibase_free_result | Résultat Interbase |
| interbase transaction | ibase_trans | ibase_commit | ibase_rollback | Transaction Interbase |
| java | | | | |
| ldap link | ldap_connect , ldap_search | ldap_count_entries , ldap_first_attribute , ldap_first_entry , ldap_get_attributes , ldap_get_dn , ldap_get_entries , ldap_get_values , ldap_get_values_len , ldap_next_attribute , ldap_next_entry | ldap_close | Lien vers un serveur LDAP |
| ldap result | ldap_read | ldap_add , ldap_compare , ldap_bind , ldap_count_entries , ldap_delete , ldap_errno , ldap_error , ldap_first_attribute , ldap_first_entry , ldap_get_attributes , ldap_get_dn , ldap_get_entries , ldap_get_values , ldap_get_values_len , ldap_get_option , ldap_list , ldap_modify , ldap_mod_add , ldap_mod_replace , ldap_next_attribute , ldap_next_entry , ldap_mod_del , ldap_set_option , ldap_unbind | ldap_free_result | Résultat de recherche LDAP |
| ldap result entry | | | | |
| mcal | mcal_open , mcal_popen | mcal_create_calendar , mcal_rename_calendar , mcal_rename_calendar , mcal_delete_calendar , mcal_fetch_event , mcal_list_events , mcal_append_event , mcal_store_event , mcal_delete_event , mcal_list_alarms , mcal_event_init , mcal_event_set_category , mcal_event_set_title , mcal_event_set_description , mcal_event_set_start , mcal_event_set_end , mcal_event_set_alarm , mcal_event_set_class , mcal_next_recurrence , mcal_event_set_recur_none , mcal_event_set_recur_daily , mcal_event_set_recur_weekly , | mcal_close | Lien vers un serveur Mcal |

| | | | | |
|---|---|---|---|---|
| | | mcal_event_set_recur_monthly_mday , mcal_event_set_recur_monthly_wday , mcal_event_set_recur_yearly , mcal_fetch_current_stream_event , mcal_event_add_attribute , mcal_expunge | | |
| SWFAction | | | | |
| SWFBitmap | | | | |
| SWFButton | | | | |
| SWFDisplayItem | | | | |
| SWFFill | | | | |
| SWFFont | | | | |
| SWFGradient | | | | |
| SWFMorph | | | | |
| SWFMovie | | | | |
| SWFShape | | | | |
| SWFSprite | | | | |
| SWFText | | | | |
| SWFTextField | | | | |
| mnogosearch agent | | | | |
| mnogosearch result | | | | |
| msql link | msql_connect | msql , msql_create_db , msql_createdb , msql_drop_db , msql_drop_db , msql_select_db , msql_select_db | msql_close | Lien vers une base de données mSQL |
| msql link persistent | msql_pconnect | msql , msql_create_db , msql_createdb , msql_drop_db , msql_drop_db , msql_select_db , msql_select_db | None | Lien persistant vers une base de données mSQL |
| msql query | msql_query | msql , msql_affected_rows , msql_data_seek , msql_dbname , msql_fetch_array , msql_fetch_field , msql_fetch_object , msql_fetch_row , msql_fieldname , msql_field_seek , msql_fieldtable , msql_fieldtype , msql_fieldflags , msql_fieldlen , msql_num_fields , msql_num_rows , msql_numfields , msql_numrows , msql_result | msql_free_result , msql_free_result | Résultat mSQL |
| mssql link | mssql_connect | mssql_query , mssql_select_db | mssql_close | Lien vers une base de données Microsft SQL Server |
| mssql link persistent | mssql_pconnect | mssql_query , mssql_select_db | None | Lien persistant vers une base de données Microsft SQL Server |
| mssql result | mssql_query | mssql_data_seek , mssql_fetch_array , mssql_fetch_field , mssql_fetch_object , mssql_fetch_row , mssql_field_length , mssql_field_name , mssql_field_seek , mssql_field_type , mssql_num_fields , mssql_num_rows , mssql_result | mssql_free_result | Résultat Microsft SQL Server |
| mysql link | mysql_connect | mysql_affected_rows , mysql_change_user , mysql_create_db , mysql_data_seek , mysql_db_name , mysql_db_query , mysql_drop_db , mysql_errno , mysql_error , mysql_insert_id , mysql_list_dbs , mysql_list_fields , mysql_list_tables , mysql_query , mysql_result , mysql_select_db , mysql_tablename , mysql_get_host_info , mysql_get_proto_info , mysql_get_server_info | mysql_close | Lien vers une base de données MySQL |
| mysql link persistent | mysql_pconnect | mysql_affected_rows , mysql_change_user , mysql_create_db , mysql_data_seek , mysql_db_name , mysql_db_query , mysql_drop_db , mysql_errno , mysql_error , mysql_insert_id , mysql_list_dbs , mysql_list_fields , mysql_list_tables , mysql_query , mysql_result , mysql_select_db , mysql_tablename , mysql_get_host_info , mysql_get_proto_info , mysql_get_server_info | None | Lien persistant vers une base de données MySQL |
| mysql result | mysql_db_query , mysql_list_dbs , mysql_list_fields , mysql_list_tables , mysql_query | mysql_data_seek , mysql_db_name , mysql_fetch_array , mysql_fetch_assoc , mysql_fetch_field , mysql_fetch_lengths , mysql_fetch_object , mysql_fetch_row , mysql_fetch_row , mysql_field_flags , mysql_field_name , mysql_field_len , mysql_field_seek , mysql_field_table , mysql_field_type , mysql_num_fields , mysql_num_rows , mysql_result , mysql_tablename | mysql_free_result | Résultat MySQL |
| oci8 collection | | | | |
| oci8 connection | ocilogon , ociplogon , ocinlogon | ocicommit , ociserverversion , ocinewcursor , ociparse , ocierror | ocilogoff | Lien vers une base de données Oracle |
| oci8 descriptor | | | | |
| oci8 server | | | | |
| oci8 session | | | | |
| oci8 statement | ocinewdescriptor | ocirollback , ocinewdescriptor , ocirowcount , ocidefinebyname , ocibindbyname , ociexecute , ocinumcols , ociresult , ocifetch , ocifetchinto , ocifetchstatement , ocicolumnisnull , ocicolumnname , ocicolumnsize , ocicolumntype , ocistatementtype , | ocifreestatement | Curseur Oracle |

| | | | | |
|---|---|---|---|---|
| | | ocierror | | |
| odbc link | odbc_connect | odbc_autocommit , odbc_commit , odbc_error , odbc_errormsg , odbc_exec , odbc_tables , odbc_tableprivileges , odbc_do , odbc_prepare , odbc_columns , odbc_columnprivileges , odbc_procedurecolumns , odbc_specialcolumns , odbc_rollback , odbc_setoption , odbc_gettypeinfo , odbc_primarykeys , odbc_foreignkeys , odbc_procedures , odbc_statistics | odbc_close | Lien vers une base de données ODBC |
| odbc link persistent | odbc_connect | odbc_autocommit , odbc_commit , odbc_error , odbc_errormsg , odbc_exec , odbc_tables , odbc_tableprivileges , odbc_do , odbc_prepare , odbc_columns , odbc_columnprivileges , odbc_procedurecolumns , odbc_specialcolumns , odbc_rollback , odbc_setoption , odbc_gettypeinfo , odbc_primarykeys , odbc_foreignkeys , odbc_procedures , odbc_statistics | None | Lien persistant vers une base de données ODBC |
| odbc result | odbc_prepare | odbc_binmode , odbc_cursor , odbc_execute , odbc_fetch_into , odbc_fetch_row , odbc_field_name , odbc_field_num , odbc_field_type , odbc_field_len , odbc_field_precision , odbc_field_scale , odbc_longreadlen , odbc_num_fields , odbc_num_rows , odbc_result , odbc_result_all , odbc_setoption | odbc_free_result | Résultat ODBC |
| velocis link | | | | |
| velocis result | | | | |
| OpenSSL key | openssl_get_privatekey , openssl_get_publickey | openssl_sign , openssl_seal , openssl_open , openssl_verify | openssl_free_key | Clé de cryptage OpenSSL |
| OpenSSL X.509 | openssl_x509_read | openssl_x509_parse , openssl_x509_checkpurpose | openssl_x509_free | Clé publique |
| oracle cursor | ora_open | ora_bind , ora_columnname , ora_columnsize , ora_columntype , ora_error , ora_errorcode , ora_exec , ora_fetch , ora_fetch_into , ora_getcolumn , ora_numcols , ora_numrows , ora_parse | ora_close | Curseur oracle |
| oracle link | ora_logon | ora_do , ora_error , ora_errorcode , ora_rollback , ora_commitoff , ora_commiton , ora_open , ora_commit | ora_logoff | Lien vers une base de données oracle |
| oracle link persistent | ora_plogon | ora_do , ora_error , ora_errorcode , ora_rollback , ora_commitoff , ora_commiton , ora_open , ora_commit | None | Lien persistant vers une base de données oracle |
| pdf document | pdf_new | pdf_add_bookmark , pdf_add_launchlink , pdf_add_locallink , pdf_add_note , pdf_add_pdflink , pdf_add_weblink , pdf_arc , pdf_attach_file , pdf_begin_page , pdf_circle , pdf_clip , pdf_closepath , pdf_closepath_fill_stroke , pdf_closepath_stroke , pdf_concat , pdf_continue_text , pdf_curveto , pdf_end_page , pdf_endpath , pdf_fill , pdf_fill_stroke , pdf_findfont , pdf_get_buffer , pdf_get_image_height , pdf_get_image_width , pdf_get_parameter , pdf_get_value , pdf_lineto , pdf_moveto , pdf_open_ccitt , pdf_open_file , pdf_open_image_file , pdf_place_image , pdf_rect , pdf_restore , pdf_rotate , pdf_save , pdf_scale , pdf_setdash , pdf_setflat , pdf_setfont , pdf_setgray , pdf_setgray_fill , pdf_setgray_stroke , pdf_setlinecap , pdf_setlinejoin , pdf_setlinewidth , pdf_setmiterlimit , pdf_setpolydash , pdf_setrgbcolor , pdf_setrgbcolor_fill , pdf_setrgbcolor_stroke , pdf_set_border_color , pdf_set_border_dash , pdf_set_border_style , pdf_set_char_spacing , pdf_set_duration , pdf_set_font , pdf_set_horiz_scaling , pdf_set_parameter , pdf_set_text_pos , pdf_set_text_rendering , pdf_set_value , pdf_set_word_spacing , pdf_show , pdf_show_boxed , pdf_show_xy , pdf_skew , pdf_stringwidth , pdf_stroke , pdf_translate , pdf_open_memory_image | pdf_close , pdf_delete | Document PDF |
| pdf image | pdf_open_image , pdf_open_image_file , pdf_open_memory_image | pdf_get_image_height , pdf_get_image_width , pdf_open_ccitt , pdf_place_image | pdf_close_image | Image dans un document PDF |
| pdf object | | | | |
| pdf outline | | | | |
| pgsql large object | pg_getlastoid , pg_loimport , pg_loimport | pg_loopen , pg_getlastoid , pg_locreate , pg_loexport , pg_loread , pg_loreadall , pg_lounlink , pg_lowrite | pg_loclose | Objet de grande taille PostGreSQL |
| pgsql link | pg_connect | pg_cmdtuples , pg_dbname , pg_end_copy , pg_errormessage , pg_host , pg_locreate , pg_loexport , pg_loimport , pg_loopen , pg_lounlink , pg_options , pg_port , pg_put_line , pg_set_client_encoding , pg_client_encoding , pg_trace , pg_untrace , pg_tty | pg_close | Lien vers une base de données PostGreSQL |
| pgsql link persistent | pg_pconnect | pg_cmdtuples , pg_dbname , pg_end_copy , pg_errormessage , pg_host , pg_locreate , pg_loexport , pg_loimport , pg_loopen , pg_lounlink , pg_options , pg_port , pg_put_line , pg_set_client_encoding , pg_client_encoding , pg_trace , pg_untrace , pg_tty | None | Lien persistant vers une base de données PostGreSQL |
| pgsql result | pg_exec | pg_fetch_array , pg_fetch_object , pg_fieldisnull , pg_fetch_row , pg_fieldname , pg_fieldnum , pg_fieldprtlen , pg_fieldsize , pg_fieldtype , pg_getlastoid , pg_numfields , pg_result , pg_numrows | pg_freeresult | Résultat PostGreSQL |
| pgsql string | | | | |

| | | | | |
|---|---|---|---|---|
| printer | | | | |
| printer brush | | | | |
| printer font | | | | |
| printer pen | | | | |
| pspell | pspell_new , pspell_new_config , pspell_new_personal | pspell_add_to_personal , pspell_add_to_session , pspell_check , pspell_clear_session , pspell_config_ignore , pspell_config_mode , pspell_config_personal , pspell_config_repl , pspell_config_runtogether , pspell_config_save_repl , pspell_save_wordlist , pspell_store_replacement , pspell_suggest | None | Dictionnaire Pspell |
| pspell config | pspell_config_create | pspell_new_config | None | Configuration Pspell |
| Sablotron XSLT | xslt_create | xslt_closelog , xslt_openlog , xslt_run , xslt_set_sax_handler , xslt_errno , xslt_error , xslt_fetch_result , xslt_free | xslt_free | Analyseur XSLT |
| shmop | shmop_open | shmop_read , shmop_write , shmop_size , shmop_delete | shmop_close | |
| sockets file descriptor set | socket | accept_connect , bind , connect , listen , read , write | close | Socket |
| sockets i/o vector | | | | |
| dir | dir | readdir , rewinddir | closedir | Dossier |
| file | fopen | feof , fflush , fgetc , fgetcsv , fgets , fgetss , flock , fpassthru , fputs , fwrite , fread , fseek , ftell , fstat , ftruncate , set_file_buffer , rewind | fclose | Fichier |
| pipe | popen | feof , fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread | pclose | Processus |
| socket | fsockopen | fflush , fgetc , fgetcsv , fgets , fgetss , fpassthru , fputs , fwrite , fread | fclose | Socket |
| stream | | | | |
| sybase-db link | sybase_connect | sybase_query , sybase_select_db | sybase_close | Lien vers une base de données Sybase avec la librairie db |
| sybase-db link persistent | sybase_pconnect | sybase_query , sybase_select_db | None | Lien persistant vers une base de données Sybase avec la librairie db |
| sybase-db result | sybase_query | sybase_data_seek , sybase_fetch_array , sybase_fetch_field , sybase_fetch_object , sybase_fetch_row , sybase_field_seek , sybase_num_fields , sybase_num_rows , sybase_result | sybase_free_result | Résultat Sybase avec la librairie DB |
| sybase-ct link | sybase_connect | sybase_affected_rows , sybase_query , sybase_select_db | sybase_close | Lien vers une base de données Sybase avec la librairie CT |
| sybase-ct link persistent | sybase_pconnect | sybase_affected_rows , sybase_query , sybase_select_db | None | Lien persistant vers une base de données Sybase avec la librairie CT |
| sybase-ct result | sybase_query | sybase_data_seek , sybase_fetch_array , sybase_fetch_field , sybase_fetch_object , sybase_fetch_row , sybase_field_seek , sybase_num_fields , sybase_num_rows , sybase_result | sybase_free_result | Résultat Sybase avec la librairie CT |
| sysvsem | sem_get | sem_acquire | sem_release | Sémaphore Système V |
| sysvshm | shm_attach | shm_remove , shm_put_var , shm_get_var , shm_remove_var | shm_detach | Mémoire partagée Système V |
| wddx | wddx_packet_start | wddx_add_vars | wddx_packet_end | Paquet wddx |
| xml | xml_parser_create | xml_set_object , xml_set_element_handler , xml_set_character_data_handler , xml_set_processing_instruction_handler , xml_set_default_handler , xml_set_unparsed_entity_decl_handler , xml_set_notation_decl_handler , xml_set_external_entity_ref_handler , xml_parse , xml_get_error_code , xml_error_string , xml_get_current_line_number , xml_get_current_column_number , xml_get_current_byte_index , xml_parse_into_struct , xml_parser_set_option , xml_parser_get_option | xml_parser_free | Analyseur syntaxique XML |
| zlib | gzopen | gzeof , gzgetc , gzgets , gzgetss , gzpassthru , gzputs , gzread , gzrewind , gzseek , gztell , gzwrite | gzclose | Fichier compressé Gzip |
| ZZIP Directory | | | | |
| ZZIP Entry | | | | |

# 7.10 Liste des protocoles supportés

Cette section recense une liste de protocole gérant des URL, qui sont intégrés dans les fonctions ayant accès aux fichiers, dans PHP. Par exemple, les fonctions fopen et copy . Ces fonctionnalités sont compilées comme des pilotes externes, et à partir de PHP 4.3.0, vous pouvez créer vos propres pilotes, avec la fonction stream_register_wrapper .

## 7.10.1 HTTP et HTTPS

PHP 3, PHP 4. https:// depuis PHP 4.3

- http://exemple.com
- http://utilisateur:motdepasse@exemple.com
- https://exemple.com
- https://utilisateur:motdepasse@exemple.com

Permet des accès en lecture uniquement, pour des fichiers accessibles sur le réseau avec la méthode GET de HTTP 1.0. Un en−tête Host: est envoyé avec la requête, pour gérer les hôtes virtuels, basés sur des noms. Si vous avez configuré une version de navigateur avec l'option user_agent dans votre fichier php.ini , ou via le contexte de flux, il sera aussi utilisé dans votre requête.

Les redirections sont supportées depuis PHP 4.0.5; si vous utilisez une version plus ancienne, vous devrez ajouter vous même les slash finaux dans vos URL.

Le flux donne l'accès au corps ( **body** ) de la ressource. Les en−têtes sont stockés dans la variable $http_response_header . Depuis PHP 4.3.0, les en−têtes sont aussi accessibles avec la fonction stream_get_meta_data .

Les connexions HTTP sont en lecture seule. Vous ne pouvez pas écrire dans des fichiers ouverts via le protocole HTTP.

| Note |
| --- |
| HTTPS est supporté depuis PHP 4.3, si vous avez compilé le support d'OpenSSL. |

## 7.10.2 FTP and FTPS

PHP 3, PHP 4. ftps:// depuis PHP 4.3

- ftp://example.com/pub/file.txt
- ftp://user:password@example.com/pub/file.txt
- ftps://example.com/pub/file.txt
- ftps://user:password@example.com/pub/file.txt

Permet l'accès aux fichiers existants, et la création de fichiers via FTP. Si le serveur ne supporte pas les connexions en mode passif, la connexion échouera.

Vous pouvez ouvrir des fichiers en lecture et en écriture, mais pas les deux en même temps. Si le fichier distant existe déjà sur le serveur FTP, vous pourrez essayer de l'ouvrir en écriture, mais la connexion échouera. Si vous devez modifier un fichier qui existe déjà, utilisez ftp_connect .

ftps:// a été introduit en PHP 4.3. C'est le même que ftp:// , mais il tente d'établir une connexion sécurisée avec le serveur FTP. Si le serveur ne le supporte pas, la connexion sera une connexion non sécurisée.

| Note |
| --- |
| FTPS est supporté depuis PHP 4.3.0, si vous avez compilé le support d'OpenSSL. |

## 7.10.3 Flôts d'entrée/sortie

PHP 3.0.13 et plus réscent, php://output et php://input depuis PHP 4.3

- php://stdin
- php://stdout
- php://stderr
- php://output
- php://input

php://stdin , php://stdout et php://stderr donne accès au flux correspondant d'entrée ou de sortie du processus PHP.

php://output vous permet d'écrire dans le buffer de sortie, de la même manière que print et echo .

php://input permet de lire des données POST bruts. C'est moins gourmand en mémoire que $HTTP_RAW_POST_DATA et il n'y a pas de directive spéciale dans php.ini .

php://stdin et php://input sont lecture seule, tandis que php://stdout , php://stderr et php://output sont en écriture seule.

php://filter est une sorte de méta−gestionnaire, prévu pour qui permet l'utilisation de filtre avec les données d'entrée au moment du démarrage du script. C'est pratique avec des fonctions compactes comme readfile , file et file_get_contents où il n'y a pas d'opportunité d'appliquer un filtre aux données lues.

Le gestionaire de php://filter prend les paramètres suivants dans le chemin.

- /resource=<flux à filtrer> ( **obligatoire** ) Ce paramètre doit être situé à la fin de votre chaîne php://filter et doit pointer sur le flux à filtrer.

```
<?php
/* Ceci est l'équivalent de :
   readfile("http://www.example.com");
   puisqu'aucun filtre n'est spécifié */

readfile("php://filter/resource=http://www.exemple.com");
?>
```

- /read=<liste de filtres de la chaîne de lecture> ( **optionnel** ) Ce paramètre prend un ou plusieurs noms de filtres, séparés par le caractère pipe | .

```
<?php
/* Cette ligne va afficher le contenu de
    www.exemple.com tout en majuscule */
readfile("php://filter/read=string.toupper/resource=http://www.exemple.com");

/* Cela va faire la même chose que ci−dessus, mais en plus,
   le résultat sera chiffré en ROT13 */
readfile("php://filter/read=string.toupper|string.rot13/resource=http://www.exemple.com");
?>
```

- /write=<liste de filtres de la chaîne d'écriture> ( **optionnel** ) Ce paramètre prend un ou plusieurs noms de filtres, séparés par le caractère pipe | .

```
<?php
/* Cette ligne va filtrer la chaîne "Bonjour le monde"
   avec le filtre rot13, puis écrire le résultat dans
    le fichier exemple.txt du dossier courant */
```

```
file_set_contents("php://filter/write=string.rot13/resource=exemple.txt","Bonjour le monde");
?>
```

- /<liste de filtres des deux chaînes> ( **optionnel** ) Tous les filtres qui ne sont pas préfixé avec read= ou write= seront appliqué aux deux chaînes de filtres, en lecture et en écriture.

### 7.10.4 Flôts compressés

zlib: PHP 4.0.4 − PHP 4.2.3 (systèmes avec fopencookie uniquement)

compress.zlib:// et compress.bzip2:// depuis PHP 4.3.0 et plus récent.

- zlib:
- compress.zlib://
- compress.bzip2://

zlib: fonctionne comme gzopen , hormis le fait que le flux peut être utilisé directement avec fread et les autres fonctions de système de fichier. Cette notation est obsolète depuis PHP 4.3.0, étant données les ambiguités dues aux noms de fichiers contenant des deux points ':'. Utilisez plutôt compress.zlib:// .

compress.zlib:// et compress.bzip2:// sont équivalents à gzopen et bzopen respectivement, et ils opèrent même sur les systèmes qui ne supportent pas fopencookie.

## 7.11 Liste des token de l'analyseur

De nombreuses parties du langage PHP sont représentées en interne par des trucs (NDT : litéralement) comme T_SR. PHP affiche des identifiants comme celui−ci dans les erreurs d'analyse, comme par exemple : unexpected T_SR, expecting ',' or ';' in script.php on line 10. .

Vous êtes suppposé savoir ce que T_SR signifie. Pour ceux qui ne le savent pas, voici une liste de ces identifiant, la syntaxe PHP attenante, et les références appropriées au manuel.

| Token | Syntaxe | Référence |
|---|---|---|
| T_AND_EQUAL | | opérateurs d'assignation |
| T_ARRAY | array() | array , syntaxe de tableau |
| T_ARRAY_CAST | (array) | transtypage |
| T_AS | as | foreach |
| T_BAD_CHARACTER | | Tous les caractères en dessous de ASCII 32 excepté \t (0x09), \n (0x0a) et \r (0x0d) |
| T_BOOLEAN_AND | | opérateurs logiques |
| T_BOOLEAN_OR | \|\| | opérateurs logiques |
| T_BOOL_CAST | (bool) or (boolean) | transtypage |
| T_BREAK | break; | break |
| T_CASE | case | switch |
| T_CHARACTER | | |
| T_CLASS | class | classes et objets |
| T_CLOSE_TAG | ?> ou %> | |
| T_COMMENT | // ou # | commentaires |
| T_CONCAT_EQUAL | .= | opérateurs d'assignation |
| T_CONST | const | |
| T_CONSTANT_ENCAPSED_STRING | "foo" or 'bar' | syntaxe chaîne de caractères |
| T_CONTINUE | continue | |
| T_CURLY_OPEN | | |
| T_DEC | −− | opérateurs d'incrémention/décrémention |
| T_DECLARE | declare | declare |
| T_DEFAULT | default | switch |
| T_DIV_EQUAL | /= | opérateurs d'assignation |
| T_DNUMBER | 0.12, etc | nombres à virgule flottante |

| | | |
|---|---|---|
| T_DO | do | do..while |
| T_DOLLAR_OPEN_CURLY_BRACES | ${ | syntaxe de variable complexe analysée |
| T_DOUBLE_ARROW | => | syntaxe de tableau |
| T_DOUBLE_CAST | (real), (double) or (float) | transtypage |
| T_ECHO | echo | echo |
| T_ELSE | else | else |
| T_ELSEIF | elseif | elseif |
| T_EMPTY | empty | empty |
| T_ENCAPSED_AND_WHITESPACE | | |
| T_ENDDECLARE | enddeclare | declare, syntaxe alternative |
| T_ENDFOR | endfor | for, syntaxe alternative |
| T_ENDFOREACH | endforeach | foreach, syntaxe alternative |
| T_ENDIF | endif | if, syntaxe alternative |
| T_ENDSWITCH | endswitch | switch, syntaxe alternative |
| T_ENDWHILE | endwhile | while, syntaxe alternative |
| T_END_HEREDOC | | syntaxe heredoc |
| T_EVAL | eval() | eval |
| T_EXIT | exit or die | exit , die |
| T_EXTENDS | extends | extends, classes et objets |
| T_FILE | __FILE__ | constantes |
| T_FOR | for | for |
| T_FOREACH | foreach | foreach |
| T_FUNCTION | function or cfunction | fonctions |
| T_GLOBAL | global | scope de variable |
| T_IF | if | if |
| T_INC | ++ | opérateurs d'incrémention/décrémention |
| T_INCLUDE | include() | include |
| T_INCLUDE_ONCE | include_once() | include_once |
| T_INLINE_HTML | | |
| T_INT_CAST | (int) or (integer) | transtypage |
| T_ISSET | isset() | isset |
| T_IS_EQUAL | == | opérateurs de comparaison |
| T_IS_GREATER_OR_EQUAL | >= | opérateurs de comparaison |
| T_IS_IDENTICAL | === | opérateurs de comparaison |
| T_IS_NOT_EQUAL | != or <> | opérateurs de comparaison |
| T_IS_NOT_IDENTICAL | !== | opérateurs de comparaison |
| T_SMALLER_OR_EQUAL | <= | opérateurs de comparaison |
| T_LINE | __LINE__ | constants |
| T_LIST | list() | list |
| T_LNUMBER | 123, 012, 0x1ac, etc | entiers |
| T_LOGICAL_AND | and | opérateurs logiques |
| T_LOGICAL_OR | or | opérateurs logiques |
| T_LOGICAL_XOR | xor | opérateurs logiques |
| T_MINUS_EQUAL | -= | opérateurs d'assignation |
| T_ML_COMMENT | /* and */ | commentaires |
| T_MOD_EQUAL | %= | opérateurs d'assignation |
| T_MUL_EQUAL | *= | opérateurs d'assignation |
| T_NEW | new | classes et objets |
| T_NUM_STRING | | |
| T_OBJECT_CAST | (object) | transtypage |
| T_OBJECT_OPERATOR | -> | classes et objets |
| T_OLD_FUNCTION | old_function | old_function |
| T_OPEN_TAG | <?php, <? or <% | sortie du mode HTML |
| T_OPEN_TAG_WITH_ECHO | <?= or <%= | sortie du mode HTML |
| T_OR_EQUAL | \|= | opérateurs d'assignation |
| T_PAAMAYIM_NEKUDOTAYIM | :: | :: |
| T_PLUS_EQUAL | += | opérateurs d'assignation |
| T_PRINT | print() | print |
| T_REQUIRE | require() | require |
| T_REQUIRE_ONCE | require_once() | require_once |
| T_RETURN | return | valeurs retournées |
| T_SL | << | opérateurs sur les bits |
| T_SL_EQUAL | <<= | opérateurs d'assignation |
| T_SR | >> | opérateurs sur les bits |
| T_SR_EQUAL | >>= | opérateurs d'assignation |
| T_START_HEREDOC | <<< | syntaxe heredoc |
| T_STATIC | static | scope de variable |
| T_STRING | | |
| T_STRING_CAST | (string) | transtypage |
| T_STRING_VARNAME | | |

| | | |
|---|---|---|
| T_SWITCH | switch | switch |
| T_UNSET | unset() | unset |
| T_UNSET_CAST | (unset) | (non documenté; forcé à NULL ) |
| T_USE | use | (pas implémenté) |
| T_VAR | var | classes et objets |
| T_VARIABLE | $foo | variables |
| T_WHILE | while | while, do..while |
| T_WHITESPACE | | |
| T_XOR_EQUAL | ^= | opérateurs d'assignation |
| T_FUNC_C | __FUNCTION__ | constantes, depuis PHP 4.3.0 |
| T_CLASS_C | __CLASS__ | constantes, depuis PHP 4.3.0 |

# 7.12 A propos du manuel

## 7.12.1 Formats

Le manuel PHP est fourni en différents formats. Ces formats sont divisés en deux groupes : ceux qui sont disponibles en ligne, et les téléchargeables.

| Note |
|---|
| Certains éditeurs ont fournis des versions imprimées de ces manuels. Nous n'en recommandons aucune, car elles sont rapidement obsolètes. |

Vous pouvez lire le manuel en ligne sur le site http://www.php.net/ et sur les nombreux miroirs. Pour un confort d'utilisation accru, il est recommandé d'utiliser un miroir proche de chez vous. Vous pouvez le voir en version HTML simple (et facilement imprimable), ou bien aux couleurs de PHP .net.

Un des points forts du manuel en ligne sur la plupart des formats téléchargeable, est l'intégration des notes des utilisateurs . Un inconvénient évident est que vous devez être en ligne pour profiter de ces formats.

Il y a de nombreux formats de manuel pour la consultation hors ligne, et le format le plus approprié dépend de votre OS et de vos goûts personnels. Pour savoir comment le manuel est généré, lisez la section 'Comment le manuel est généré' .

Le format le plus portable est le manuel au format HTML. Le manuel est fourni dans un format en une seule page HTML, ou comme un ensemble de fichier de taille réduite (mais un bon millier de fichiers tout de même). Les formats HTML et texte simple sont disponibles sous forme de fichiers d'archives compressées avec bzip2.

Un autre format populaire et portable est le format PDF , le plus pratique pour les impressions. Mais avant de lancer une telle impression, pensez bien que le manuel au complet faire plus de 2000 pages, et qu'il est constamment révisé.

| Note |
|---|
| Si vous n'avez pas de programme capable de lire les fichiers PDF , téléchargez le Array . |

Pour les propriétaires de PDA Palm, deux formats Palm document et iSilo sont optimisés pour cette plate−forme. Vous pouvez emporter votre PDA avec vous dans vos déplacements quotidiens, et utiliser les programmes ARRAY et Array pour améliorer vos connaissances de PHP , ou bien profiter de ses références rapides.

Pour les plates−formes Windows, le format Windows HTML Help fournit une version HTML du

manuel, avec l'application Windows HTML Help : il intègre un moteur de recherche complet, un index et des signets. De nombreux IDE sous Windows fournissent des liens avec ce format pour une meilleure intégration.

## 7.12.2 A propos des notes utilisateurs

Les notes des utilisateurs jouent un rôle très important dans le développement de ce manuel. En permettant aux lecteurs de contribuer par des exemples, des remarques et critiques, ou encore des clarifications, nous intégrons des aspects très importants du langage dans le manuel. Jusqu'à ce que les notes les plus importantes soient intégrées dans la documentation, elles sont disponibles sur le site lui même, et dans certains formats hors ligne.

| Note |
| --- |
| Les notes des utilisateurs ne sont pas modérées avant d'apparaître sur le site, et même si elles sont modérées, leur véracité ne peut être garantie, pas plus qu'il n'existe de garantie quand à l'exactitude du manuel lui−même. |

## 7.12.3 Comment lire la définition d'une fonction (prototype)

Chaque fonction est documentée pour permettre un accès rapide, et savoir décoder le manuel rendra votre lecture plus facile. Plutôt que de dépendre d'exemples prêts en copier/coller, il est mieux de savoir lire la définition d'une fonction. Voici comment :

| Note |
| --- |
| Pré−requis : Connaissances de bases des types.<br><br>Bien que PHP soit un langage sans typage fort, une connaissance de base des types est essentielle, car ils ont quand même des sens importants. |

Les définitions de fonctions vous indiquent quel type de données est retourné . Examinons la fonction strlen comme exemple :

strlen

PHP 3, PHP 4

int strlen ( string str )

strlen retourne la taille de la chaîne str .

| Partie | Description |
| --- | --- |
| strlen | Le nom de la fonction. |
| ( PHP 3, PHP 4) | strlen() est présente dans toutes les versions de PHP 3 et PHP 4. |
| int | Type de valeur retourné par cette fonction, qui est en l'occurrence un entier (i.e. la taille d'une chaîne est mesurée par un nombre). |
| ( string str ) | Le premier (et ici le seul) paramètre à fournir à strlen est le paramètre str , qui doit être du type chaîne de caractères. |

Nous pourrions réécrire ce prototype avec une version plus générique :

type_retourné nom_de_la fonction ( type_du_paramètre nom_du_paramètre )

Plusieurs fonctions ont besoin de plusieurs paramètres, comme par exemple in_array . Son prototype est le suivant :

bool in_array ( mixed needle, array haystack [, bool strict])

Qu'est ce que cela signifie? in_array retourne un booléen TRUE en cas de réussite ( le paramètre needle a été trouvé dans le tableau haystack ) ou FALSE en cas d'échec ( le paramètre needle n'a pas été trouvé dans le tableau haystack ). Le premier paramètre s'appelle needle et il peut être de différents types : il porte donc la mention **mixed** . Le paramètre needle (ce que nous recherchons) peut être une valeur scalaire ( chaîne de caractères , entier , ou nombre à virgule flottante ), ou encore un tableau . haystack (le tableau dans lequel nous recherchons) est le second paramètre. Le troisième paramètre, strict , est optionnel. Tous les paramètres optionnels sont placés entre crochets. La manuel indique que le paramètre strict vaut par défaut FALSE . Reportez−vous au manuel de chaque fonction pour savoir comment elle fonctionne.

## 7.12.4 Versions de PHP documentées dans ce manuel

Cette documentation contient des informations sur PHP 4, avec des notes compatibilités et de migration depuis PHP 3. Le comportement, les paramètres, les valeurs retournées, etc... qui ont changé entre deux versions du manuel PHP sont documentées dans les notes utilisateurs, et dans le manuel lui même.

Vous pouvez trouvez des portions de documentation destinées aux versions CVS de PHP , ce qui signifie que les toutes dernières versions de développement de PHP proposent ces fonctionnalités. Si vous n'êtes pas un développeur PHP , et que vous n'avez pas la possibilité d'utiliser la dernière version de PHP , les fonctionnalités indiquées comme "disponibles uniquement en CVS" ne vous seront pas accessibles. Toutefois, ces fonctionnalités seront surement disponible dans la prochaine version stable de PHP . Si vous voulez obtenir une version des derniers sources de PHP , passez par le serveur CVS anonyme .

Vous pouvez aussi rencontrer de la documentation qui est disponible pour une version de PHP qui n'est pas encore publiée (par exemple, des fonctions destinées à la version PHP 4.3.0, alors que la version publique actuelle est PHP 4.2.3). La plupart du temps, ce n'est pas une erreur de documentation, mais une indication sur les fonctionnalités qui seront disponible avec la prochaine publication de PHP .

## 7.12.5 Où trouver plus d'informations sur PHP ?

Ce manuel n'a pas pour objectif de fournir des présentations sur les pratiques de programmation. Si vous être un néophyte total, ou même un programmeur débutant, vous pouvez trouver difficile d'apprendre la programmation PHP avec uniquement ce manuel : il serait mieux de trouver des ressources plus orientée vers l'apprentissage. Vous pouvez aussi trouver une liste des livres traitant de PHP sur http://www.php.net/books.php .

Il y a un bon nombre de listes de diffusion actives, traitant de tous les aspects du langage et de la programmation pHP. Si vous êtes bloqué par un problème, vous pourrez surement trouver de l'aide auprès de ces listes. Il existe un recensement des listes de diffusion à l'URL http://www.php.net/support.php , ainsi qu'une liste d'archives, et de ressources de support en ligne. De plus, à http://www.php.net/links.php , il y a une liste de sites web dédiés à PHP , avec des articles, des forum, des bibliothèques de code.

## 7.12.6 Comment aider à l'amélioration de la documentation

Il y a trois façon de participer à l'amélioration de la documentation.

Si vous trouvez des erreurs, dans n'importe quelle traduction de la documentation utiliser le système de rapport de bugs : http://bugs.php.net/ . Classez le rapport sous la catégorie "Documentation Problem" . Vous pouvez aussi poster vos problèmes ici même.

| Note |
| --- |
| N'abusez pas du système de bugs pour envoyer des demandes d'aide. Utilisez les listes de diffusion ou les sites de communauté, tels que mentionnés ci−dessus. |

En contribuant aux notes, vous pouvez fournir de nouveaux exemples, mettre en lumière des effets de bords ou apporter des clarifications pour les autres lecteurs. Mais ne prenez pas le système d'annotation pour un système de soumission de bugs. Vous pouvez en savoir plus sur les annotations dans la section 'A propos des notes d'utilisateurs'

Si vous connaissez l'anglais et une autre langue, vous pouvez aussi contribuer en participant à la traduction. Si vous voulez démarrer une nouvelle traduction, ou participer au projet, alors commencez par lire http://cvs.php.net/co.php/phpdoc/howto/howto.html.tar.gz .

## 7.12.7 Comment sont générées les documentations

Ce manuel est écrit en XML , en utilisant la DTD de DocBook XML , et DSSSL ( Document Style and Semantics Specification Language ) pour le format, et aussi, expérimentalement, XSLT ( eXtensible Stylesheet Language Transformations ) pour les maintenances et les formats.

Grâce au format XML comme source, nous avons la possibilité de générer de nombreux formats tout en ayant une seule source pour tous les formats. Les outils utilisés pour formater les versions HTML et TEX sont Jade , écrit par James Clark et The Modular DocBook Stylesheets écrit par Norman Walsh . Nous utilisons Array pour générer le format Windows HTML Help format du manuel, et bien sur, PHP lui−même pour réaliser certaines conversions et formats.

Vous pouvez télécharger le manuel dans différentes langues, y compris un format en texte plein, HTML , PDF , PalmPilot DOC , PalmPilot iSilo et Windows HTML Help , à l'URL http://www.php.net/docs.php . Les manuels sont mis à jours automatiquement, avec la mise à jour des sources.

Vous pouvez trouver plus d'informations sur le téléchargement des sources XML de cette documentation à http://cvs.php.net/ . La documentation est stockée dans le module phpdoc .

## 7.12.8 Traductions

La manuel PHP est disponible non seulement en anglais, mais aussi dans différentes langues. Le texte du manuel est écrit d'abord en anglais, puis des équipes à travers le monde, assure la traduction du manuel dans leur langue natale. Si la traduction d'une section n'est pas encore disponible, le système de création de la documentation présentera alors la version anglaise.

Les contributeurs aux documentations partent des codes sources XML disponibles sur le CVS http://cvs.php.net/ puis traduisent dans leur langue. Ils **n'utilisent pas** le format HTML , le texte plein, ou encore le format PDF . C'est le système d'édition qui se charge de faire les conversions du format XML vers un format lisible.

| Note |
| --- |
| Si vous voulez démarrer une nouvelle e traduction dans votre langue, entrez en contact avec l'équipe de documentation en vous inscrivant à la liste de diffusion : phpdoc−subscribe@lists.php.net . L'adresse de la liste de diffusion est phpdoc@lists.php.net . Indiquez dans le message que vous êtes intéressé pour traduire la documentation dans une nouvelle langue, et quelqu'un viendra vous aider à démarrer un nouveau site de traduction, ou rejoindre l'équipe qui a pris en charge cette traduction. |

Actuellement, le manuel est disponible, partiellement ou en totalité, dans les langues suivantes : Allemand, Chinois (simplifié), Chinois (traditionnel), Coréen, Espagnol, Finnois, Français, Hébreu, Hongrois, Italien, Japonais, Néerlandais, Polonais, Portugais brésilien, Roumain, Russe, Slovaque, Slovène, Suédois, Tchèque et Turc.

Ils peuvent tous être téléchargés ici : http://www.php.net/docs.php .

## 7.12.9 Traduction française

La version française est traduite quotidiennement et disponible chez Nexen ( nexen.net/ ). Ce manuel a été généré à partir de la documentation originale en anglais du PHP Documentation Group, au format XML. Elle est 100% compatible avec le système de génération du groupe PHP . Elle est traité avec un système développé par Nexen pour produire divers autres formats.

Pour les problèmes liés à la traduction, vous pouvez contacter directement les éditeurs de ce manuel damien.seguy@nexen.net et didou@keliglia.com .

# 8 Index

## 8.1 Index des fonctions

- **A**
- 
  - abs
  - acos
  - acosh
  - addcslashes
  - addslashes
  - apache_child_terminate
  - apache_get_version
  - apache_lookup_uri
  - apache_note
  - apache_request_headers
  - apache_response_headers
  - apache_setenv
  - array
  - array_change_key_case
  - array_chunk
  - array_combine
  - array_count_values
  - array_diff
  - array_diff_assoc
  - array_diff_uassoc
  - array_fill
  - array_filter
  - array_flip
  - array_intersect
  - array_intersect_assoc
  - array_key_exists
  - array_keys
  - array_map
  - array_merge
  - array_merge_recursive
  - array_multisort
  - array_pad
  - array_pop
  - array_push
  - array_rand
  - array_reduce
  - array_reverse
  - array_search
  - array_shift
  - array_slice
  - array_splice
  - array_sum
  - array_udiff
  - array_udiff_assoc
  - array_udiff_uassoc
  - array_unique
  - array_unshift

- array_values
- array_walk
- arsort
- ascii2ebcdic
- asin
- asinh
- asort
- assert
- assert_options
- atan
- atan2
- atanh

**B**

- base64_decode
- base64_encode
- base_convert
- basename
- bin2hex
- bindec

**C**

- cal_days_in_month
- cal_from_jd
- cal_info
- cal_to_jd
- call_user_func
- call_user_func_array
- call_user_method
- call_user_method_array
- ceil
- chdir
- checkdate
- checkdnsrr
- chgrp
- chmod
- chop
- chown
- chr
- chroot
- chunk_split
- class_exists
- clearstatcache
- closedir
- closelog
- compact
- connection_aborted
- connection_status
- connection_timeout
- constant
- convert_cyr_string
- copy
- cos
- cosh
- count
- count_chars

- crc32
- create_function
- crypt
- curl_close
- curl_errno
- curl_error
- curl_exec
- curl_getinfo
- curl_init
- curl_multi_add_handle
- curl_multi_close
- curl_multi_exec
- curl_multi_getcontent
- curl_multi_info_read
- curl_multi_init
- curl_multi_remove_handle
- curl_multi_select
- curl_setopt
- curl_version
- current

**D**

- date
- dbase_add_record
- dbase_close
- dbase_create
- dbase_delete_record
- dbase_get_header_info
- dbase_get_record
- dbase_get_record_with_names
- dbase_numfields
- dbase_numrecords
- dbase_open
- dbase_pack
- dbase_replace_record
- debug_backtrace
- debug_print_backtrace
- debugger_off
- debugger_on
- decbin
- dechex
- decoct
- define
- define_syslog_variables
- defined
- deg2rad
- delete
- die
- dir
- dirname
- disk_free_space
- disk_total_space
- diskfreespace
- dl
- dns_check_record

- dns_get_mx
- dns_get_record
- doubleval

- **E**
- each
- easter_date
- easter_days
- ebcdic2ascii
- echo
- empty
- end
- error_log
- error_reporting
- escapeshellarg
- escapeshellcmd
- eval
- exec
- exif_imagetype
- exif_read_data
- exif_thumbnail
- exit
- exp
- explode
- expm1
- extension_loaded
- extract
- ezmlm_hash

- **F**
- fclose
- feof
- fflush
- fgetc
- fgetcsv
- fgets
- fgetss
- file
- file_exists
- file_get_contents
- file_put_contents
- fileatime
- filectime
- filegroup
- fileinode
- filemtime
- fileowner
- fileperms
- filesize
- filetype
- floatval
- flock
- floor
- flush
- fmod
- fnmatch

- fopen
- fpassthru
- fprintf
- fputs
- fread
- frenchtojd
- fscanf
- fseek
- fsockopen
- fstat
- ftell
- ftp_alloc
- ftp_cdup
- ftp_chdir
- ftp_chmod
- ftp_close
- ftp_connect
- ftp_delete
- ftp_exec
- ftp_fget
- ftp_fput
- ftp_get
- ftp_get_option
- ftp_login
- ftp_mdtm
- ftp_mkdir
- ftp_nb_continue
- ftp_nb_fget
- ftp_nb_fput
- ftp_nb_get
- ftp_nb_put
- ftp_nlist
- ftp_pasv
- ftp_put
- ftp_pwd
- ftp_quit
- ftp_raw
- ftp_rawlist
- ftp_rename
- ftp_rmdir
- ftp_set_option
- ftp_site
- ftp_size
- ftp_ssl_connect
- ftp_systype
- ftruncate
- func_get_arg
- func_get_args
- func_num_args
- function_exists
- fwrite

**G**

- gd_info
- get_browser

- get_cfg_var
- get_class
- get_class_methods
- get_class_vars
- get_current_user
- get_declared_classes
- get_defined_constants
- get_defined_functions
- get_defined_vars
- get_extension_funcs
- get_html_translation_table
- get_include_path
- get_included_files
- get_loaded_extensions
- get_magic_quotes_gpc
- get_magic_quotes_runtime
- get_meta_tags
- get_object_vars
- get_parent_class
- get_required_files
- get_resource_type
- getallheaders
- getcwd
- getdate
- getenv
- gethostbyaddr
- gethostbyname
- gethostbynamel
- getimagesize
- getlastmod
- getmxrr
- getmygid
- getmyinode
- getmypid
- getmyuid
- getopt
- getprotobyname
- getprotobynumber
- getrandmax
- getrusage
- getservbyname
- getservbyport
- gettimeofday
- gettype
- glob
- gmdate
- gmmktime
- gmstrftime
- gregoriantojd
- gzclose
- gzcompress
- gzdeflate
- gzencode
- gzeof

- gzfile
- gzgetc
- gzgets
- gzgetss
- gzinflate
- gzopen
- gzpassthru
- gzputs
- gzread
- gzrewind
- gzseek
- gztell
- gzuncompress
- gzwrite

- **H**
- header
- headers_list
- headers_sent
- hebrev
- hebrevc
- hexdec
- highlight_file
- highlight_string
- html_entity_decode
- htmlentities
- htmlspecialchars
- http_build_query
- hypot

- **I**
- ignore_user_abort
- image2wbmp
- image_type_to_mime_type
- imagealphablending
- imageantialias
- imagearc
- imagechar
- imagecharup
- imagecolorallocate
- imagecolorallocatealpha
- imagecolorat
- imagecolorclosest
- imagecolorclosestalpha
- imagecolorclosesthwb
- imagecolordeallocate
- imagecolorexact
- imagecolorexactalpha
- imagecolormatch
- imagecolorresolve
- imagecolorresolvealpha
- imagecolorset
- imagecolorsforindex
- imagecolorstotal
- imagecolortransparent
- imagecopy

- imagecopymerge
- imagecopymergegray
- imagecopyresampled
- imagecopyresized
- imagecreate
- imagecreatefromgd
- imagecreatefromgd2
- imagecreatefromgd2part
- imagecreatefromgif
- imagecreatefromjpeg
- imagecreatefrompng
- imagecreatefromstring
- imagecreatefromwbmp
- imagecreatefromxbm
- imagecreatefromxpm
- imagecreatetruecolor
- imagedashedline
- imagedestroy
- imageellipse
- imagefill
- imagefilledarc
- imagefilledellipse
- imagefilledpolygon
- imagefilledrectangle
- imagefilltoborder
- imagefontheight
- imagefontwidth
- imageftbbox
- imagefttext
- imagegammacorrect
- imagegd
- imagegd2
- imagegif
- imageinterlace
- imageistruecolor
- imagejpeg
- imageline
- imageloadfont
- imagepalettecopy
- imagepng
- imagepolygon
- imagepsbbox
- imagepscopyfont
- imagepsencodefont
- imagepsextendfont
- imagepsfreefont
- imagepsloadfont
- imagepsslantfont
- imagepstext
- imagerectangle
- imagerotate
- imagesavealpha
- imagesetbrush
- imagesetpixel

- imagesetstyle
- imagesetthickness
- imagesettile
- imagestring
- imagestringup
- imagesx
- imagesy
- imagetruecolortopalette
- imagettfbbox
- imagettftext
- imagetypes
- imagewbmp
- imap_8bit
- imap_alerts
- imap_append
- imap_base64
- imap_binary
- imap_body
- imap_bodystruct
- imap_check
- imap_clearflag_full
- imap_close
- imap_createmailbox
- imap_delete
- imap_deletemailbox
- imap_errors
- imap_expunge
- imap_fetch_overview
- imap_fetchbody
- imap_fetchheader
- imap_fetchstructure
- imap_get_quota
- imap_get_quotaroot
- imap_getacl
- imap_getmailboxes
- imap_getsubscribed
- imap_header
- imap_headerinfo
- imap_headers
- imap_last_error
- imap_list
- imap_listmailbox
- imap_listscan
- imap_listsubscribed
- imap_lsub
- imap_mail
- imap_mail_compose
- imap_mail_copy
- imap_mail_move
- imap_mailboxmsginfo
- imap_mime_header_decode
- imap_msgno
- imap_num_msg
- imap_num_recent

- imap_open
- imap_ping
- imap_qprint
- imap_renamemailbox
- imap_reopen
- imap_rfc822_parse_adrlist
- imap_rfc822_parse_headers
- imap_rfc822_write_address
- imap_scanmailbox
- imap_search
- imap_set_quota
- imap_setacl
- imap_setflag_full
- imap_sort
- imap_status
- imap_subscribe
- imap_thread
- imap_timeout
- imap_uid
- imap_undelete
- imap_unsubscribe
- imap_utf7_decode
- imap_utf7_encode
- imap_utf8
- implode
- import_request_variables
- in_array
- ini_alter
- ini_get
- ini_get_all
- ini_restore
- ini_set
- intval
- ip2long
- iptcembed
- iptcparse
- is_a
- is_array
- is_bool
- is_callable
- is_dir
- is_double
- is_executable
- is_file
- is_finite
- is_float
- is_infinite
- is_int
- is_integer
- is_link
- is_long
- is_nan
- is_null
- is_numeric

- is_object
- is_readable
- is_real
- is_resource
- is_scalar
- is_string
- is_subclass_of
- is_uploaded_file
- is_writable
- is_writeable
- isset

- **J**
- jddayofweek
- jdmonthname
- jdtofrench
- jdtogregorian
- jdtojewish
- jdtojulian
- jdtounix
- jewishtojd
- join
- jpeg2wbmp
- juliantojd

- **K**
- key
- krsort
- ksort

- **L**
- lcg_value
- ldap_8859_to_t61
- ldap_add
- ldap_bind
- ldap_close
- ldap_compare
- ldap_connect
- ldap_count_entries
- ldap_delete
- ldap_dn2ufn
- ldap_err2str
- ldap_errno
- ldap_error
- ldap_explode_dn
- ldap_first_attribute
- ldap_first_entry
- ldap_first_reference
- ldap_free_result
- ldap_get_attributes
- ldap_get_dn
- ldap_get_entries
- ldap_get_option
- ldap_get_values
- ldap_get_values_len
- ldap_list
- ldap_mod_add

- ldap_mod_del
- ldap_mod_replace
- ldap_modify
- ldap_next_attribute
- ldap_next_entry
- ldap_next_reference
- ldap_parse_reference
- ldap_parse_result
- ldap_read
- ldap_rename
- ldap_search
- ldap_set_option
- ldap_set_rebind_proc
- ldap_sort
- ldap_start_tls
- ldap_t61_to_8859
- ldap_unbind
- levenshtein
- link
- linkinfo
- list
- localeconv
- localtime
- log
- log10
- log1p
- long2ip
- lstat
- ltrim

- **M**
- 
  - mail
  - main
  - max
  - mb_convert_case
  - mb_convert_encoding
  - mb_convert_kana
  - mb_convert_variables
  - mb_decode_mimeheader
  - mb_decode_numericentity
  - mb_detect_encoding
  - mb_detect_order
  - mb_encode_mimeheader
  - mb_encode_numericentity
  - mb_ereg
  - mb_ereg_match
  - mb_ereg_replace
  - mb_ereg_search
  - mb_ereg_search_getpos
  - mb_ereg_search_getregs
  - mb_ereg_search_init
  - mb_ereg_search_pos
  - mb_ereg_search_regs
  - mb_ereg_search_setpos
  - mb_eregi

- mb_eregi_replace
- mb_get_info
- mb_http_input
- mb_http_output
- mb_internal_encoding
- mb_language
- mb_output_handler
- mb_parse_str
- mb_preferred_mime_name
- mb_regex_encoding
- mb_regex_set_options
- mb_send_mail
- mb_split
- mb_strcut
- mb_strimwidth
- mb_strlen
- mb_strpos
- mb_strrpos
- mb_strtolower
- mb_strtoupper
- mb_strwidth
- mb_substitute_character
- mb_substr
- mb_substr_count
- mcrypt_cbc
- mcrypt_cfb
- mcrypt_create_iv
- mcrypt_decrypt
- mcrypt_ecb
- mcrypt_enc_get_algorithms_name
- mcrypt_enc_get_block_size
- mcrypt_enc_get_iv_size
- mcrypt_enc_get_key_size
- mcrypt_enc_get_modes_name
- mcrypt_enc_get_supported_key_sizes
- mcrypt_enc_is_block_algorithm
- mcrypt_enc_is_block_algorithm_mode
- mcrypt_enc_is_block_mode
- mcrypt_enc_self_test
- mcrypt_encrypt
- mcrypt_generic
- mcrypt_generic_deinit
- mcrypt_generic_end
- mcrypt_generic_init
- mcrypt_get_block_size
- mcrypt_get_cipher_name
- mcrypt_get_iv_size
- mcrypt_get_key_size
- mcrypt_list_algorithms
- mcrypt_list_modes
- mcrypt_module_close
- mcrypt_module_get_algo_block_size
- mcrypt_module_get_algo_key_size
- mcrypt_module_get_supported_key_sizes

- mcrypt_module_is_block_algorithm
- mcrypt_module_is_block_algorithm_mode
- mcrypt_module_is_block_mode
- mcrypt_module_open
- mcrypt_module_self_test
- mcrypt_ofb
- md5
- md5_file
- mdecrypt_generic
- memory_get_usage
- metaphone
- method_exists
- microtime
- min
- ming_setcubicthreshold
- ming_setscale
- ming_useswfversion
- mkdir
- mktime
- money_format
- move_uploaded_file
- mt_getrandmax
- mt_rand
- mt_srand
- mysql_affected_rows
- mysql_change_user
- mysql_client_encoding
- mysql_close
- mysql_connect
- mysql_create_db
- mysql_data_seek
- mysql_db_name
- mysql_db_query
- mysql_drop_db
- mysql_errno
- mysql_error
- mysql_escape_string
- mysql_fetch_array
- mysql_fetch_assoc
- mysql_fetch_field
- mysql_fetch_lengths
- mysql_fetch_object
- mysql_fetch_row
- mysql_field_flags
- mysql_field_len
- mysql_field_name
- mysql_field_seek
- mysql_field_table
- mysql_field_type
- mysql_free_result
- mysql_get_client_info
- mysql_get_host_info
- mysql_get_proto_info
- mysql_get_server_info

- mysql_info
- mysql_insert_id
- mysql_list_dbs
- mysql_list_fields
- mysql_list_processes
- mysql_list_tables
- mysql_num_fields
- mysql_num_rows
- mysql_pconnect
- mysql_ping
- mysql_query
- mysql_real_escape_string
- mysql_result
- mysql_select_db
- mysql_stat
- mysql_tablename
- mysql_thread_id
- mysql_unbuffered_query

**N**

- natcasesort
- natsort
- next
- nl2br
- nl_langinfo
- number_format

**O**

- ob_clean
- ob_end_clean
- ob_end_flush
- ob_flush
- ob_get_clean
- ob_get_contents
- ob_get_flush
- ob_get_length
- ob_get_level
- ob_get_status
- ob_gzhandler
- ob_implicit_flush
- ob_list_handlers
- ob_start
- ocibindbyname
- ocicancel
- ocicloselob
- ocicollappend
- ocicollassign
- ocicollassignelem
- ocicollgetelem
- ocicollmax
- ocicollsize
- ocicolltrim
- ocicolumname
- ocicolumnisnull
- ocicolumnprecision
- ocicolumnscale

- ocicolumnsize
- ocicolumntype
- ocicolumntyperaw
- ocicommit
- ocidefinebyname
- ocierror
- ociexecute
- ocifetch
- ocifetchinto
- ocifetchstatement
- ocifreecollection
- ocifreecursor
- ocifreedesc
- ocifreestatement
- ociinternaldebug
- ociloadlob
- ocilogoff
- ocilogon
- ocinewcollection
- ocinewcursor
- ocinewdescriptor
- ocinlogon
- ocinumcols
- ociparse
- ociplogon
- ociresult
- ocirollback
- ocirowcount
- ocisavelob
- ocisavelobfile
- ociserverversion
- ocisetprefetch
- ocistatementtype
- ociwritelobtofile
- ociwritetemporarylob
- octdec
- odbc_autocommit
- odbc_binmode
- odbc_close
- odbc_close_all
- odbc_columnprivileges
- odbc_columns
- odbc_commit
- odbc_connect
- odbc_cursor
- odbc_data_source
- odbc_do
- odbc_error
- odbc_errormsg
- odbc_exec
- odbc_execute
- odbc_fetch_array
- odbc_fetch_into
- odbc_fetch_object

- odbc_fetch_row
- odbc_field_len
- odbc_field_name
- odbc_field_num
- odbc_field_precision
- odbc_field_scale
- odbc_field_type
- odbc_foreignkeys
- odbc_free_result
- odbc_gettypeinfo
- odbc_longreadlen
- odbc_next_result
- odbc_num_fields
- odbc_num_rows
- odbc_pconnect
- odbc_prepare
- odbc_primarykeys
- odbc_procedurecolumns
- odbc_procedures
- odbc_result
- odbc_result_all
- odbc_rollback
- odbc_setoption
- odbc_specialcolumns
- odbc_statistics
- odbc_tableprivileges
- odbc_tables
- opendir
- openlog
- ord
- output_add_rewrite_var
- output_reset_rewrite_vars

**P**

- pack
- parse_ini_file
- parse_str
- parse_url
- passthru
- pathinfo
- pclose
- pdf_add_annotation
- pdf_add_bookmark
- pdf_add_launchlink
- pdf_add_locallink
- pdf_add_note
- pdf_add_outline
- pdf_add_pdflink
- pdf_add_thumbnail
- pdf_add_weblink
- pdf_arc
- pdf_arcn
- pdf_attach_file
- pdf_begin_page
- pdf_begin_pattern

- pdf_begin_template
- pdf_circle
- pdf_clip
- pdf_close
- pdf_close_image
- pdf_close_pdi
- pdf_close_pdi_page
- pdf_closepath
- pdf_closepath_fill_stroke
- pdf_closepath_stroke
- pdf_concat
- pdf_continue_text
- pdf_curveto
- pdf_delete
- pdf_end_page
- pdf_end_pattern
- pdf_end_template
- pdf_endpath
- pdf_fill
- pdf_fill_stroke
- pdf_findfont
- pdf_get_buffer
- pdf_get_font
- pdf_get_fontname
- pdf_get_fontsize
- pdf_get_image_height
- pdf_get_image_width
- pdf_get_majorversion
- pdf_get_minorversion
- pdf_get_parameter
- pdf_get_pdi_parameter
- pdf_get_pdi_value
- pdf_get_value
- pdf_initgraphics
- pdf_lineto
- pdf_makespotcolor
- pdf_moveto
- pdf_new
- pdf_open
- pdf_open_ccitt
- pdf_open_file
- pdf_open_gif
- pdf_open_image
- pdf_open_image_file
- pdf_open_jpeg
- pdf_open_memory_image
- pdf_open_pdi
- pdf_open_pdi_page
- pdf_open_png
- pdf_open_tiff
- pdf_place_image
- pdf_place_pdi_page
- pdf_rect
- pdf_restore

- pdf_rotate
- pdf_save
- pdf_scale
- pdf_set_border_color
- pdf_set_border_dash
- pdf_set_border_style
- pdf_set_char_spacing
- pdf_set_duration
- pdf_set_font
- pdf_set_horiz_scaling
- pdf_set_info
- pdf_set_info_author
- pdf_set_info_creator
- pdf_set_info_keywords
- pdf_set_info_subject
- pdf_set_info_title
- pdf_set_leading
- pdf_set_parameter
- pdf_set_text_matrix
- pdf_set_text_pos
- pdf_set_text_rendering
- pdf_set_text_rise
- pdf_set_value
- pdf_set_word_spacing
- pdf_setcolor
- pdf_setdash
- pdf_setflat
- pdf_setfont
- pdf_setgray
- pdf_setgray_fill
- pdf_setgray_stroke
- pdf_setlinecap
- pdf_setlinejoin
- pdf_setlinewidth
- pdf_setmatrix
- pdf_setmiterlimit
- pdf_setpolydash
- pdf_setrgbcolor
- pdf_setrgbcolor_fill
- pdf_setrgbcolor_stroke
- pdf_show
- pdf_show_boxed
- pdf_show_xy
- pdf_skew
- pdf_stringwidth
- pdf_stroke
- pdf_translate
- pfsockopen
- pg_affected_rows
- pg_cancel_query
- pg_client_encoding
- pg_close
- pg_connect
- pg_connection_busy

- pg_connection_reset
- pg_connection_status
- pg_convert
- pg_copy_from
- pg_copy_to
- pg_dbname
- pg_delete
- pg_end_copy
- pg_escape_bytea
- pg_escape_string
- pg_fetch_all
- pg_fetch_array
- pg_fetch_assoc
- pg_fetch_object
- pg_fetch_result
- pg_fetch_row
- pg_field_is_null
- pg_field_name
- pg_field_num
- pg_field_prtlen
- pg_field_size
- pg_field_type
- pg_free_result
- pg_get_notify
- pg_get_pid
- pg_get_result
- pg_host
- pg_insert
- pg_last_error
- pg_last_notice
- pg_last_oid
- pg_lo_close
- pg_lo_create
- pg_lo_export
- pg_lo_import
- pg_lo_open
- pg_lo_read
- pg_lo_read_all
- pg_lo_seek
- pg_lo_tell
- pg_lo_unlink
- pg_lo_write
- pg_meta_data
- pg_num_fields
- pg_num_rows
- pg_options
- pg_pconnect
- pg_ping
- pg_port
- pg_put_line
- pg_query
- pg_result_error
- pg_result_seek
- pg_result_status

- pg_select
- pg_send_query
- pg_set_client_encoding
- pg_trace
- pg_tty
- pg_unescape_bytea
- pg_untrace
- pg_update
- php_ini_scanned_files
- php_logo_guid
- php_sapi_name
- php_uname
- phpcredits
- phpinfo
- phpversion
- pi
- png2wbmp
- popen
- pos
- pow
- preg_grep
- preg_match
- preg_match_all
- preg_quote
- preg_replace
- preg_replace_callback
- preg_split
- prev
- print
- print_r
- printf
- proc_close
- proc_get_status
- proc_nice
- proc_open
- proc_terminate
- putenv

**Q**

- quoted_printable_decode
- quotemeta

**R**

- rad2deg
- rand
- range
- rawurldecode
- rawurlencode
- read_exif_data
- readdir
- readfile
- readgzfile
- readlink
- realpath
- register_shutdown_function
- register_tick_function

- rename
- reset
- restore_error_handler
- restore_include_path
- rewind
- rewinddir
- rmdir
- round
- rsort
- rtrim

- **S**
- 
  - scandir
  - serialize
  - session_cache_expire
  - session_cache_limiter
  - session_decode
  - session_destroy
  - session_encode
  - session_get_cookie_params
  - session_id
  - session_is_registered
  - session_module_name
  - session_name
  - session_regenerate_id
  - session_register
  - session_save_path
  - session_set_cookie_params
  - session_set_save_handler
  - session_start
  - session_unregister
  - session_unset
  - session_write_close
  - set_error_handler
  - set_file_buffer
  - set_include_path
  - set_magic_quotes_runtime
  - set_time_limit
  - setcookie
  - setlocale
  - settype
  - sha1
  - sha1_file
  - shell_exec
  - show_source
  - shuffle
  - similar_text
  - sin
  - sinh
  - sizeof
  - sleep
  - socket_accept
  - socket_bind
  - socket_clear_error
  - socket_close

- socket_connect
- socket_create
- socket_create_listen
- socket_create_pair
- socket_get_option
- socket_get_status
- socket_getpeername
- socket_getsockname
- socket_iovec_add
- socket_iovec_alloc
- socket_iovec_delete
- socket_iovec_fetch
- socket_iovec_free
- socket_iovec_set
- socket_last_error
- socket_listen
- socket_read
- socket_readv
- socket_recv
- socket_recvfrom
- socket_recvmsg
- socket_select
- socket_send
- socket_sendmsg
- socket_sendto
- socket_set_block
- socket_set_blocking
- socket_set_nonblock
- socket_set_option
- socket_set_timeout
- socket_shutdown
- socket_strerror
- socket_write
- socket_writev
- sort
- soundex
- sprintf
- sqrt
- srand
- sscanf
- stat
- str_ireplace
- str_pad
- str_repeat
- str_replace
- str_rot13
- str_shuffle
- str_split
- str_word_count
- strcasecmp
- strchr
- strcmp
- strcoll
- strcspn

- strftime
- strip_tags
- stripcslashes
- stripos
- stripslashes
- stristr
- strlen
- strnatcasecmp
- strnatcmp
- strncasecmp
- strncmp
- strpos
- strrchr
- strrev
- strripos
- strrpos
- strspn
- strstr
- strtok
- strtolower
- strtotime
- strtoupper
- strtr
- strval
- substr
- substr_count
- substr_replace
- swfaction
- swfbitmap
- swfbitmap−>getheight
- swfbitmap−>getwidth
- swfbutton
- swfbutton−>addaction
- swfbutton−>addshape
- swfbutton−>setaction
- swfbutton−>setdown
- swfbutton−>sethit
- swfbutton−>setover
- swfbutton−>setup
- swfbutton_keypress
- swfdisplayitem
- swfdisplayitem−>addcolor
- swfdisplayitem−>move
- swfdisplayitem−>moveto
- swfdisplayitem−>multcolor
- swfdisplayitem−>remove
- swfdisplayitem−>rotate
- swfdisplayitem−>rotateto
- swfdisplayitem−>scale
- swfdisplayitem−>scaleto
- swfdisplayitem−>setdepth
- swfdisplayitem−>setname
- swfdisplayitem−>setratio
- swfdisplayitem−>skewx

- swfdisplayitem−>skewxto
- swfdisplayitem−>skewy
- swfdisplayitem−>skewyto
- swffill
- swffill−>moveto
- swffill−>rotateto
- swffill−>scaleto
- swffill−>skewxto
- swffill−>skewyto
- swffont
- swffont−>getwidth
- swfgradient
- swfgradient−>addentry
- swfmorph
- swfmorph−>getshape1
- swfmorph−>getshape2
- swfmovie
- swfmovie−>add
- swfmovie−>nextframe
- swfmovie−>output
- swfmovie−>remove
- swfmovie−>save
- swfmovie−>setbackground
- swfmovie−>setdimension
- swfmovie−>setframes
- swfmovie−>setrate
- swfmovie−>streammp3
- swfshape
- swfshape−>addfill
- swfshape−>drawcurve
- swfshape−>drawcurveto
- swfshape−>drawline
- swfshape−>drawlineto
- swfshape−>movepen
- swfshape−>movepento
- swfshape−>setleftfill
- swfshape−>setline
- swfshape−>setrightfill
- swfsprite
- swfsprite−>add
- swfsprite−>nextframe
- swfsprite−>remove
- swfsprite−>setframes
- swftext
- swftext−>addstring
- swftext−>getwidth
- swftext−>moveto
- swftext−>setcolor
- swftext−>setfont
- swftext−>setheight
- swftext−>setspacing
- swftextfield
- swftextfield−>addstring
- swftextfield−>align

- swftextfield->setbounds
- swftextfield->setcolor
- swftextfield->setfont
- swftextfield->setheight
- swftextfield->setindentation
- swftextfield->setleftmargin
- swftextfield->setlinespacing
- swftextfield->setmargins
- swftextfield->setname
- swftextfield->setrightmargin
- symlink
- syslog
- system

- **T**
- 
  - tan
  - tanh
  - tempnam
  - time
  - tmpfile
  - touch
  - trigger_error
  - trim
- **U**
- 
  - uasort
  - ucfirst
  - ucwords
  - udm_add_search_limit
  - udm_alloc_agent
  - udm_api_version
  - udm_cat_list
  - udm_cat_path
  - udm_check_charset
  - udm_check_stored
  - udm_clear_search_limits
  - udm_close_stored
  - udm_crc32
  - udm_errno
  - udm_error
  - udm_find
  - udm_free_agent
  - udm_free_ispell_data
  - udm_free_res
  - udm_get_doc_count
  - udm_get_res_field
  - udm_get_res_param
  - udm_load_ispell_data
  - udm_open_stored
  - udm_set_agent_param
  - uksort
  - umask
  - uniqid
  - unixtojd
  - unlink
  - unpack

- unregister_tick_function
- unserialize
- unset
- urldecode
- urlencode
- user_error
- usleep
- usort
- utf8_decode
- utf8_encode

- **V**
- var_dump
- var_export
- version_compare
- virtual
- vprintf
- vsprintf

- **W**
- wddx_add_vars
- wddx_deserialize
- wddx_packet_end
- wddx_packet_start
- wddx_serialize_value
- wddx_serialize_vars
- wordwrap

- **X**
- xml_error_string
- xml_get_current_byte_index
- xml_get_current_column_number
- xml_get_current_line_number
- xml_get_error_code
- xml_parse
- xml_parse_into_struct
- xml_parser_create
- xml_parser_create_ns
- xml_parser_free
- xml_parser_get_option
- xml_parser_set_option
- xml_set_character_data_handler
- xml_set_default_handler
- xml_set_element_handler
- xml_set_end_namespace_decl_handler
- xml_set_external_entity_ref_handler
- xml_set_notation_decl_handler
- xml_set_object
- xml_set_processing_instruction_handler
- xml_set_start_namespace_decl_handler
- xml_set_unparsed_entity_decl_handler

- **Z**
- zend_logo_guid
- zend_version
- zlib_get_coding_type

# 8.2 Index des concepts

- **A**
-
  - Accepte une connexion sur une socket
  - Active l'interruption de script sur déconnexion du visiteur
  - Active le debugger interne de php
  - Active le mode auto−validation
  - Active le suivi d'une connexion postgresql
  - Active ou désactive l'affichage des données de déboggage oracle
  - Active ou désactive l'entrelacement
  - Active ou désactive le mode passif
  - Active/désactive l'envoi implicite
  - Active/désactive l'option magic_quotes_runtime.
  - Active/désactive le mode bloquant d'une socket
  - Affecte le gestionnaire xml par défaut
  - Affecte les gestionnaires d'entités non déclarés
  - Affecte les gestionnaires d'instructions exécutables.
  - Affecte les gestionnaires de début et de fin de balise xml
  - Affecte les gestionnaires de texte littéral
  - Affecte les options d'un analyseur xml
  - Affecte un type à une variable
  - Affichage des informations d'en−tête d'une base de données dbase.
  - Affiche de nombreuses informations sur le php
  - Affiche des informations lisibles pour une variable
  - Affiche du texte à la prochaine ligne pdf
  - Affiche le contenu d'un fichier en une seule fois
  - Affiche le reste du fichier
  - Affiche le résultat sous la forme d'une table html.
  - Affiche le texte dans un cadre
  - Affiche le texte à la position courante
  - Affiche les informations d'une variable.
  - Affiche ou affecte le paramètre "apache request notes"
  - Affiche un fichier
  - Affiche un text à une position donnée
  - Affiche une backtrace
  - Affiche une chaîne de caractères
  - Affiche une chaîne de caractères
  - Affiche une chaîne de caractères formattée
  - Affiche une chaîne formatée
  - Affiche une image wbmp
  - Ajoute au texte
  - Ajoute des anti−slashs dans une chaîne
  - Ajoute des slashes dans une chaîne, à la mode du langage c
  - Ajoute différentes limitations de recherche
  - Ajoute du texte
  - Ajoute un attribut à l'entrée courante
  - Ajoute un enregistrement dans une base dbase
  - Ajoute un fichier attaché à la page pdf
  - Ajoute un lien web sur la page pdf courante
  - Ajoute un nouveau vecteur dans le tableau
  - Ajoute un objet dans une animation
  - Ajoute un objet à un sprite
  - Ajoute un objet à une collection oracle
  - Ajoute un remplissage plein à la forme flash

- Ajoute un signet dans la page courante
- Ajoute une action au bouton
- Ajoute une annotation dans la page pdf courante
- Ajoute une annotation de lancement dans la page pdf courante
- Ajoute une annotation de lien dans la page pdf courante
- Ajoute une annotation sur un lien vers un fichier dans la page pdf courante
- Ajoute une chaîne dans une boîte aux lettres.
- Ajoute une couleur à la liste du gradient
- Ajoute une couleur à une transoformation
- Ajoute une entrée dans un dossier ldap
- Ajoute une forme à un bouton
- Ajoute une miniature sur la page pdf courante
- Ajoute une ressource curl à un curl multiple
- Ajoute une règle de réécriture d'url
- Ajouter des variables à un paquet wddx
- Alias de count
- Alias de disk_free_space
- Alias de exif_read_data
- Alias de floatval
- Alias de get_included_files
- Alias de ini_set
- Alias de is_float
- Alias de is_int
- Alias de odbc_field_len
- Alias de rtrim
- Alias de stream_set_timeout
- Alias de stream_set_write_buffer
- Alias de trigger_error
- Alias de addshape(shape, swfbutton_down)
- Alias de addshape(shape, swfbutton_hit)
- Alias de addshape(shape, swfbutton_over)
- Alias de la fonction exit
- Alias de swfbutton−>addshape(shape, swfbutton_up)
- Alloue de l'espace pour un téléchargement de fichier
- Alloue une couleur pour une image
- Alloue une couleur à une image
- Alloue une session mnogosearch
- Analyse les données http get/post/cookie et assigne les variables globales
- Analyse un bloc binaire iptc http://www.iptc.org/ et recherche les balises simples.
- Analyse un en−tête mail
- Analyse un fichier en fonction d'un format
- Analyse une adresse email
- Analyse une chaîne à l'aide d'un format
- Analyse une requête http
- Analyse une requête sql avec oracle
- Analyse une structure xml
- Analyse une url et retourne ses composants
- Annule la fonction exécutée à chaque tick
- Annule la réécriture d'url
- Annule les transactions oracle en cours
- Annule toutes les limitates de recherche mnogosearch
- Annule une requête asynchrone
- Annule une transaction
- Appelle une fonction utilisateur

- Appelle une fonction utilisateur avec les paramètres rassemblés en tableau
- Appelle une méthode utilisateur avec un tableau de paramètres
- Appelle une méthode utilisateur d'un objet
- Applique la syntaxe colorisée à du code php
- Applique sur fonction sur des tableaux
- Applique une correction gamma à l'image
- Arc cosinus
- Arc cosinus hyperbolique
- Arc sinus
- Arc sinus hyperbolique
- Arc tangeant hyperbolique
- Arc tangent
- Arc tangent de deux variables
- Arrondi
- Arrondit au nombre supérieur
- Arrondit à l'entier inférieur
- Assigne l'action du bouton
- Assigne l'élément val à une collection oracle à l'index ndx
- Assigne une collection depuis une autre collection oracle
- Attend une connexion (listen) sur une socket
- Autentification au serveur ldap
- Avance le pointeur interne d'un tableau

**C**

- Calcule des statistiques sur une table
- Calcule l'intersection de deux tableaux avec des tests sur les index
- Calcule l'intersection de tableaux
- Calcule la clé métaphone
- Calcule la clé soundex
- Calcule la différence de deux tableaux associatifs, avec une fonction utilisateur
- Calcule la différence de deux tableaux, en prenant en compte les clés
- Calcule la différence entre des tableaux avec vérification sur les index. les données sont comparées en utilisant une fonction callback
- Calcule la différence entre deux tableaux
- Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction utilisateur
- Calcule la différence entre deux tableaux en utilisant une fonction callback pour comparer les données
- Calcule la distance levenshtein entre deux chaînes
- Calcule la longueur d'une chaîne
- Calcule la similarité de deux chaînes
- Calcule la somme de contrôle crc32
- Calcule la somme des valeurs du tableau
- Calcule la taille d'une chaîne
- Calcule la valeur de hash demandée par ezmlm
- Calcule le md5 d'un fichier
- Calcule le md5 d'une chaîne
- Calcule le nombre de couleurs d'une palette
- Calcule le rectangle d'encadrement pour un texte, en utilisant la police courante et freetype2
- Calcule le sha1 d'un fichier
- Calcule le sha1 d'une chaîne de caractères
- Change de dossier
- Change de dossier et passe au dossier parent
- Change la casse des clés du tableau

♦ Change la couleur dans une palette à l'index donné.
♦ Change la priorité d'exécution du processus courant
♦ Change le "umask" courant
♦ Change le codage vectoriel d'un caractère dans une police
♦ Change le dossier ftp courant
♦ Change le dossier racine
♦ Change le groupe propriétaire d'un fichier
♦ Change le groupe propriétaire du fichier
♦ Change le mode du fichier
♦ Change le nom de session de l'utilisateur actif
♦ Charge les données ispell
♦ Charge un fichier sur un serveur ftp
♦ Charge un fichier sur un serveur ftp
♦ Charge un lob oracle
♦ Charge une extension php à la volée
♦ Charge une nouvelle police
♦ Charge une police
♦ Charge une police postscript type 1 depuis un fichier
♦ Chiffrage indéchiffrable (hashing)
♦ Chiffre
♦ Chiffre un texte
♦ Chiffre/déchiffre des données en mode cfb
♦ Chiffre/déchiffre des données en mode ecb
♦ Chiffre/déchiffre des données en mode ofb
♦ Chiffre/déchriffre des données en mode cbc
♦ Choisi une fonction utilisateur comme gestionnaire d'erreurs
♦ Choisit l'encodage du client
♦ Choisit la couleur rgb de remplissage et de dessin
♦ Choisit la couleur utilisée pour le dessin
♦ Choisit la couleur utilisée pour le remplissage
♦ Choisit le point de départ de la recherche par expression régulière
♦ Choisit le style de bord autour des liens et annotations
♦ Classe dossier
♦ Clos un paquet wddx
♦ Colorisation de la syntaxe d'un fichier
♦ Combine plusieurs tableaux ensembles, récursivement
♦ Commence l'analyse d'un fichier xml
♦ Commencer un nouveau paquet wddx avec une structure
♦ Compacte des données dans une chaîne binaire
♦ Compacte une base dbase
♦ Comparaison binaire de chaînes
♦ Comparaison binaire des n premiers caractères
♦ Comparaison de chaînes avec l'algorithme d'"ordre naturel"
♦ Comparaison de chaînes avec l'algorithme d'"ordre naturel" (insensible à la casse)
♦ Comparaison de chaînes binaires
♦ Comparaison de chaînes localisée
♦ Compare deux versions de php
♦ Compare en binaire des chaînes de caractères
♦ Compare une entrée avec des valeurs d'attributs
♦ Complète un tableau jusqu'à la longueur spécifiée, avec une valeur.
♦ Complète une chaîne jusqu'à une taille donnée
♦ Compresse une chaîne
♦ Compresse une chaîne avec la méthode deflate
♦ Compte de le nombre d'occurrences de segments dans une chaîne

- Compte le nombre d'enregistrements dans une base dbase
- Compte le nombre d'entrées après une recherche
- Compte le nombre d'occurrences d'une sous−chaîne
- Compte le nombre d'éléments d'une variable
- Compte le nombre de champs d'une base dbase
- Compte le nombre de mots utilisés dans une chaîne
- Compte le nombre de valeurs dans un tableau
- Concatène une matrice avec le ctm
- Configure des pointillés complexes
- Configure l'échelle (??)
- Configure l'échelle du document
- Configure la "miter limit"
- Configure la couleur de dessin et de remplissage
- Configure la couleur de dessin et de remplissage à un niveau de gris
- Configure la couleur de dessin à un niveau de gris
- Configure la couleur de remplissage à un niveau de gris
- Configure la couleur des bords autour des liens et annotations
- Configure la largeur de ligne
- Configure la matrice de transformation
- Configure la police courante
- Configure la position à plat (flatness)
- Configure la rotation
- Configure le gestionnaire de caractères
- Configure le gestionnaire xml de données
- Configure le gestionnaire xml de notations
- Configure le gestionnaire xml de références externes
- Configure le mode de pointillé
- Configure le paramètre de linecap
- Configure le paramètre de linejoin
- Configure le style des lignes autour des liens at annotations
- Configure le style pour le dessin des lignes
- Configure les chaînes et les expressions régulières pour le support des caractères multi−octets
- Configure les fonctions de stockage de sessions
- Configure ou retourne le timeout
- Configure une fonction de callback pour refaire des liaisons, lors de recherche de référants.
- Connexion à une source
- Constantes définies dans le coeur de php, zend et les modules sapi
- Constantes standards dans php
- Construit un 'struct iovec'
- Conversion d'encodage
- Convert it un dn en format ufn (user friendly naming)
- Convertit d'octal en décimal
- Convertit de binaire en décimal
- Convertit de décimal en binaire
- Convertit de décimal en hexadécimal
- Convertit de décimal en octal
- Convertit de hexadécimal en décimal
- Convertit de radians en degrés
- Convertit des données 8bit en texte utf−7.
- Convertit des données binaires en représentation hexadécimale
- Convertit du texte au format mime en utf8
- Convertit entre les différents "kana"

- Convertit l'encodage de variables
- Convertit le nombre de jours du calendrier julien en date grégorienne.
- Convertit le nombre de jours du calendrier julien en date du calendrier français républicain
- Convertit le nombre de jours du calendrier julien en date du calendrier juif.
- Convertit le nombre de jours du calendrier julien en date du calendrier julien.
- Convertit le nombre de jours du calendrier julien en date du calendrier julien.
- Convertit le nombre de jours julien en un calendrier et retourne des détails
- Convertit les caractères 8859 en caractères t61
- Convertit les caractères spéciaux en entités html
- Convertit les caractères t6 en caractères 8859
- Convertit tous les caractères éligibles en entités html
- Convertit toutes les entités html en caractères normaux
- Convertit un calendrier en nombre de jours julien
- Convertit un nombre de degrés en radians
- Convertit un nombre de jours julien en timestamp unix
- Convertit un nombre entre des bases arbitraires
- Convertit un numéro d'erreur ldap en message d'erreur
- Convertit un tableau associatif en une commande postgresql
- Convertit un texte logique hébreux en texte visuel
- Convertit un texte logique hébreux en texte visuel, avec retours à la ligne
- Convertit un timestamp unix en nombre de jours julien
- Convertit une adresse ip (ipv4) en adresse ip numérique
- Convertit une chaîne contenant une adresse (ipv4) ip numérique en adresse littérale.
- Convertit une chaîne d'un jeu de caractères cyrillique à l'autre
- Convertit une chaîne de caractères en tableau
- Convertit une chaîne en nombre à virgule flottante
- Convertit une chaîne iso−8859−1 en utf−8
- Convertit une chaîne quoted−printable en chaîne 8 bits
- Convertit une chaîne utf−8 en iso−8859
- Convertit une chaîne à 8 bits en une chaîne à base64.
- Convertit une chaîne à 8 bits en une chaîne à guillemets.
- Convertit une chaîne à guillemets en une chaîne à 8 bits.
- Convertit une date du calendrier français républicain en nombre de jours du calendrier julien.
- Convertit une date du calendrier juif en nombre de jours du calendrier julien.
- Convertit une date grégorienne en nombre de jours du calendrier julien
- Convertit une image en vraies couleurs en image à palette
- Convertit une image jpeg en image wbmp
- Convertit une image png en image wbmp
- Copie et fusionne une partie d'une image
- Copie et fusionne une partie d'une image en niveaux de gris
- Copie et redimensionne une partie d'une image
- Copie la palette d'une image à l'autre
- Copie les messages spécifiés dans une boîte aux lettres.
- Copie un fichier
- Copie une partie d'une image
- Copie une table dans un tableau
- Copie, redimensionne, rééchantillonne une image
- Cosinus
- Cosinus hyperbolic
- Coupe une chaîne en segments
- Coupe une chaîne en segments
- Coupe une partie de chaîne

♦ Création d'un analyseur xml
♦ Crée un analyseur xml
♦ Crée un dossier
♦ Crée un dossier sur un serveur ftp
♦ Crée un fichier avec un nom unique
♦ Crée un fichier temporaire
♦ Crée un lien
♦ Crée un lien symbolique
♦ Crée un message mime
♦ Crée un morphing
♦ Crée un nouveau bouton
♦ Crée un nouveau champs texte
♦ Crée un nouvel objet d'affichage displayitem
♦ Crée un nouvel objet pdf
♦ Crée un nouvel objet texte
♦ Crée un objet 'animation'
♦ Crée un objet bitmap
♦ Crée un objet de grande taille postgresql
♦ Crée un objet de remplissage
♦ Crée un objet gradient
♦ Crée un processus de pointeur de fichier
♦ Crée un sprite
♦ Crée un tableau
♦ Crée un tableau contenant les variables et leur valeur
♦ Crée un tableau contenant un intervalle d'éléments
♦ Crée un tableau à partir de deux autres tableaux
♦ Crée un vecteur d'initialisation à partir d'une source aléatoire.
♦ Crée une base de données dbase
♦ Crée une base de données mysql
♦ Crée une chaîne compressée
♦ Crée une connexion sur une socket
♦ Crée une fonction anonyme (style lambda)
♦ Crée une image depuis un fichier wbmp
♦ Crée une image wbmp
♦ Crée une image à partir d'un fichier xbm
♦ Crée une image à partir d'un fichier xpm
♦ Crée une image à partir d'une chaîne
♦ Crée une nouvelle action
♦ Crée une nouvelle boîte aux lettres
♦ Crée une nouvelle forme
♦ Crée une nouvelle image jpeg à partir d'un fichier ou d'une url
♦ Crée une nouvelle image png à partir d'un fichier ou d'une url
♦ Crée une nouvelle image en vraies couleurs
♦ Crée une nouvelle image à palette
♦ Crée une nouvelle image à partir d'un fichier gd ou d'une url
♦ Crée une nouvelle image à partir d'un fichier gd2 ou d'une url
♦ Crée une nouvelle image à partir d'un fichier ou d'une url.
♦ Crée une nouvelle image à partir d'une partie de fichier gd2 ou d'une url
♦ Crée une paire de socket identiques et les stocke dans fds.
♦ Crée une socket (point de communication)
♦ Crée une variable php à partir d'une valeur linéarisée

- **D**
- ♦ Dessine la ligne le long du chemin
  ♦ Dessine un arc de cercle dans le sens horaire

- Dessine un arc de cercle pdf dans le sens anti−horaire
- Dessine un caractère horizontalement
- Dessine un caractère verticalement
- Dessine un cercle dans un document pdf
- Dessine un pixel
- Dessine un polygone
- Dessine un polygone rempli
- Dessine un rectangle
- Dessine un rectangle
- Dessine un rectangle rempli
- Dessine un texte avec une police truetype
- Dessine un texte sur une image avec une police postscript type1
- Dessine une chaîne horizontale
- Dessine une chaîne verticale
- Dessine une courbe
- Dessine une courbe de bezier
- Dessine une courbe relativement
- Dessine une ellipse
- Dessine une ellipse partielle
- Dessine une ellipse partielle et la remplit
- Dessine une ellipse pleine
- Dessine une ligne
- Dessine une ligne
- Dessine une ligne pdf
- Dessine une ligne pointillée
- Dessine une ligne relativement
- Décharge le module de chiffrement
- Déchiffre
- Déchiffre un texte
- Déclenche une erreur utilisateur
- Décode les données de session
- Décode les entités html en caractères
- Décode les éléments mime d'une en−tête
- Décode un texte encodé en base64
- Décode une chaîne en mime base64
- Décode une chaîne encodée url
- Décode une chaîne modifiée utf−7.
- Décode une chaîne url
- Décode une en−tête mime
- Décompresse une chaîne
- Décompresse une chaîne compressée
- Déconditionne des données depuis une chaîne binaire.
- Déconnecte d'un serveur ldap
- Déconnecte d'un serveur oracle
- Dédoublonne un tableau
- Définit l'option pour enregistrer les informations complètes du canal alpha lors de la sauvegarde d'images png.
- Définit la couleur transparente
- Définit une constante
- Démarre tls
- Dépile un élément au début d'un tableau
- Dépile un élément de la fin d'un tableau
- Déplace des messages dans une boîte aux lettres
- Déplace l'origine de l'objet swffill

- Déplace le pointeur de lecture
- Déplace le pointeur de résultat
- Déplace le pointeur interne de résultat
- Déplace le stylo
- Déplace le stylo de texte
- Déplace le stylo relativement
- Déplace un fichier téléchargé
- Déplace un objet en coordonnées globales
- Déplace un objet en coordonnées relatives
- Désactive le debugger interne de php.
- Détecte l'encodage mime
- Détecte le type d'encodage d'un caractère http
- Détecte un encodage
- Détermine le type d'une image
- Détermine si l'argument peut être appelé comme fonction
- Détermine si un objet est une sous−classe
- Détermine si une extension est chargée ou non
- Détermine si une image est une image truecolor.
- Détermine si une variable contient une valeur non nulle
- Détermine si une variable est affectée
- Détermine si une variable est de type chaîne
- Détermine si une variable est de type float
- Détermine si une variable est de type int
- Détermine si une variable est de type integer
- Détermine si une variable est de type object
- Détermine si une variable est de type real
- Détermine si une variable est un booléen
- Détermine si une variable est un tableau
- Détermine si une variable est un type numérique
- Détermine si une variable est une ressource
- Détruit le vecteur iovec specifié par iovec_id
- Détruit les données du buffer de sortie, et éteint la bufferisation de sortie
- Détruit toutes les variables d'une session
- Détruit un analyseur xml
- Détruit un pointeur de lob oracle
- Détruit une image
- Détruit une session
- Détruit une session mnogosearch
- Détruit une variable

**E**

- Echappe les méta−caractères
- Echappement des caractères spéciaux des expressions régulières.
- Eclate une chaîne par expression régulière
- Ecrit dans un fichier
- Ecrit dans un fichier compressé, en gzip
- Ecrit dans une socket
- Ecrit des données dans un descripteur de fichier, fd, en utilisant le tableau scatter−gather définie par iovec_id
- Ecrit du texte dans une image avec la police courante et freetype 2
- Ecrit les données de session et ferme la session
- Ecrit un blob temporaire
- Ecrit un fichier sur un serveur ftp, et le lit depuis un fichier (non bloquant)
- Ecrit un lob oracle dans un fichier
- Ecrit un objet de grande taille pgsql

- Ecrit une chaîne dans un fichier
- Ecrit une chaîne dans un fichier compressé gzip
- Ecrit une chaîne formatée dans un flux
- Ecriture du fichier en mode binaire
- Efface des lignes postgresql
- Efface et remplace une portion de tableau
- Efface le buffer de sortie
- Efface le cache de stat
- Efface le résultat de la mémoire
- Efface tous les messages marqués pour l'effacement.
- Efface toutes les erreurs précédemment générées par une socket
- Efface un attribut à l'entrée courante
- Efface un dossier
- Efface un dossier ftp
- Efface un enregistrement dans une base dbase
- Efface un fichier
- Efface un fichier sur un serveur ftp
- Efface un objet de grande taille
- Efface un objet pdf
- Efface un vecteur
- Efface une base de données mysql
- Efface une boîte aux lettres
- Efface une entrée dans un dossier
- Effacer
- Effectue la césure d'une chaîne
- Effectue une recherche
- Effectue une requête partielle pour l'uri spécifiée et renvoie toutes les informations.
- Effectue une selection
- Effectue une sous−requête apache
- Effectue une transformation rot13
- Empile un ou plusieurs éléments au début d'un tableau
- Empile un ou plusieurs éléments à la fin d'un tableau
- Enclenche la bufferisation de sortie
- Encode des entités html
- Encode les données de session
- Encode une chaîne en mime base64
- Encode une chaîne en url
- Encode une chaîne en url, selon la rfc1738
- Encode une chaîne pour une en−tête mime
- Enlève la marque d'effacement d'un message
- Enregistre une fonction exécutée à chaque tick
- Enregistre une fonction pour exécution à l'extinction
- Enregistre une variable dans une session
- Enregistrer plusieurs valeurs dans un paquet wddx
- Enregistrer une valeur dans un paquet wddx
- Envoi tout le contenu généré dans un fichier
- Envoie de mail
- Envoie des données à une socket connectée
- Envoie la commande site au serveur
- Envoie le buffer de sortie
- Envoie les données du buffer de sortie, et éteint la bufferisation de sortie
- Envoie un cookie
- Envoie un en−tête http
- Envoie un fichier mp3 en streaming

- Envoie un fichier sur le serveur ftp (non−bloquant)
- Envoie un mail encodé iso−2022−jp (mail japonais)
- Envoie un message mail
- Envoie un message à une socket, qu'elle soit orientée connexion ou pas
- Envoie une chaîne au serveur postgresql
- Envoie une commande ftp brute
- Envoie une image gif vers un navigateur ou un fichier
- Envoie une image jpeg vers un navigateur ou un fichier.
- Envoie une image png vers un navigateur ou un fichier.
- Envoie une message à une socket, qu'elle soit connectée ou pas
- Envoie une requête mysql à un serveur mysql
- Envoie une requête sql à un serveur mysql
- Envoie votre animation au navigateur
- Etablit une connexion postgresql persistante
- Etablit une connexion à un serveur oracle
- Etablit une nouvelle correction avec un serveur oracle
- Eteind une socket en lecture et/ou écriture
- Etend ou condense une police de caractères
- Etire un objet en coordonnées globales
- Etire un objet relativement
- Exponentielle
- Exporte un objet de grande taille vers un fichier
- Expression exponentielle
- Expression régulière avec support des caractères multi−octets, sans tenir compte de la casse
- Expression régulière globale
- Expression régulière posix pour les chaînes multi−octets
- Expression régulière standard
- Extrait des informations d'un résultat
- Extrait les informations d'une référence d'entrée
- Extrait tous les méta tags d'un fichier
- Extrait une portion de tableau
- Exécute l'appel système select() un tableau de sockets avec une durée d'expiration
- Exécute les sous−requêtes d'un curl
- Exécute un programme externe
- Exécute un programme externe et affiche le résultat brut.
- Exécute un programme externe et affiche le résultat.
- Exécute une chaîne comme un script php
- Exécute une commande et ouvre les pointeurs de fichiers pour les entrées sorties
- Exécute une commande sql oracle
- Exécute une commande sur le serveur ftp
- Exécute une commande via le shell et retourne le résultat sous forme de chaîne.
- Exécute une fonction sur chacun des membres d'un tableau.
- Exécute une requête
- Exécute une requête postgresql asynchrone
- Exécute une requête sql préparée.
- Exécute une requête sql sans mobiliser les résultats
- Exécute une session curl

**F**

- Fait correspondre un peu plus les couleurs de la version palette d'une image aux couleurs de sa version truecolor
- Fait tourner une forme relativement
- Fait tourner une image d'un angle
- Fait une copie d'une police chargée pour modification ultérieure.

- Fait une liste détaillée de fichiers dans un dossier
- Fausse documentation pour main
- Ferme la connexion mnogosearch avec le document enregistré
- Ferme la connexion mysql
- Ferme la connexion à l'historique système
- Ferme la page pdf
- Ferme le fichier pdf
- Ferme le fichier pdf d'entrée
- Ferme le pointeur sur le dossier
- Ferme toutes les connexions odbc
- Ferme un descripteur de lob
- Ferme un fichier
- Ferme un fichier compressé gzip
- Ferme un objet de grande taille postgresql
- Ferme un processus de pointeur de fichier
- Ferme un processus ouvert par proc_open() et retourne le code de sortie du processus.
- Ferme une base dbase
- Ferme une connexion au serveur ldap
- Ferme une connexion ftp
- Ferme une connexion ftp
- Ferme une connexion odbc
- Ferme une image dans un document pdf
- Ferme une session curl
- Ferme une socket
- Filtre les éléments d'un tableau
- Fixe et lit différentes options d'assertions
- Fixe la valeur d'une variable d'environnement
- Fixe le niveau de rapport d'erreurs php
- Fixe le niveau de seuil cubique (??)
- Fixe le nombre maximum d'image dans le sprite
- Fixe le temps maximum d'exécution d'un script
- Fonction de callback pour la compression automatique des buffers
- Fonction de traitement des affichages web
- Fonctionnement des expressions régulières.
- Formate un nombre pour l'affichage
- Formate une date/heure gmt/cut
- Formate une date/heure gmt/cut en fonction des paramétrages locaux.
- Formate une date/heure locale
- Formate une date/heure locale avec les options locales.
- Free result memory
- Fusionne plusieurs tableaux

**G**

- Gestion des colonnes de type long
- Génère la syntaxe colorisée d'un fichier php
- Génère le contexte de débogage
- Génère un identifiant unique
- Génère un message dans l'historique système
- Génère une chaîne de requête en encodage url
- Génère une image au format gd, vers le navigateur ou un fichier
- Génère une image au format gd2, vers le navigateur ou un fichier
- Génère une valeur aléatoire
- Génère une valeur aléatoire (meilleure méthode)
- Générateur de congruence combinée linéaire

- **I**


- ♦ Identification avec une connexion ftp
- ♦ Importe les variables dans la table des symboles
- ♦ Importe les variables de get/post/cookie dans l'environnement global
- ♦ Importe un objet de grande taille depuis un fichier
- ♦ Imprime les crédits de php
- ♦ Incline (abscisses)
- ♦ Incline (ordonnées)
- ♦ Incline le système de coordonnées
- ♦ Incline suivant les x
- ♦ Incline suivant les x relativement
- ♦ Incline suivant les y
- ♦ Incline suivant les y relativement
- ♦ Incline une police de caractères postscript
- ♦ Indique de quoi est capable le navigateur client.
- ♦ Indique la quantité de mémoire utilisée par php
- ♦ Indique le nombre de lignes qui doivent être pré−lues par oracle
- ♦ Indique si l'internaute a abandonné la connexion http
- ♦ Indique si la fin d'un fichier compressé est atteinte
- ♦ Indique si le fichier a été téléchargé par http post
- ♦ Indique si le fichier est exécutable
- ♦ Indique si le fichier est un lien symbolique.
- ♦ Indique si le fichier est un véritable fichier.
- ♦ Indique si le nom de fichier est un dossier
- ♦ Indique si le script a expiré
- ♦ Indique si les en−têtes http ont déjà été envoyés
- ♦ Indique si un algorithme fonctionne par blocs
- ♦ Indique si un fichier est autorisé en écriture
- ♦ Indique si un fichier est autorisé en écriture
- ♦ Indique si un mode fonctionne par blocs
- ♦ Indique si un mode travaille par blocs
- ♦ Indique si un nombre est fini
- ♦ Indique si une fonction est définie
- ♦ Indique si une valeur appartient à un tableau
- ♦ Indique si une valeur est un nombre
- ♦ Indique si une variable est un scalaire
- ♦ Indique si une variable vaut null
- ♦ Indique un fichier est autorisé en lecture
- ♦ Initialise le générateur de nombres aléatoires
- ♦ Initialise tous les buffers nécessaires
- ♦ Initialise toutes les constantes liées au syslog
- ♦ Initialise un nouveau pattern pdf
- ♦ Initialise un nouveau pointeur vide de lob/file
- ♦ Initialise un nouveau template pdf
- ♦ Initialise une nouvelle collection
- ♦ Initialise une nouvelle page de document pdf
- ♦ Initialise une session
- ♦ Initialise une session curl
- ♦ Initialise une valeur aléatoire (meilleure méthode)
- ♦ Insère des lignes dans une table à partir d'un tableau
- ♦ Insère un retour à la ligne html à chaque nouvelle ligne
- ♦ Insère un tableau dans une table
- ♦ Interroge l'autre extrémité de la communication
- ♦ Interroge la socket locale

- Intègre des données binaires iptc dans une image jpeg
- Inverse l'ordre des éléments d'un tableau
- Inverse une chaîne

- **L**
- La plus grand valeur aléatoire possible
- La plus grande valeur
- La plus petite valeur
- Lecture de fichier compressé binaire
- Lecture du fichier en mode binaire
- Lecture du pointeur de fiche courante (cursorname)
- Let et/ou modifie l'identifiant courant de session
- Libère la mémoire
- Libère la mémoire allouée pour ispell par mnogosearch
- Libère la mémoire occupée par une police postscript type 1
- Libère les ressources associées à un résultat
- Libère toutes les ressources occupées par une commande oracle
- Libère toutes les ressources réservées par un résultat oracle
- Libère un résultat mnogosearch
- Lie un nom à une socket
- Lie une variable php à une variable sql d'oracle
- Linéarise une variable
- Lire le nom du groupe
- Lire un paquet wddx
- Liste les bases de données disponibles sur le serveur mysql
- Liste les boîtes aux lettres
- Liste les boîtes aux lettres souscrites
- Liste les boîtes aux lettres, et retourne le détail pour chacune.
- Liste les champs du résultat mysql
- Liste les clés étrangères
- Liste les colonnes d'une table
- Liste les colonnes et leurs droits associés
- Liste les colonnes utilisées dans une clé primaire
- Liste les fichiers et dossier dans un dossier
- Liste les fonctions d'une extension
- Liste les gestionnaires d'affichage utilisés
- Liste les paramètres des procédures
- Liste les processus mysql
- Liste les procédure stockées
- Liste les tables d'une base de données
- Liste les tables d'une source.
- Liste les tables et leurs privilèges
- Liste les types de données supportés par une source
- Liste tous les algorithmes de chiffrement supportés
- Liste tous les modes de chiffrement supportés
- Liste toutes les boîtes aux lettres enregistrées
- Liste toutes les boîtes aux lettres souscrites
- Liste toutes les catégories soeurs d'une catégorie
- Liste toutes les classes définies
- Liste toutes les fonctions définies
- Liste toutes les variables définies
- Lit certains paramètres
- Lit des données d'une socket
- Lit des données dans un fd, en utilisant le tableau scatter−gather définit par iovec_id
- Lit des informations à propos de la dernière requête

- Lit des options passés dans la ligne de commande
- Lit des paramètres numériques
- Lit des paramètres numériques dans le document pdf d'entrée
- Lit des paramètres textuels dans le document pdi
- Lit différenctes options pour la connexion ftp courante
- Lit directement le type de colonne oracle
- Lit et modifie les options des fonctions d'expression régulières à support de caractères multi−octets
- Lit et/ou modifie le chemin de sauvegarde des sessions
- Lit et/ou modifie le limiteur de cache de session
- Lit et/ou modifie le module de session courant
- Lit et/ou modifie le nom de la session
- Lit l'attribut suivant
- Lit l'en−tête du message
- Lit l'en−tête d'un message
- Lit l'encodage du client
- Lit l'heure locale
- Lit l'identifiant de processus du serveur
- Lit l'index de la couleur la plus proche d'une couleur spécifiée avec sa teinte, blanc et noir (hue, white and blackness)
- Lit l'échelle d'un champ
- Lit la configuration du cookie de session
- Lit la configuration interne de l'extension mbstring
- Lit la configuration locale
- Lit la dernière erreur générée par une socket
- Lit la liste des boîtes aux lettres
- Lit la liste des boîtes aux lettres, et y recherche une chaîne
- Lit la liste des boîtes aux lettres, et y recherche une chaîne
- Lit la longueur d'un champ
- Lit la position courante du pointeur de lecture
- Lit la prochaine entrée
- Lit la prochaine ligne dans le résultat oracle
- Lit la précision d'une colonne oracle
- Lit la référence suivante
- Lit la structure d'un message.
- Lit la structure d'une section du corps d'un mail
- Lit la taille d'une colonne oracle
- Lit la totalité d'un fichier compressé
- Lit la valeur d'un élément d'une collection oracle à l'index ndx
- Lit la valeur d'une option de configuration
- Lit la valeur de la directive de configuration include_path
- Lit la version courante du moteur zend
- Lit la version des api mnogosearch
- Lit le chemin de la catégorie courante
- Lit le contenu courant du buffer de sortie puis l'efface
- Lit le corps d'un message
- Lit le dernier code d'erreur
- Lit le dernier message d'erreur
- Lit le dernier message d'erreur sur la connexion
- Lit le dernier segment de chaîne multi−octets qui correspond au masque
- Lit le dn d'une entrée
- Lit le fichier et renvoie le résultat dans un tableau
- Lit le message d'erreur associé à un résultat
- Lit le message d'erreur de l'analyseur xml

♦ Lit le message de notify
♦ Lit le mime−type d'un type d'image
♦ Lit le nom de la colonne
♦ Lit le nom de la table qui contient un champ
♦ Lit le nom du chiffrement utilisé
♦ Lit le nombre total de documents dans les bases mnogosearch
♦ Lit le statut de la connexion postgresql
♦ Lit le statut du buffer de sortie
♦ Lit le statut du résultat
♦ Lit le tampon contenant le fichier pdf généré
♦ Lit les attributs d'une entrée
♦ Lit les données dns associées à un hôte
♦ Lit les données du vecteur specifié par iovec_id[iovec_position]
♦ Lit les données méta de la table postgresql
♦ Lit les en−têtes exif des images jpeg et tiff
♦ Lit les informations concernant un processus
♦ Lit les informations détaillant un transfert curl
♦ Lit les informations sur l'hôte mysql
♦ Lit les informations sur le client mysql
♦ Lit les informations sur le protocole mysql
♦ Lit les informations sur le serveur mysql
♦ Lit les informations sur les transferts actuels
♦ Lit les informations sur un fichier à partir d'un pointeur de fichier
♦ Lit les informations à propos de la boîte aux lettres courante.
♦ Lit les noms des bases de données
♦ Lit les options d'un analyseur xml
♦ Lit les options de socket
♦ Lit les paramètres de résultats mnogosearch
♦ Lit les quotas de chaque utilisateur
♦ Lit les quotas des boîtes aux lettres
♦ Lit tous les en−têtes de la requête http
♦ Lit tous les en−têtes de réponse http
♦ Lit tous les en−têtes http de la requête
♦ Lit tout le fichier compressé
♦ Lit tout un fichier dans une chaîne
♦ Lit toutes les entrées du résultat
♦ Lit toutes les lignes d'un résultat
♦ Lit toutes les valeurs binaires d'une entrée
♦ Lit toutes les valeurs d'une entrée ldap
♦ Lit toutes les valeurs de configuration
♦ Lit un caractère dans un fichier
♦ Lit un caractère dans un fichier compressé
♦ Lit un champ de résultat mnogosearch
♦ Lit un champ de résultat uodbc
♦ Lit un enregistrement dans une base dbase
♦ Lit un enregistrement dans une base dbase, sous la forme d'un tableau associatif
♦ Lit un fichier sur un serveur ftp, et l'écrit dans un fichier (non bloquant)
♦ Lit un fichier sur un serveur ftp, et l'écrit dans un fichier (non bloquant)
♦ Lit un objet de grande taille
♦ Lit un objet de grande taille en totalité
♦ Lit un résultat asynchrone
♦ Lit un sommaire des en−têtes de messages
♦ Lit une entrée
♦ Lit une entrée du dossier

♦ Lit une image depuis un fichier
♦ Lit une ligne dans un fichier compressé
♦ Lit une ligne dans un fichier compressé, et supprime les balises
♦ Lit une ligne dans un tableau
♦ Lit une ligne de résultat
♦ Lit une ligne de résultat dans un objet
♦ Lit une ligne de résultat dans un tableau associatif
♦ Lit une ligne de résultat pgsql dans un objet
♦ Lit une ligne de résultat pgsql dans un tableau
♦ Lit une ligne de résultat pgsql sous forme de tableau numérique
♦ Lit une ligne de résultat, et la place dans un tableau.
♦ Lit une ligne de résultats dans un tableau associatif
♦ Lit une sous−chaîne
♦ Lit/modifie l'encodage d'affichage
♦ Lit/modifie l'encodage interne
♦ Lit/modifie l'ordre de détection des encodages
♦ Lit/modifie le langage courant
♦ Lit/modifie les caractères de substitution
♦ Lit/écrit la valeur courante d'une option
♦ Logarithme en base 10
♦ Logarithme naturel (népérien)

- **M**
- ♦ Marque le fichier pour l'effacement, dans la boîte aux lettres courante
♦ Message d'erreur mnogosearch
♦ Met en majuscule la première lettre de tous les mots
♦ Met la socket en mode bloquant
♦ Met le premier caractère en majuscule
♦ Met tous les caractères en majuscules
♦ Met tous les caractères en minuscules
♦ Met un nombre au format monétaire
♦ Modifie certains paramètres
♦ Modifie l'alignement du texte
♦ Modifie l'espacement de lignes
♦ Modifie l'espacement de police
♦ Modifie l'image utilisée pour le carrelage
♦ Modifie l'indentation de la première ligne
♦ Modifie l'échelle de la forme
♦ Modifie l'épaisseur d'un trait
♦ Modifie la brosse pour le dessin des lignes
♦ Modifie la casse d'une chaîne
♦ Modifie la couleur de fond
♦ Modifie la couleur de la police
♦ Modifie la couleur de rastériation de droite
♦ Modifie la couleur de rastérisation de gauche
♦ Modifie la couleur du champs texte
♦ Modifie la date de modification et de dernièr accès d'un fichier
♦ Modifie la gestion des colonnes de données binaires
♦ Modifie la hauteur de la police courante
♦ Modifie la hauteur de la police du champs texte
♦ Modifie la ligne courant dans un résultat
♦ Modifie la marge de droite
♦ Modifie la marge de gauche
♦ Modifie la place en profondeur (z−order)
♦ Modifie la police du champs

♦ Modifie la position dans un objet de grande taille
♦ Modifie la position du texte
♦ Modifie la valeur d'une option de configuration
♦ Modifie la valeur d'une option ldap
♦ Modifie la valeur de la directive de configuration include_path
♦ Modifie la vitesse de l'animation
♦ Modifie le acl de la boîte aux lettres
♦ Modifie le chemin de clipping pdf
♦ Modifie le mode de blending d'une image
♦ Modifie le nom d'une entrée
♦ Modifie le nombre total d'images dans l'animation
♦ Modifie le pointeur de fichier
♦ Modifie le quota d'une boîte aux lettres
♦ Modifie le ratio de l'objet
♦ Modifie le style de ligne de la forme
♦ Modifie les dimensions de l'animation
♦ Modifie les données du vecteur iovec_id[iovec_position]
♦ Modifie les droits d'un fichier via ftp
♦ Modifie les informations de localisation
♦ Modifie les lignes d'une table
♦ Modifie les marges du champs texte flash
♦ Modifie les options de la connexion ftp
♦ Modifie les options de socket
♦ Modifie les paramètres de l'agent mnogosearch
♦ Modifie les paramètres du cookie de session
♦ Modifie les paramètres numériques
♦ Modifie les paramètres odbc.
♦ Modifie une entrée ldap
♦ Modifie une option de transfert curl
♦ Modifie une variable de subprocess_env apache
♦ Multiplie la couleur de transformation
♦ Mélange les caractères d'une chaîne de caractères
♦ Mélange les éléments d'un tableau

- **N**
- ♦ Nom de la base de données
♦ Nombre de colonnes dans un résultat
♦ Nombre de lignes dans un résultat
♦ Nomme le champs texte
♦ Nomme un objet
♦ Numéro d'erreur mnogosearch
♦ Numéro de colonne

- **O**
- ♦ Obsolète : ajoute un signet dans la page courante
♦ Obsolète : ajoute une annotation
♦ Obsolète : configure l'espace entre deux mots
♦ Obsolète : configure l'espacement de caractères character spacing
♦ Obsolète : configure l'échelle horizontale du texte
♦ Obsolète : configure l'élévation de texte
♦ Obsolète : configure la distance entre deux lignes de texte
♦ Obsolète : configure la durée entre deux pages
♦ Obsolète : configure la matrice de texte
♦ Obsolète : détermine le rendu du texte
♦ Obsolète : gère les polices
♦ Obsolète : lit le nom de la police

♦ Obsolète : ouvre un nouvel objet pdf
♦ Obsolète : ouvre une image gif
♦ Obsolète : ouvre une image jpeg
♦ Obsolète : ouvre une image png
♦ Obsolète : ouvre une image tiff
♦ Obsolète : remplit le champ d'auteur du document
♦ Obsolète : remplit le champ de créateur du document
♦ Obsolète : remplit le champ de mots clés du document
♦ Obsolète : remplit le champ de sujet du document
♦ Obsolète : remplit le champ de titre du document
♦ Obsolète : sélectionne la taille et le nom d'une police
♦ Obsolète : termine le chemin courant
♦ Obsolète: charge une police
♦ Options disponibles pour les expressions régulières.
♦ Ouverture d'un fichier ou d'une url
♦ Ouvre la connexion à l'historique système
♦ Ouvre le module de l'algorithme et le mode à utiliser
♦ Ouvre un dossier, et récupère un pointeur dessus.
♦ Ouvre un fichier compressé avec gzip
♦ Ouvre un fichier pdf
♦ Ouvre un flux imap vers une boîte aux lettres
♦ Ouvre un flux imap vers une nouvelle boîte aux lettres.
♦ Ouvre un nouveau fichier pdf
♦ Ouvre un objet de grande taille postgresql
♦ Ouvre une base dbase
♦ Ouvre une connexion ftp
♦ Ouvre une connexion ftp sécurisée avec ssl
♦ Ouvre une connexion mnogosearch avec un document stocké
♦ Ouvre une connexion persistante à un serveur mysql.
♦ Ouvre une connexion persistante à un serveur oracle
♦ Ouvre une connexion persistante à une source de données.
♦ Ouvre une connexion postgresql
♦ Ouvre une connexion à un serveur mysql
♦ Ouvre une image
♦ Ouvre une image contenant des données brutes ccitt
♦ Ouvre une image créée en mémoire par php
♦ Ouvre une socket de connexion internet ou unix
♦ Ouvre une socket de connexion internet ou unix persistante.
♦ Ouvre une socket sur un port pour accepter les connexions

- **P**
- ♦ Passe à l'image suivante
♦ Pingue la connexion à la base
♦ Pingue le serveur, et se reconnecte au besoin
♦ Place le point courant pdf
♦ Place un point de couleur pdf
♦ Place une image dans la page
♦ Place une page dans le document
♦ Plus grande valeur aléatoire possible
♦ Positionne le pointeur de tableau en fin de tableau
♦ Positionne un flag sur un message
♦ Prend une ou plusieurs valeurs, au hasard dans un tableau
♦ Protège le caractères d'une chaîne binaire en mode bytea
♦ Protège les caractères spéciaux d'une commande sql.
♦ Protège les caractères spéciaux du shell

- Protège une chaîne de caractères
- Protège une chaîne de caractères pour utilisation en ligne de commande
- Protège une chaîne pour la passer à mysql_query()
- Prépare et exécute une requête sql
- Prépare le module pour le déchargement
- Prépare une commande pour l'exécution
- Prépare une page
- Prépare une police pour utilisation ultérieure avec pdf_setfont

**R**

- Racine carrée
- Rassemble des informations sur la langue et la configuration locale.
- Rassemble les éléments d'un tableau en une chaîne
- Rassemble les éléments d'un tableau en une chaîne
- Recheche des chemins qui vérifie un masque
- Recherche dans un niveau
- Recherche dans un tableau la clé associée à une valeur
- Recherche par expression régulière avec support des caractères multi−octets
- Recherche par expression régulière multi−octets
- Recherche sur le serveur ldap
- Rechercher et remplacer par expression régulière standard
- Rechercher/remplacer avec une expression régulière et fonction de callback
- Recommence la lecture du fichier
- Recule le pointeur courant de tableau
- Regular expression match ignoring case with multibyte support
- Relance la connexion au serveur postgressql
- Remet le pointeur interne de tableau au début
- Remet à zéro l'environnement graphique pdf
- Remplace des chaînes dans une chaîne
- Remplace des segments de chaînes, avec le support des expressions régulières mutli−octet
- Remplace l'identifiant de session courant par un nouveau
- Remplace les clés par les valeurs, et les valeurs par les clés
- Remplace toutes les occurrences dans une chaîne
- Remplace un attribut dans l'entrée courante
- Remplace un enregistrement dans une base dbase
- Remplace un segment dans une chaîne
- Remplis un tableau avec une même valeur
- Remplit
- Remplit avec une région avec une couleur spécifique
- Remplit et passe le pinceau sur le chemin pdf courant
- Remplit le chemin pdf courant avec la couleur courante
- Remplit un champ de l'en−tête de document pdf
- Renomme un fichier
- Renomme un fichier sur un serveur ftp
- Renomme une boîte aux lettres
- Renvoie l'espace disque disponible dans le répertoire.
- Renvoie l'heure à laquelle l'inode a été accédé pour la dernière fois.
- Renvoie la date de dernière modification du fichier.
- Renvoie la date à laquelle le fichier a été accédé pour la dernière fois.
- Renvoie la ligne courante et cherche les champs csv
- Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier et élimine les balises html
- Renvoie la ligne courante sur laquelle se trouve le pointeur du fichier.
- Renvoie la position du pointeur du fichier

- Renvoie la taille du fichier
- Renvoie le nom du dossier
- Renvoie le nom du fichier vers lequel pointe un lien symbolique
- Renvoie le nom du propriétaire du fichier
- Renvoie le numéro d'inode du fichier
- Renvoie les informations à propos d'un fichier
- Renvoie les informations à propos d'un lien
- Renvoie les permissions affectées au fichier.
- Renvoie une chaîne en majuscules
- Renvoie une chaîne en minuscules
- Replace le pointeur de fichier au début
- Reprend le téléchargement d'un fichier (non bloquant)
- Repère la dernière occurence d'un caractère dans une chaîne
- Repère la première occurence d'un caractère dans une chaîne
- Repère un fichier à partir d'un masque
- Rercherche la position d'une occurence dans une chaîne, sans tenir compte de la casse
- Restaure la valeur de l'option de configuration
- Restaure la valeur de la directive de configuration include_path
- Retarde l'exécution
- Retarde l'exécution en micro−secondes
- Retire un curl multiple d'un jeu de curl
- Retourne true si un objet a pour parent une classe donnée
- Retourne chaque paire clé/valeur d'un tableau
- Retourne des détails sur un calendrier
- Retourne des détails sur une colonne mysql
- Retourne des informations sur la connexion courante
- Retourne des informations sur un chemin système
- Retourne des informations à propos de la librairie gd installée
- Retourne des statistiques sur les caractères utilisés dans une chaîne
- Retourne l'adresse ip correspondant à un hôte
- Retourne l'arbre de reference organisé par thread
- Retourne l'en−tête d'un message
- Retourne l'ensemble optimal de colonnes, qui permet de définir uniquement une ligne dans une table
- Retourne l'heure actuelle
- Retourne l'identifiant du dernier objet
- Retourne l'identifiant du logo php
- Retourne l'identifiant du thread courant
- Retourne l'identifiant généré par la dernière requête insert.
- Retourne l'index d'une couleur avec son canal alpha
- Retourne l'index de l'octet courant d'un analyseur xml
- Retourne l'index de la couleur d'un pixel donné
- Retourne l'index de la couleur donnée
- Retourne l'index de la couleur donnée, ou la plus proche possible.
- Retourne l'index de la couleur la plus proche d'une couleur donnée.
- Retourne l'inode du script
- Retourne l'offset du début du prochain segment repéré par une expression régulière
- Retourne l'option d'actino pour keypress (char)
- Retourne l'uid d'un message.
- Retourne l'uid du propriétaire du script actuel
- Retourne l'élément courant d'un tableau
- Retourne la classe d'un objet
- Retourne la configuration actuelle de l'option magic_quotes_gpc

- Retourne la configuration actuelle de l'option magic_quotes_runtime
- Retourne la configuration actuelle du cache expire
- Retourne la couleur associée à un index
- Retourne la couleur la plus proche, en tenant compte du canal alpha
- Retourne la date de dernière modification d'un fichier sur un serveur ftp.
- Retourne la date de dernière modification de la page
- Retourne la date/heure
- Retourne la dernière erreur (si elle existe) qui est survenu lors de la dernière requête.
- Retourne la dernière erreur de stmt|conn|global sur oracle
- Retourne la dernière note du serveur postgresql
- Retourne la fin de la chaîne
- Retourne la hauteur d'une bitmap
- Retourne la hauteur d'une image
- Retourne la hauteur de l'image
- Retourne la hauteur de la police
- Retourne la largeur d'un texte avec la police courante
- Retourne la largeur d'une bitmap
- Retourne la largeur d'une chaîne
- Retourne la largeur d'une image
- Retourne la largeur d'une image
- Retourne la largeur de la police
- Retourne la ligne suivante d'un résultat oracle dans un tableau
- Retourne la liste d'ip correspondante à un hôte.
- Retourne la liste de tous les modules compilés et chargés
- Retourne la liste des constantes et leur valeur
- Retourne la liste des entêtes de réponse du script courant
- Retourne la liste des fichiers .ini analysés dans les dossiers de configuration supplémentaires
- Retourne la liste des fichiers dans un dossier
- Retourne la longueur du contenu du buffer de sortie
- Retourne la méhode de compression utilisée avec gzip
- Retourne la plus grande valeur d'une collection oracle
- Retourne la position courante dans un objet de grande taille
- Retourne la position et la longueur du segement de chaîne qui vérifie le masque de l'expression régulière
- Retourne la première entrée
- Retourne la somme de contrôle crc32 d'une chaîne
- Retourne la table de traduction des entités utilisée par htmlspecialchars et htmlentities
- Retourne la taille d'un champ de résultat mysql
- Retourne la taille d'un dossier
- Retourne la taille d'un fichier.
- Retourne la taille d'une chaîne
- Retourne la taille d'une collection oracle
- Retourne la taille d'une colonne oracle
- Retourne la taille d'une image
- Retourne la taille de blocs d'un algorithme
- Retourne la taille de blocs d'un algorithme
- Retourne la taille de blocs d'un chiffrement
- Retourne la taille de chaque colonne d'une ligne de résultat
- Retourne la taille de la chaîne
- Retourne la taille de la clé d'un chiffrement
- Retourne la taille du vi d'un algorithme

♦ Retourne la taille du vi utilisé par un couple chiffrement/mode
♦ Retourne la taille imprimée
♦ Retourne la taille interne de stockage d'un champ donné.
♦ Retourne la taille maximale de clé
♦ Retourne la taille maximale de la clé pour un mode
♦ Retourne la valeur d'une colonne dans un résultat oracle
♦ Retourne la valeur d'une constante
♦ Retourne la valeur d'une option de php
♦ Retourne la valeur d'une variable d'environnement
♦ Retourne la valeur de la variable, au format chaîne
♦ Retourne la valeur de pi
♦ Retourne la valeur numérique entière équivalent d'une variable
♦ Retourne la version courante de curl
♦ Retourne la version d'apache
♦ Retourne le acl pour la boîte aux lettres
♦ Retourne le chemin canonique absolu
♦ Retourne le code ascii d'un caractère
♦ Retourne le code php utilisé pour générer une variable
♦ Retourne le contenu du buffer de sortie
♦ Retourne le contenu obtenu avec l'option curlopt_returntransfer
♦ Retourne le dossier de travail
♦ Retourne le gid du propriétaire du script
♦ Retourne le jeu de caractères courant pour les expressions régulières
♦ Retourne le logo de zend
♦ Retourne le message ldap de la dernière commande ldap
♦ Retourne le niveau d'utilisation des ressources
♦ Retourne le nom d'hôte
♦ Retourne le nom d'hôte correspondant à une ip
♦ Retourne le nom d'un champ postgresql
♦ Retourne le nom d'une colonne
♦ Retourne le nom d'une colonne oracle
♦ Retourne le nom de l'algorithme
♦ Retourne le nom de la classe d'un objet
♦ Retourne le nom de la table où se trouve une colonne
♦ Retourne le nom de protocole associé à un numéro de protocole
♦ Retourne le nom de tty
♦ Retourne le nom du dossier courant
♦ Retourne le nom du jeu de caractères utilisé par le client mysql
♦ Retourne le nom du mode
♦ Retourne le nom du mois.
♦ Retourne le nom du possesseur du script courant.
♦ Retourne le nombre courant de colonne d'un analyseur xml
♦ Retourne le nombre courant de colonne d'un analyseur xml.
♦ Retourne le nombre d'arguments passé à la fonction
♦ Retourne le nombre de champ
♦ Retourne le nombre de champs d'un résultat mysql
♦ Retourne le nombre de colonnes dans un résultat
♦ Retourne le nombre de jours dans un mois, pour une année donnée
♦ Retourne le nombre de jours entre le 21 mars et pâques, pour une année donnée.
♦ Retourne le nombre de lignes affectées
♦ Retourne le nombre de lignes affectées lors de la dernière requête sql
♦ Retourne le nombre de lignes affectées par oracle
♦ Retourne le nombre de lignes d'un résultat
♦ Retourne le nombre de lignes postgresql

- Retourne le nombre de message dans la boîte aux lettres courante.
- Retourne le nombre de messages récents dans la boîte aux lettres courante.
- Retourne le nombre de niveaux d'imbrications du système de bufferisation de sortie
- Retourne le numéro d'erreur de la dernière commande mysql
- Retourne le numéro d'erreur ldap de la dernière commande exécutée.
- Retourne le numéro d'une colonne
- Retourne le numéro de la version courante de php
- Retourne le numéro de ligne courant d'un analyseur xml.
- Retourne le numéro de port
- Retourne le numéro de port associé à un service internet et un protocole.
- Retourne le numéro de processus courant
- Retourne le numéro de protocole associé à un nom de protocole
- Retourne le numéro de séquence de message pour un uid donné.
- Retourne le numéro de version majeur de pdflib
- Retourne le numéro de version mineure de pdflib
- Retourne le numéro du jour de la semaine.
- Retourne le rectangle entourant un texte et dessiné avec une police postscript type1.
- Retourne le rectangle entourant un texte et dessiné avec une police truetype.
- Retourne le reste de la division
- Retourne le résultat de exp(number) − 1, calculé de manière précise, même si val est proche de 0
- Retourne le résultat de log(1 + number), calculé de manière précise, même si val est proche de zéro.
- Retourne le résultat de sqrt( num1*num1 + num2*num2)
- Retourne le segment de chaîne trouvé par une expression régulière multi−octets
- Retourne le service internet qui correspond au port et protocole.
- Retourne le statut courant du serveur
- Retourne le texte associé avec l'erreur générée lors de la dernière requête.
- Retourne le timestamp unix actuel avec microsecondes
- Retourne le timestamp unix actuel.
- Retourne le timestamp unix d'une date
- Retourne le timestamp unix d'une date gmt
- Retourne le type d'interface utilisé entre le serveur web et php
- Retourne le type d'un champ postgresql donné par index
- Retourne le type de commande oci
- Retourne le type de données d'une colonne oracle
- Retourne le type de fichier
- Retourne le type de la colonne courante
- Retourne le type de la variable
- Retourne le type de ressource
- Retourne les arguments d'une fonction sous forme de tableau
- Retourne les bits de status de la connexion http
- Retourne les données enregistrées dans une colonne sous forme d'objet.
- Retourne les en−têtes de tous les messages d'une boîte aux lettres.
- Retourne les enregistrements mx d'un hôte
- Retourne les informations de statut sur une boîte aux lettres autres que la boîte courante.
- Retourne les informations sur le système d'exploitation
- Retourne les informations sur une socket
- Retourne les informations à propos d'un fichier ou d'un lien symbolique.
- Retourne les noms des méthodes d'une classe
- Retourne les options postgresql
- Retourne les sockets associées à curl, pour utilisation
- Retourne les types d'images supportés par la version courante de php

- Retourne les valeurs d'un résultat
- Retourne les valeurs d'un tableau
- Retourne les valeurs par défaut des attributs d'une classe.
- Retourne toutes les alertes
- Retourne toutes les clés d'un tableau
- Retourne toutes les erreurs
- Retourne toutes les lignes d'un résultat oracle
- Retourne un caractère spécifique
- Retourne un champ d'un résultat
- Retourne un entier représentant le dernier message d'erreur curl
- Retourne un identifiant de type de serveur ftp.
- Retourne un index de couleur ou son alternative la plus proche, y compris le canal alpha
- Retourne un nouveau curl multiple
- Retourne un nouveau pointeur à utiliser pour lier les pointeurs de références
- Retourne un segment de chaîne
- Retourne un tableau associatif des propriétés d'un objet
- Retourne un tableau avec les noms des fichiers qui sont inclus dans un script
- Retourne un tableau avec les résultat de la recherche
- Retourne un tableau contenant les tailles de clés acceptées par un algorithme
- Retourne un tableau contenant les tailles de clés supportée par un algorithme de chiffrement
- Retourne un tableau de message après recherche.
- Retourne un timestamp unix pour pâques, à minuit
- Retourne un élément de la liste des arguments
- Retourne une adresse email formatée correctement
- Retourne une chaîne contenant le dernier message d'erreur curl
- Retourne une chaîne contenant les informations de version du serveur oracle
- Retourne une chaîne décrivant un message d'erreur
- Retourne une chaîne formattée
- Retourne une chaîne formatée
- Retourne une clé d'un tableau associatif
- Retourne une ligne de résultat mysql sous la forme d'un objet
- Retourne une ligne de résultat mysql sous la forme d'un tableau
- Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux
- Retourne une section extraite du corps d'un message
- Retourne à la première entrée du dossier
- Return first attribute
- Return first reference
- Reçoit des données d'une socket connectée
- Reçoit des données d'une socket, connectée ou pas
- Reçoit des messages sur uen socket, qu'elle soit orientée connexion ou pas
- Réactive l'ancienne fonction de gestion des erreurs
- Récupère la miniature d'une image tiff ou jpeg
- Réduit itérativement un tableau
- Répéte une chaîne
- Résolution dns d'une adresse ip
- Rétablit l'ancien environnement graphique pdf

- **S**
- 
  - Sauve dans un fichier
  - Sauve l'environnement graphique courant
  - Sauve un fichier lob oracle
  - Sauve un lob oracle

- Scinde une chaîne
- Scinde une chaîne en tableau avec une expression régulière multi−octets
- Se connecte à un serveur ldap
- Sinus
- Sinyus hyperbolique
- Souscrit à une boîte aux lettres
- Stocke un message d'erreur
- Supprime le protection d'une chaîne de type bytea
- Supprime les anti−slash d'une chaîne c
- Supprime les balises html et php d'une chaîne
- Supprime les caractères invisibles de début de chaîne
- Supprime les espaces de fin de chaîne
- Supprime les espaces en début et fin de chaîne
- Supprime num éléments à la fin de la collection oracle
- Supprime un flag sur un message
- Supprime un objet d'une animation
- Supprime un objet d'une animation
- Supprime un objet dans un sprite
- Supprime une collection oracle
- Supprime une couleur d'une image
- Supprime une variable de la session
- Supprimer les anti−slash d'une chaîne
- Synchronise avec le serveur postgresql
- Synonyme de checkdnsrr
- Synonyme de getmxrr
- Synonyme de odbc_exec
- Sélectionne la forme de départ
- Sélectionne la forme de fin
- Sélectionne la largeur et hauteur du champs
- Sélectionne la police courante
- Sélectionne le mode non bloquant d'un descripteur de fichier
- Sélectionne une base de données mysql
- Sépare le nom du fichier et le nom du dossier.
- Sépare les différents composants d'un dn
- Sépare un tableau en tableaux de taille inférieure

**T**

- Tangente
- Tangente hyperbolique
- Termine la lecture de ressources oracle
- Termine la page pdf courante
- Termine la souscription à une boîte aux lettres
- Termine le chemin et dessine les bords
- Termine le chemin pdf courant
- Termine le chemin, dessine les bords et remplit la forme
- Termine le pattern pdf
- Termine le processus apache après cette requête
- Termine le script courant
- Termine le suivi d'une connexion postgresql
- Termine le template pdf
- Termine un chiffrage
- Termine un flux imap
- Termine un jeu de transferts curl
- Termine un processus ouvert par proc_open()
- Termine une connexion postgresql

- Teste un module ouvert
- Teste la fin du fichier
- Teste le chiffrage par blocs d'un algorithme
- Teste le chiffrage par blocs d'un mode
- Teste si la valeur d'une colonne oracle est null
- Teste si le mode retourne les données par blocs
- Teste si un champ postgresql est à null
- Teste un mode
- Tourne la forme
- Tourne un objet en angle absolu
- Traite un fichier de configuration
- Transforme un texte anglais en timestamp
- Transforme une chaîne ascii en ebcdic
- Transforme une chaîne ebcdic en ascii
- Transforme une liste de variables en tableau
- Transforme une variable en tableau
- Translate l'origine du système de coordonnées
- Tri d'un tableau avec l'algorithme à "ordre naturel"
- Tri d'un tableau avec l'algorithme à "ordre naturel" insensible à la casse
- Tri multi−dimensionnel
- Trie des messages
- Trie en ordre inverse
- Trie les entrées d'un résultat ldap
- Trie un tableau
- Trie un tableau en ordre
- Trie un tableau en ordre inverse
- Trie un tableau en sens inverse et suivant les clés
- Trie un tableau en utilisant une fonction de comparaison personnalisée
- Trie un tableau en utilisant une fonction de comparaison personnalisée
- Trie un tableau par ses clés en utilisant une fonction de comparaison définie par l'utilisateur
- Trie un tableau suivant les clés
- Tronque un fichier
- Tronque une chaîne
- Trouve la position d'un caractère dans une chaîne
- Trouve la position de la dernière occurence d'un caractère dans une chaîne
- Trouve la position de la dernière occurence d'une chaîne dans une autre de façon insensible à la casse.
- Trouve la première occurence d'un caractère dans une chaîne
- Trouve la première occurence dans une chaîne
- Trouve la première occurence dans une chaîne (insensible à la casse)
- Trouve le premier segment de chaîne
- Trouve un segment de chaîne ne contenant pas certains caractères
- Type de données d'un champ
- Télécharge un fichier depuis un serveur ftp
- Télécharge un fichier via ftp dans un flux local

**U**

- Utilise la version de swf (??)
- Utilise un analyseur xml à l'intérieur d'un objet.
- Utilise une variable php pour la phase de définition, dans une commande select oracle
- Utiliser ou non les fonctions d'antialias

**V**

- Va à la prochaine image du sprite

♦ Valeur absolue
♦ Valide les transactions oracle en cours
♦ Valide une date grégorienne
♦ Valide une transaction odbc
♦ Verrouille le fichier
♦ Version insensible à la casse de str_replace
♦ Vide le tampon, le retourne en tant que chaîne et stoppe la mise en tampon.
♦ Vide les buffers de sortie
♦ Vérifie l'existence d'une constante
♦ Vérifie la connexion mnogosearch avec document stocké
♦ Vérifie le courrier de la boîte aux lettres courante.
♦ Vérifie qu'une classe a été définie
♦ Vérifie que la méthode existe pour une classe.
♦ Vérifie que le flux imap est toujours actif
♦ Vérifie si la connexion postgresql est occupée
♦ Vérifie si plusieurs résultats sont disponibles
♦ Vérifie si un fichier existe
♦ Vérifie si un jeu de caractère est supporté par mnogosearch
♦ Vérifie si une assertion est fausse
♦ Vérifie si une clé existe dans un tableau
♦ Vérifie si une variable est enregistrée dans la session

## 8.3 Index des exemples

♦ **A**
♦
◊ Exemple avec abs
◊ Accès aux membres d'une classe
◊ Accès aux valeurs rgb
◊ Activer l'extension bzip2 sous windows
◊ Exemple avec addaction.php">swfbutton−>addaction
◊ exemple avec addcslashes et charlist (2)
◊ Exemple avec addfill.php">swfshape−>addfill
◊ Addition de tableaux
◊ Affichage d'adresse ip
◊ Affichage des fichiers inclus et requis
◊ Affichage des informations d'en−tête d'un fichier de base de données dbase
◊ Affiche la liste des attributs d'une entrée
◊ Afficher des données issues d'un formulaire
◊ Afficher sans l'aide de echo
◊ Afficher une somme de contrôle crc32
◊ Afficher une structure xml
◊ Ajout d'entrées dans un tableau.
◊ Alternatives à scandir
◊ Analyser le résultat de socket_select
◊ Exemple avec apache_get_version
◊ Exemple avec apache_request_headers
◊ Argument de fonction de lecture
◊ Exemple avec array_change_key_case
◊ Exemple avec array_chunk
◊ Exemple avec array_combine
◊ Exemple avec array_count_values
◊ Exemple avec array_diff
◊ Exemple avec array_diff_assoc
◊ Exemple avec array_diff_uassoc

◊ Exemple avec array_fill
◊ array_filter sans callback
◊ Exemple avec array_flip : collision
◊ Exemple avec array_intersect
◊ Exemple avec array_intersect_assoc
◊ Exemple avec array_key_exists
◊ Exemple avec array_merge
◊ Exemple avec array_merge_recursive
◊ Exemple avec array_pad
◊ Exemple avec array_pop
◊ Exemple avec array_push
◊ Exemple avec array_rand
◊ Exemple avec array_reduce
◊ Exemple avec array_reverse
◊ Exemple avec array_search
◊ Exemple avec array_shift
◊ Exemple avec array_sum
◊ Exemple avec array_udiff
◊ Exemple avec array_udiff_assoc
◊ Exemple avec array_udiff_uassoc
◊ array_unique et les types de valeurs
◊ array_values pour les utilisateurs php 3
◊ Exemple avec array_walk
◊ Exemple avec arsort
◊ Exemple avec asort
◊ Attention aux erreurs de fractions

♦ **B**

♦ ◊ Exemple avec base64_decode
◊ base64_decode example
◊ Exemple avec base64_encode
◊ Exemple avec base_convert
◊ Exemple avec basename

♦ **C**

♦ ◊ Calcul du hash et enregistrement d'un utilisateur de mailing liste
◊ Exemple avec call_user_func et une méthode de classe
◊ Exemple avec call_user_func_array
◊ Exemple avec call_user_method
◊ Exemple avec ceil
◊ Exemple avec chr
◊ Exemple avec chunk_split
◊ Classer un tableau multi−dimensionnel
◊ Collection
◊ Collection
◊ Colorisation d'url
◊ Commande utilisables avec mysql_info
◊ Comment utiliser le paramètre mode de chmod
◊ Commentaires emboîtés
◊ Exemple avec compact
◊ Compactage d'une chaîne
◊ Comparaison d'objets composés en php 4
◊ Compilation cgi avec php (unix)
◊ Compter le nombre de hits d'un utilisateur sur une page
◊ Configuration de php.ini pour les utilisateurs de sjis
◊ Configuration de l'utilisateur d'exécution d'apache

◊ Configuration isapi de sambar
◊ Configuration par défaut des paramètres
◊ Configuration pour utiliser simultanément php 3 et 4
◊ Configurer php avec apache 2.0 en module
◊ Connexion anonyme à un serveur ldap
◊ Constructeur de classe en php 4
◊ Conversion automatique indéfinie(2)
◊ Conversion des fichiers .htaccess pour php 4
◊ Conversion en booléen
◊ Conversion html en texte
◊ Exemple avec copy
◊ Exemple avec count_chars
◊ Exemple avec crypt
◊ Création d'un document pdf en mémoire
◊ création d'un flux d'image gd, et affichage
◊ Création d'un objet
◊ Création d'un tableau de tableaux
◊ Création d'une base dbase
◊ création d'une image gd et affichage de cette image
◊ Création d'une image png avec php
◊ Créer un fichier gzip
◊ Exemple avec current

♦ **D**

♦ ◊ Exemple avec debug_backtrace
◊ Dernier jour du mois
◊ Dessin d'un cercle avec imagearc
◊ Destruction d'une session avec $_session
◊ Exemple avec dir
◊ Exemple avec dirname
◊ Exemple avec disk_free_space
◊ Exemple avec disk_total_space
◊ Divisions entières
◊ Exemple avec dl
◊ Exemple avec dns_get_record et dns_any
◊ Décoder des entités html
◊ Définir une constante
◊ Définition d'une constante
◊ Désactiver l'analyse des fichiers php dans un dossier avec .htaccess
◊ détection automatique des formats d'image supportés avec imagetypes

♦ **E**

♦ ◊ Exemple avec easter_days
◊ Effacer une ressource existante
◊ Exemple avec empty
◊ Encryption d'une valeur avec tripledes sous 2.4.x en mode ecb
◊ Exemple avec end
◊ enregistrer une variable avec register_globals activé
◊ Enumerer tous les messages d'erreur ldap
◊ Envoi de mail complexe avec mail
◊ Equivalent de variables variables
◊ Exemple avec escapeshellarg
◊ Exemple avec escapeshellcmd
◊ exemple avec eval − concaténation de texte
◊ Exempe avec constant
◊ Exempe avec mb_convert_case

◊ Exemple imap_mime_header_decode
◊ Exemple imap_status
◊ Exemple mysql_data_seek
◊ Exemple mysql_db_name
◊ exemple mysql_num_rows
◊ Exemple mysql_stat
◊ Exemple restore_include_path
◊ Exemple setline.php">swfshape−>setline
◊ Exemple setname.php">swfdisplayitem−>setname
◊ Exemple str_shuffle
◊ Exemple au format de date iso 8601:1988
◊ Exemple complet avec identification ldap
◊ Exemple complet de vérification de mot de passe avec ldap
◊ exemple d'envoi de page compressée avec ob_gzhandler
◊ Exemple d'installation de php sous openbsd avec ports
◊ Exemple d'introduction
◊ Exemple d'utilisation de max
◊ Exemple d'utilisation de miniature avec exif_thumbnail
◊ Exemple d'écriture dans un fichier avec fwrite
◊ Exemple de mysql_fetch_array avec mysql_both
◊ Exemple de swfsprite
◊ Exemple de bufferisation de sortie
◊ Exemple de chaîne heredoc
◊ Exemple de comparaison d'objets en php 5
◊ Exemple de configuration apache
◊ Exemple de connexion à un serveur ldap ssl
◊ exemple de création de base mysql avec mysql_create_db
◊ Exemple de fonction de callback
◊ exemple de gestion d'erreur lors de la création d'image (gracieusement offert par vic@zymsys.com )
◊ exemple de gestion d'erreur lors de la création d'image (gracieusement offert par vic@zymsys.com )
◊ Exemple de gestion de sortie avec fonction utilisateur
◊ exemple de gestion des erreurs durant la création d'image (gracieusement offert par vic@zymsys.com )
◊ exemple de gestion des erreurs durant la création d'une image wbmp (gracieusement proposé par vic@zymsys.com)
◊ Exemple de l'extension zlib
◊ Exemple de message du débuggeur
◊ Exemple de modification d'option odbc
◊ Exemple de modification de niveau d'erreur
◊ Exemple de nombres à virgule flottante
◊ Exemple de présentation de l'extension mysql
◊ Exemple de script avec mbstring
◊ Exemple de session : page2.php
◊ Exemple de variable fonction
◊ Exemple ftp
◊ Exemple getrusage
◊ Exemple mysql_close
◊ Exemple pdfclock de pdflib
◊ exemple récursif avec count (php >= 4.2.0)
◊ Exemples avec array_slice
◊ Exemples avec array_splice
◊ Exemples avec array_unshift

◊ Exemples avec easter_date
◊ Exemples avec error_log
◊ Exemples avec min
◊ Exemples avec next
◊ Exemples avec next
◊ Exemples avec pg_connect
◊ exemples avec preg_replace_callback et create_function
◊ Exemples avec print
◊ Exemples avec require
◊ Exemples avec soundex
◊ Exemples avec stripos
◊ Exemples avec l'opérateur ::
◊ Exemples avec l'opérateur parent
◊ Exemples avec les opérateurs d'assignation
◊ Exemples avec les opérateurs sur les bits
◊ Exemples avec quelques constantes liées au système
◊ Exemples d'interpolation des chaînes de caractères
◊ Exemples d'utilisation de str_word_count
◊ Exemples de chaînes
◊ Exemples de conversions
◊ Exemples de masques invalides
◊ Exemples de vérification de type de variable
◊ Exemples liés à str_split
◊ Exemple avec exif_imagetype
◊ Exemple avec exif_read_data
◊ Exemple avec exit
◊ Exemple avec explode
◊ Exemple avec extension_loaded
◊ Exemple avec extract
◊ Extrait du php.ini
◊ Exécute une requête postgresql asynchrone

♦ **F**

♦ ◊ Exemple avec fclose
◊ Exemple avec fgetc
◊ Fichier batch pour exécuter un script php en ligne de commande (script.bat)
◊ Fichier batch pour exécuter un script php en ligne de commande (script.bat)
◊ Exemple avec file
◊ Exemple avec filemtime
◊ Exemple avec floatval
◊ Exemple avec floor
◊ Exemple avec fmod
◊ Fonctions calendrier
◊ Fonctions d'objets : test_script.php
◊ Fonctions dans une autre fonctions
◊ Fonctions à nombre d'arguments variable
◊ Exemple avec fopen sous windows
◊ Format monétaire avec sprintf
◊ Formatage avec date
◊ Exemple avec ftp_alloc
◊ Exemple avec ftp_chdir
◊ Exemple avec ftp_connect
◊ Exemple avec ftp_get_option
◊ Exemple avec ftp_login
◊ Exemple avec ftp_nlist

◊ Exemple avec ftp_put
◊ Exemple avec ftp_set_option
◊ Exemple avec func_get_arg
◊ Exemple avec func_get_args
◊ Exemple avec func_num_args
◊ Exemple avec function_exists

♦ **G**

♦ ◊ Genérer et intercepter une erreur
◊ Gestion d'erreurs avancées en php
◊ Gestion des assertions avec un gestionnaire personnalisé
◊ Exemple avec get_browser
◊ Exemple avec get_class
◊ Exemple avec get_class_methods
◊ Exemple avec get_class_vars
◊ Exemple avec get_defined_constants
◊ Exemple avec get_defined_functions
◊ Exemple avec get_defined_vars
◊ Exemple avec get_extension_funcs
◊ Exemple avec get_include_path
◊ Exemple avec get_loaded_extensions
◊ Exemple avec get_magic_quotes_gpc
◊ Exemple avec get_object_vars
◊ Exemple avec get_parent_class
◊ Exemple avec get_resource_type
◊ Exemple avec getdate
◊ Exemple avec getenv
◊ Exemple avec gethostbyaddr
◊ Exemple avec gethostbyaddr
◊ Exemple avec gethostbynamel
◊ getimagesize qui retourne iptc
◊ Exemple avec getlastmod
◊ Exemple avec getopt
◊ Exemple avec gmdate
◊ Exemple avec gmstrftime
◊ Exemple avec gzopen
◊ Exemple avec gzread

♦ **H**

♦ ◊ Exemple avec headers_list
◊ Exemple avec headers_sent
◊ Exemple avec hexdec
◊ Exemple avec htmlentities
◊ Exemple avec htmlspecialchars
◊ Héritage de classes (2)

♦ **I**

♦ ◊ Identification http avec nom d'utilisateur/mot de passe forcé
◊ Exemple avec image2wbmp
◊ Image_type_to_mime_type (file)
◊ Exemple avec imagecolorallocate
◊ Exemple avec imagecolordeallocate
◊ Exemple avec imageellipse
◊ Exemple avec imagefilledellipse
◊ Exemple avec imagepng
◊ Exemple avec imagepolygon
◊ Exemple avec imagepsloadfont

◊ Exemple avec imagesetstyle
◊ Exemple avec imagestring
◊ Exemple avec imagesx
◊ Exemple avec imagesy
◊ Exemple avec imagettftext
◊ Exemple avec imagetypes
◊ Exemple avec imap_append
◊ Exemple avec imap_createmailbox
◊ Exemple avec imap_delete
◊ Exemple avec imap_fetch_overview
◊ Exemple avec imap_get_quota pour php 4.3
◊ Exemple avec imap_get_quotaroot
◊ Exemple avec imap_getmailboxes
◊ Exemple avec imap_list
◊ Exemple avec imap_mail_compose
◊ Exemple avec imap_mailboxmsginfo
◊ Exemple avec imap_open
◊ Exemple avec imap_rfc822_parse_adrlist
◊ Exemple avec imap_rfc822_write_address
◊ Exemple avec imap_set_quota
◊ Exemple avec imap_setflag_full
◊ Exemple avec implode
◊ implémentation de array_keys pour les utilisateurs de php 3
◊ Exemple avec import_request_variables
◊ exemple avec in_array et deux tableaux en paramètres
◊ include et return
◊ include_once est insensible à la casse sous windows
◊ Inclure une image php en pdf
◊ Indexation automatique de tableau
◊ Exemple avec ini_get_all
◊ initialiser une session curl et récupérer une page web
◊ Insertion à grande vitesse dans une table
◊ Installation sur mac os x server
◊ Installer php comme module apache
◊ Instruction alternative if () elseif et else
◊ Instruction break
◊ Instruction continue
◊ Instruction d'installation (module partagé apache 2)
◊ Instruction declare
◊ Instruction elseif ()
◊ Instruction for() alternative
◊ Instruction foreach() et while() (3)
◊ Instruction if () else
◊ Instruction if () et bloc
◊ Instruction switch() alternative
◊ Instruction while () et break
◊ Instructions d'installation avec apache (version module)
◊ Instructions d'installation caudium
◊ Inverser une chaîne avec strrev
◊ Exemple avec inversion de la table de traduction des caractères en entités html
◊ Exemple avec is_bool
◊ Exemple avec is_callable
◊ Exemple avec is_scalar

◊ Exemple avec isset et un tableau
♦ **J**
♦ ◊ Exemple avec jdtojewish
◊ Jeu de caractères qui fonctionnent avec php
♦ **K**
♦ ◊ Exemple avec krsort
◊ Exemple avec ksort
♦ **L**
♦ ◊ La valeur null
◊ Exemple avec le paramètre convmap
◊ Le séparateur d'instructions
◊ Lecture d'un fichier ligne par ligne
◊ Lecture d'un objet postgres
◊ Les arguments à valeur par défaut doivent être en premier : valide
◊ Exemple avec les sockets : client tcp/ip simple
◊ Les variables sont locales à la fonction (2)
◊ Les variables statiques et la récursivité
◊ Les variables statiques et les références (2)
◊ Les variables super globales
◊ Lire des informations sur les champs postgresql
◊ Lire des messages d'erreurs avec popen
◊ Lire les méta−données d'une table postgresql
◊ Lire un nom de domaine dans une url
◊ Exemple avec list en tenant compte de l'ordre
◊ liste tous les fichiers du dossier courant, sauf "." et ".."
◊ Liste toutes les valeurs de l'attribut "mail" d'une entrée
◊ lit et affiche le contenu d'un fichier csv avec fgetcsv
◊ Exemple avec localeconv
◊ Exemple avec ltrim
♦ **M**
♦ ◊ Manipuler des éléments de tableau
◊ Manipuler des éléments de tableau
◊ Exemple avec mb_convert_encoding
◊ Exemple avec mb_convert_kana
◊ Exemple avec mb_convert_kana
◊ Exemple avec mb_convert_variables
◊ Exemple avec mb_detect_encoding
◊ Exemple avec mb_detect_order
◊ Exemple avec mb_encode_numericentity
◊ Exemple avec mb_internal_encoding
◊ Exemple avec mb_output_handler
◊ Exemple avec mb_preferred_mime_string
◊ Exemple avec mb_strimwidth
◊ Exemple avec mb_strtolower
◊ Exemple avec mb_strtoupper
◊ Exemple avec mb_substitute_character
◊ Exemple avec mb_substr_count
◊ Exemple avec mcrypt_create_iv
◊ Exemple avec mcrypt_enc_get_algorithms_name
◊ Exemple avec mcrypt_enc_get_modes_name
◊ Exemple avec mcrypt_enc_get_supported_key_sizes
◊ Exemple avec mcrypt_encrypt
◊ Exemple avec mcrypt_get_block_size
◊ Exemple avec mcrypt_get_block_size

◊ Exemple avec mcrypt_get_block_size
◊ Exemple avec mcrypt_get_cipher_name
◊ Exemple avec mcrypt_list_algorithms
◊ Exemple avec mcrypt_list_modes
◊ Exemple avec md5
◊ Exemple avec mdecrypt_generic
◊ Exemple avec memory_get_usage
◊ Exemple avec microtime
◊ Migration depuis 2.0: concaténation de chaînes
◊ Migration depuis 2.0: valeur retournées, nouvelle façon
◊ Migration des variables globales
◊ Migration: nouvelle syntaxe if..endif
◊ Migration: nouvelle syntaxe while..endwhile
◊ Mise en italique d'un mot dans un texte
◊ Exemple avec mkdir
◊ Modification d'une variable d'environnement
◊ Exemple avec money_format
◊ Exemple avec mt_srand
◊ Exemple avec multcolor.php">swfdisplayitem−>multcolor
◊ Exemple avec mysql_client_encoding
◊ Exemple avec mysql_connect
◊ Exemple avec mysql_errno
◊ Exemple avec mysql_error
◊ Exemple avec mysql_escape_string
◊ Exemple avec mysql_fetch_assoc
◊ Exemple avec mysql_fetch_field
◊ Exemple avec mysql_fetch_object
◊ Exemple avec mysql_field_name
◊ Exemple avec mysql_get_client_info
◊ Exemple avec mysql_get_host_info
◊ Exemple avec mysql_get_proto_info
◊ Exemple avec mysql_get_server_info
◊ Exemple avec mysql_insert_id
◊ Exemple avec mysql_list_dbs
◊ Exemple avec mysql_list_fields
◊ Exemple avec mysql_list_processes
◊ Exemple avec mysql_list_tables
◊ Exemple avec mysql_query (2)
◊ Exemple avec mysql_real_escape_string
◊ Exemple avec mysql_result
◊ Exemple avec mysql_tablename
◊ Exemple avec mysql_thread_id
◊ Méthode avancée

♦ **N**

♦ ◊ Exemple avec natsort
◊ Nombre variable d'argument sous forme de tableau
◊ Notre premier script php : bonjour.php
◊ Nouveau comportement de strtok
◊ Nouvelles balises php
◊ Exemple avec number_format

♦ **O**

♦ ◊ Exemple avec ob_end_clean
◊ Exemple avec ob_end_flush
◊ Exemple avec ob_get_clean

◊ Exemple avec ob_get_flush
◊ Exemple avec ob_list_handlers
◊ Exemple avec ocibindbyname
◊ Exemple avec ocicolumnname
◊ Exemple avec ocicolumnsize
◊ Exemple avec ocicolumntype
◊ Ocicommit
◊ Exemple avec ocidefinebyname
◊ Exemple avec ocifetchinto
◊ Exemple avec ocifetchstatement
◊ Exemple avec ocilogon
◊ Exemple avec ocinewdescriptor
◊ Exemple avec ocinlogon
◊ Exemple avec ocinumcols
◊ Exemple avec ocirowcount
◊ Exemple avec ociserverversion
◊ Exemple avec ocistatementtype
◊ Exemple avec odbc_fetch_into en php 4.2.0
◊ Exemple avec odbc_result
◊ Exemple avec opendir
◊ Opérations arithmétiques et caractères
◊ Opérations arithmétiques sur un caractère
◊ Exemple avec ord
◊ Exemple avec output_add_rewrite_var
◊ Exemple avec output_reset_rewrite_vars

♦ **P**

♦ ◊ Exemple avec parse_ini_file
◊ Exemple avec parse_str
◊ Exemple avec parse_url
◊ parsemoldb.php : analyse moldb.xml et crée un tableau d'objets moléculaires
◊ Passage de valeur par référence, ponctuellement
◊ Passer du mode php au mode html et vice−versa
◊ passer en revue $_post avec each
◊ Exemple avec pathinfo
◊ Exemple avec pdf_add_bookmark
◊ Exemple avec pdf_findfont
◊ Exemple avec pdf_setcolor : couleurs cmyk
◊ Personnalisation de la page de crédits php
◊ Exemple avec pg_affected_rows
◊ Exemple avec pg_close
◊ Exemple avec pg_connection_busy
◊ Exemple avec pg_connection_reset
◊ Exemple avec pg_connection_status
◊ Exemple avec pg_dbname
◊ Exemple avec pg_delete
◊ Exemple avec pg_fetch_all
◊ Exemple avec pg_fetch_array
◊ Exemple avec pg_fetch_assoc
◊ Exemple avec pg_field_is_null
◊ Exemple avec pg_get_notify
◊ Exemple avec pg_get_pid
◊ Exemple avec pg_insert
◊ Exemple avec pg_ping
◊ Exemple avec pg_select

◊ Exemple avec pg_update
◊ Exemple avec php_logo_guid
◊ Exemple avec php_sapi_name
◊ Exemple avec phpinfo
◊ Exemple avec phpversion
◊ Exemple avec pi
◊ Postgres retourne une ligne
◊ Exemple avec preg_grep
◊ Exemple avec proc_open
◊ Produit une liste de tous les services d'une société avec ldap_list
◊ Profiler une section de code php

♦ **Q**

♦ ◊ Quelques exemples avec pow

♦ **R**

♦ ◊ Exemple avec range
◊ Exemple avec rawurlencode
◊ Exemple avec realpath
◊ Recherche avec ldap
◊ Recherche ldap
◊ Recherche les couples de balises html (gourmand)
◊ Exemple avec register_tick_function
◊ Reinstaller apache avec pthread
◊ Exemple avec rename
◊ reprise du téléchargement à la position 100 avec ftp_nb_get
◊ require_once est insensible à la casse sur windows
◊ Requête sql update
◊ Resuming an upload with ftp_nb_put
◊ Retourner une référence d'une fonction
◊ Exemple avec rotateto.php">swfdisplayitem−>rotateto
◊ Exemple avec round
◊ Exemple avec rsort
◊ Exemple avec rtrim
◊ Références et constructeurs
◊ Résultat de get_meta_tags
◊ résultat de set_error_handler et trigger_error

♦ **S**

♦ ◊ Sauver un fichier par méthode put en php 4
◊ Scinde une chaîne et capture les offsets
◊ Exemple avec serialize
◊ Exemple avec session_cache_expire
◊ Exemple avec session_cache_limiter
◊ Exemple avec session_name
◊ Exemple avec session_regenerate_id
◊ Exemple avec session_register
◊ exemple avec session_set_save_handler
◊ Set server controls
◊ Exemple avec set_include_path
◊ Exemple avec setcookie
◊ Exemple avec setlocale pour windows
◊ Exemple avec settype
◊ Exemple avec sha1
◊ Exemple avec shuffle
◊ Exemple avec sin
◊ Exemple avec socket_sendto

◊ Exemple avec socket_strerror
◊ Exemple avec sort
◊ sprintf : format de monétaire
◊ spécifier une catégorie avec un format hiérachisé
◊ Exemple avec sqrt
◊ Exemple avec srand
◊ Exemple avec sscanf : utilisation des paramètres optionnels
◊ Stocker des données sur un serveur distant
◊ Exemple avec str_ireplace
◊ Exemple avec str_pad
◊ Exemple avec str_repeat
◊ Exemple avec str_replace
◊ Exemple avec strcasecmp
◊ Exemple avec strip_tags
◊ Exemple avec stripslashes
◊ Exemple avec stristr
◊ Exemple avec strlen
◊ Exemple avec strnatcmp
◊ Exemple avec strpos
◊ Exemple avec strrchr
◊ Exemple avec strrpos
◊ Exemple avec strspn
◊ Exemple avec strstr
◊ Exemple avec strtolower
◊ Exemple avec strtoupper
◊ Exemple avec strtr avec deux arguments
◊ Structure d'instruction exécutable
◊ Structure d'un entier
◊ Structure du tableau retourné par ldap_get_entries
◊ Exemple avec substr_count
◊ Exemple avec substr_replace
◊ Exemple avec swfaction
◊ Exemple avec swfbitmap
◊ Exemple avec swfgradient
◊ Exemple avec swfmorph
◊ Exemple avec swfmovie−>streammp3
◊ Exemple avec swfshape
◊ Exemple avec swftext
◊ Syntaxe complexe de chaîne

♦ **T**

♦ ◊ Tableau d'index commençant à 1
◊ Tableaux multi−dimensionnels, et récursifs
◊ Exemple avec tempnam
◊ Test d'existence d'un fichier
◊ Exemple avec tmpfile
◊ Exemple avec touch
◊ Transtypage
◊ Transtypage en chaîne
◊ Transtypage xml −> html
◊ tri avec usort sur un objet
◊ Exemple avec trigger_error
◊ Exemple avec trim
◊ Types mysql
◊ Télécharger plusieurs fichiers simultanément

♦ **U**
♦ ◊ Exemple avec ucfirst
◊ Exemple avec ucwords
◊ Exemple avec udm_api_version
◊ Exemple avec udm_cat_list
◊ Exemple avec udm_load_ispell_data et udm_ispell_type_server
◊ Exemple avec uksort
◊ Un exemple simple avec strripos
◊ Un exemple simple de liste retournée par php_ini_scanned_files
◊ un moyen pratique pour remplacer opendir par glob
◊ Exemple avec uniqid
◊ Exemple avec unpack
◊ Exemple avec unserialize
◊ Exemple avec unset avec une variable globale
◊ Exemple avec urldecode
◊ exemple avec urlencode et htmlentities
◊ Using mcrypt_module_open in encryption
◊ Utilisateurs.txt
◊ Utilisation d'une connexion udp
◊ Utilisation d'une constante dans un index de tableau(2)
◊ Utilisation d'une image pour soumettre un formulaire
◊ Utilisation de ftp_raw pour se logger manuellement
◊ Utilisation de gd_info
◊ utilisation de header pour générer un fichier de type pdf ou d'un autre type
◊ Utilisation de http_build_query avec un objet
◊ Utilisation de syslog
◊ Utilisation de curl et php pour récupérer une page
◊ Utilisation de fonctions anonymes comme fonction de callback
◊ Utilisation de paquets incrémentaux
◊ Utilisation de procédures stockées
◊ Utilisation des accolades {} dans les chaînes pour les tableaux
◊ Utilisation des constantes true et false avec une structure de contrôle
◊ Utilisation des fichiers binaires avec fread
◊ Utilisation des objets de grande taille (large objects)
◊ Utilisation des tableaux avec setcookie
◊ Utilisation du paramètre return de print_r
◊ Utiliser dbase_numfields
◊ Utiliser un ref cursor issue d'une commande select
◊ Utiliser une valeur négative de length
♦ **V**
♦ ◊ Validation de téléchargement de fichiers
◊ Exemple avec var_dump
◊ Exemple avec var_export
◊ Variables de formulaires complexes
◊ Exemple avec version_compare
◊ Vérification de la version du protocole avec ldap_get_option
◊ Vérification des erreurs avec strtotime
◊ vérifier la présence de constantes avec defined
◊ Vérifier le nom d'une couleur avec un masque shell
♦ **W**
♦ ◊ wddx_serialize_vars
◊ Exemple avec while ()
◊ Windows include_path
◊ Exemple avec wordwrap avec césure

♦ **X**
♦ ◊ Xmltest2.xml
♦ **Z**
♦ ◊ Exemple avec zend_version