



Licence Info

Programmation Spécifique : Programmation Web

XHTML / CSS / PHP /
Javascript



Plis fòs ba pengwen là !

QUID ?

- APACHE : Serveur WEB
- HTML : Langage hypertexte de description de documents
- PHP : Langage de programmation interprété coté serveur
- MYSQL : Gestionnaire de base de données
- Javascript : Langage de programmation interprété coté client.

Pourquoi ces choix :

- Le prix (gratuit)
- La qualité (utilisés par les entreprises
ex : Apache = 2/3 des serveurs WEB)

Plan du cours

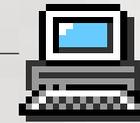
- Pour tout : 6h TD / xxh TP
- Html
 - xhtml
 - Css (feuilles de style).
- PHP - MySQL
 - Php de base.
 - Interaction avec des SGBDD
 - Création d'images à la volée / creation de pdf....

Serveur WEB

- Un serveur WEB est une application capable de répondre aux requêtes de clients.
- Il est connecté au clients via un réseau (souvent).
- Il communique avec le client à l'aide du protocole HTTP.
- Il fournit aux clients une page HTML.



Serveur



Clients

Types de Sites

On distingue trois types de sites :

- Site Statique : les requêtes portent sur des fichiers html.
- Site Dynamique : les requêtes déclenche sur le serveur l'execution d'un script. Le script produit une page HTML.
- Site "Sapin de Noël" : les requêtes portent sur des fichiers html contenant des scripts coté client.

Attention : un commercial qui parle d'un site dynamique parle (parlait) souvent d'un sapin de noel....

On peut avoir un site dynamique et sapin de noel...

HTML

Langage hypertexte de description de documents

- Hypertexte : permet une navigation non-linéaire
- Interprété par un client http :
 - adaptation au client (taille, polices...)
- Fournit une description simple d'un document.
- Documents composés de texte, images.

Nous utiliserons : XHTML 1.0

- Conforme aux recommandations du w3c
- Indépendant (?) du navigateur.

HTML

Architecture d'un document

- Format d'un document html : texte simple.
- Ce qui est dans le fichier est affiché, sauf instructions html.
- Instructions sous formes de balises (ou « tags »).
- 2 grandes parties dans un document :
 - Doctype : indique quel version du langage est utilisée
 - Entête (head) : informations générales sur la page
 - Corps (body) : Ce qui sera affiché sur le client.

HTML

fonctionnement des balises

- Une balise est délimitée par des crochets.
- Toutes doivent être ouvertes et fermées.
- Elles peuvent avoir des attributs (options).

```
<p>
```

Du texte ici

```

```

Puis encore du texte.

```
</p>
```

HTML

Exemple

Première page HTML typique :

```
<!doctype .....>
<html>
  <head>
    <title> Ma premi&egrave;re page</title>
  </head>
  <body>
    <p>Le texte dans ma premi&eagrave;re page. </p>
  </body>
</html>
```

HTML

Balises de structuration du document

- `<doctype>` : renseigne sur la version du langage utilisé
- `<html></html>` : ...
- `<head></head>` : entête.
- `<body></body>` : corps du document.

HTML

Balises de mise en forme des pages

pour décrire le découpage des pages en zones :

- `<div> </div>` : une unité qui fait sens pour le site (ex : une entete, un menu,)

Pour ce découpage, il y eut (désormais obsolète et malvenu) :

- `<frameset><frame>`
- `<table> <tr><td>`

HTML

Balises de mise en forme de texte :

- `<p></p>` : un paragraphe.
- ` ` : accentué (gras visuellement)
- ` ` : mis en évidence (italique visuellement)
- `` : exposant. (superscript).
- `` : indice (subscript).

HTML

Balises de mise en forme de texte 2

- `<xmp></xmp>` : non interprété (affiché tel quel).
- `<h1></h1>` : premier niveau de titre. (aussi `<h2>...`)
- `<!-- toto -->` : commentaire.
- `` : liste numérotée.
 - `` : élément de liste.
- `` : liste à puces.

HTML

Balises de mise en forme de texte :
les forbidden : Ne pas utiliser !

- `
` : passage à la ligne.
- `<hr>` : ligne horizontale.
- `<i></i>` : italique.
- `` : gras (bold).

HTML

Balises utiles

- `` : insertion d'images.
- `<a>` : ancre. Notamment pour les liens.
- `<table></table>` : tableau.
 - `<tr></tr>` : ligne d'un tableau.
 - `<td></td>` : case d'un tableau.
- Attribut : `colspan/rowspan` : multi colonne/ligne.

HTML

Balises blocs et inline

Deux types de balises :

- blocs (des parties entières de texte)

ex : `<p>` `<div>` `<h1>`

- en ligne (inline) : enrichit une portion de texte

ex : `` `` ``

- Les blocs se suivent (par défaut) l'un en dessous de l'autre.
- Les inline se suivent (par défaut) l'un a coté de l'autre.
- Une inline ne peut contenir que des inline (et du texte)
- une bloc peut contenir des blocs, des inline, du texte.

HTML

Les feuilles de styles

On les appelle aussi *css* pour « Cascading Style Sheets ».

- Objectif : Définir le style d'un document html indépendamment de son contenu.
- Intérêt : Votre site est beau, bien rempli...vous voulez changer l'image de fond de toutes les pages...
- Mise en oeuvre : Dans un fichier, référencé dans le document html.

Version des css actuelles : 2.0



Incompatible avec certains navigateurs

HTML

Fonctionnement des feuilles de styles

Le document html indique l'utilisation de feuilles de styles avec la ligne suivante dans l'entête :

```
<link rel=stylesheet type="text/css" href="mafeuille.css">
```

La feuille de style indique ensuite des règles pour le formatage.



```
/* Définitions des titres niveau 2 */
```

```
h2 {color:#0000ca; text-align:left; font-size:large; font-style:italic}
```

HTML

Limiter les problèmes de polices grâce aux css

Voici comment préciser trois polices possibles dans l'ordre de préférence, en fonction de leur disponibilité.

```
h3
{
font-family: verdana, arial, sans-serif;
font-size: 14pt; color: red;
background-color: #ffffff;
}
```

HTML

surcharger des tags grâce aux css

Ici, on utilise la balise `<p>` pour laquelle on définit une nouvelle classe : `petit`

```
p.petit
{ font-family: times, serif;
font-size: 10pt;
color:black;
font-style:italic;
background-color : #ffffff;
}
```

```
<p class="petit"> ceci va etre ecrit en petit </p>
```

HTML

utiliser au mieux l'hérité dans les css

Vous pouvez créer des sous-classes applicables à toute balise.

```
.colored{ color:red; }  
.marge { margin-left:30px }
```

Ceci est bien pratique pour faire des choses comme :

```
<h1 class="colored"> Un gros titre rouge </h1>  
<b class="colored"> du texte gras et rouge </b>  
<p class="marge"> un paragraphe avec une marge </p>
```

HTML

utiliser au mieux l'hérité dans les css 2

La balise `` est utilisée en CSS pour modifier l'aspect d'une zone particulière de données.

```
.rouge{ color:red; }
```

Ceci est bien pratique pour faire des choses comme :

```
<span style= "font-size: 16pt; color: fuchsia; background-color: #FFFFFF; " >un truc très occasionnel</span>
```

```
<span class="red">juste mettre en rouge.</span>
```



Ne pas en abuser...

HTML

utiliser au mieux l'hérédité dans les css 3

La balise <div> est utilisée en CSS pour modifier l'aspect de paragraphes

```
.marge{ margin-left:30px }
```

Ceci est bien pratique pour faire des choses comme :

```
<div class= "marge" >tout un paragraphe avec sa marge. Mais c'est  
comme <p> alors ? Ben non, on verra ca plus loin...</div>
```

HTML

Les liens dans les css

Les pseudos classes liées aux liens (balise <a>) :

link : lien normal.

visited : lien déjà visité.

hover : on passe dessus avec la souris.

active : on a cliqué dessus.

```
a:link { color:blue; text-decoration:underline; background-color: #ffffff; }
a:visited { color:purple; text-decoration:underline ; background-color:
#ffffff; }
a:hover { color:red; text-decoration:underline; background-color:
#ffffff; }
a:active { color: red; font-weight: bold; background-color: #ffffff; }
```

HTML

Positionnement dans les css

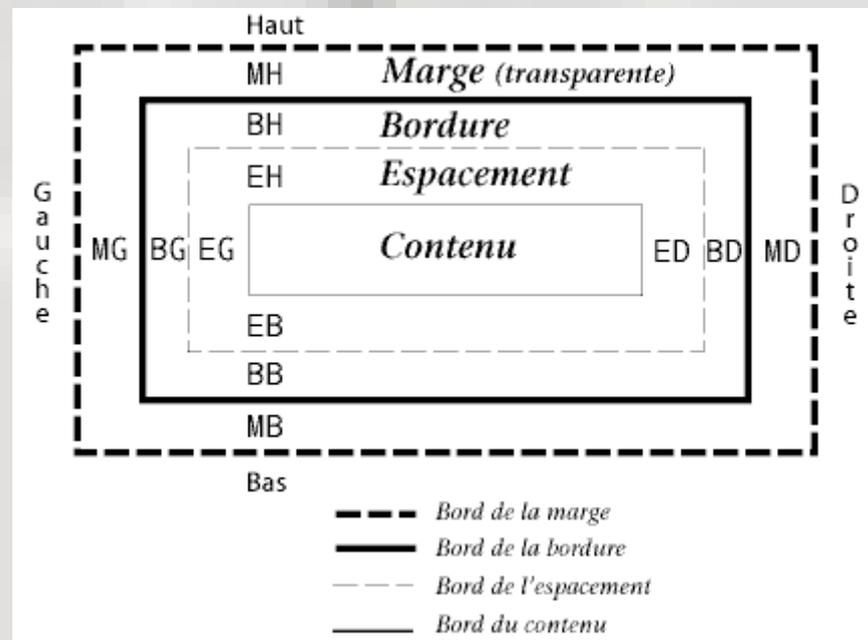
Le modèle de boîte : vaut pour toutes les balises bloc :

En anglais (et en css) :

margin : marge externe.

border : bordure

padding : marge interne



Positionnement et CSS

Positionnement dans le flux

Le positionnement normal : dans le flux.

Les éléments les uns en dessous des autres, puis par rapport :

- aux marges internes du conteneur.
- aux marges externes des éléments.

Avec ceci, on peut "déplacer" un élément dans son conteneur, et le contenu d'un élément au sein de cet élément.

Mais on ne peut pas placer deux blocs l'un à côté de l'autre !

Positionnement et CSS

Positionnement

fichier html :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <title> des tests </title>
    <link rel="stylesheet" href="flux.css" type="text/css" />
  </head>
  <body>
    <div id="entete">      entete          </div>
    <div id="navigation"> barre de navigation  </div>
    <div id="milieu">      Ici le milieu
      <div id="menu">     le menu      </div>
      <div id="corps">   le corps du texte</div>
    </div>
    <div id="footer">     un pied de page  </div>
  </body>
</html>
```

Positionnement et CSS

Positionnement

Et une css quasi vide :

```
body{
  color: #000;
  background-color: #800;
}

#entete{      background-color : #080; }

#navigation{  background-color : #008;}

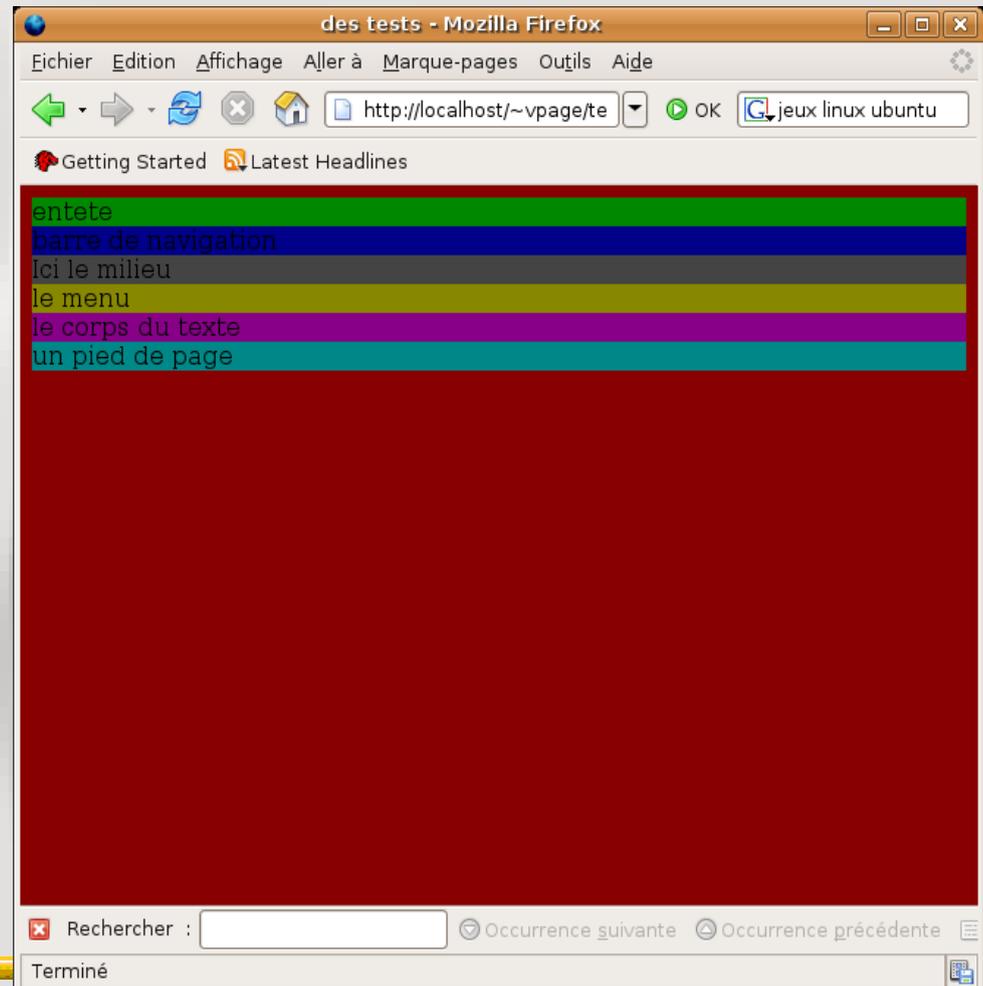
#milieu{      background-color : #444;}

#menu{        background-color : #880;}

#corps{       background-color : #808;}

#footer{      background-color : #088;}
```

Pour ce résultat :



Positionnement et CSS

Positionnement : marges et centrage

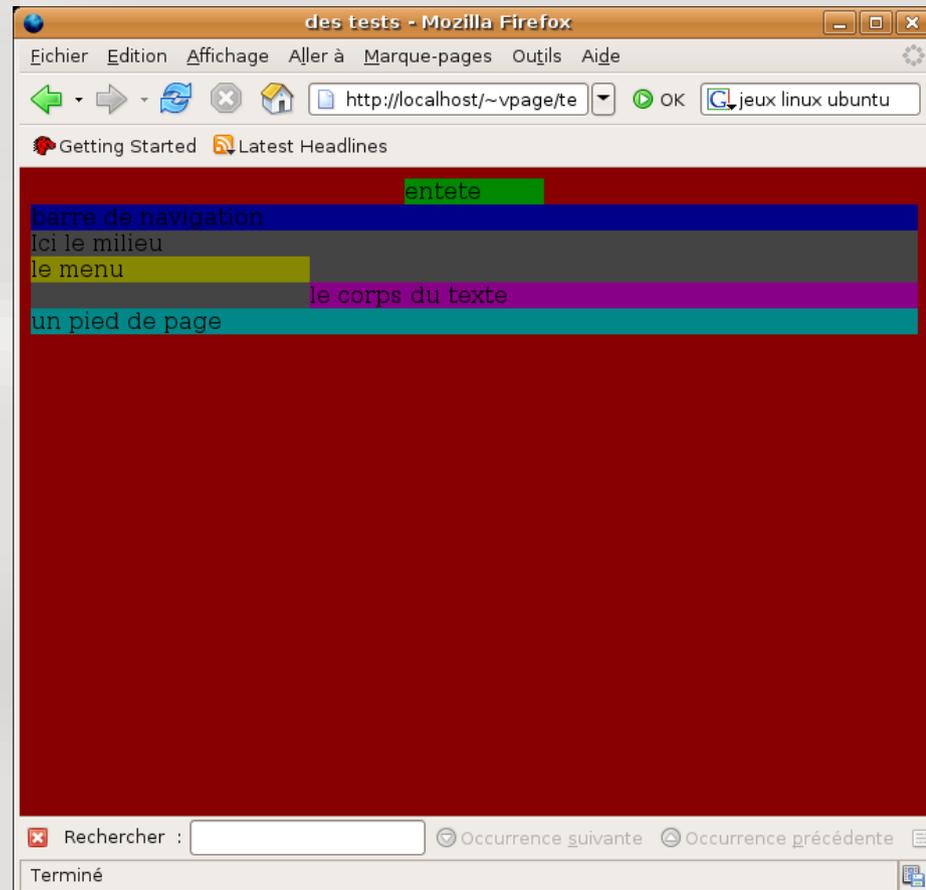
on rajoute les lignes suivantes

```
#entete{      width : 100px;
              margin : auto;
}

#menu{       width : 200px;}

#corps{     margin-left:200px;}
```

Pour ce résultat :



Positionnement et CSS

Positionnement flottant

Déplacer un élément a gauche ou a droite dans son parent.

Le reste du contenu du parent se positionne autour de l'élément déplacé.

Attention : Comme le float sort du flux courant, il n'est pas compté dans le calcul de la hauteur du conteneur, si celle-ci n'est pas spécifiée. Il peut donc "dépasser" en hauteur.

Récapitulatif :

** L'élément est d'abord placé normalement dans le flux. Donc si un autre élément bloc est placé avant lui, l'élément flottant se positionnera en dessous; et si un élément bloc le suit, ce dernier sera placé dessous. Bref, le comportement classique du positionnement en flux*

** Ensuite l'élément qui a reçu la propriété float est "poussé" à gauche ou à droite de son conteneur.*

** Les éléments qui le suivent au sein du conteneur se placent alors autour de lui.*

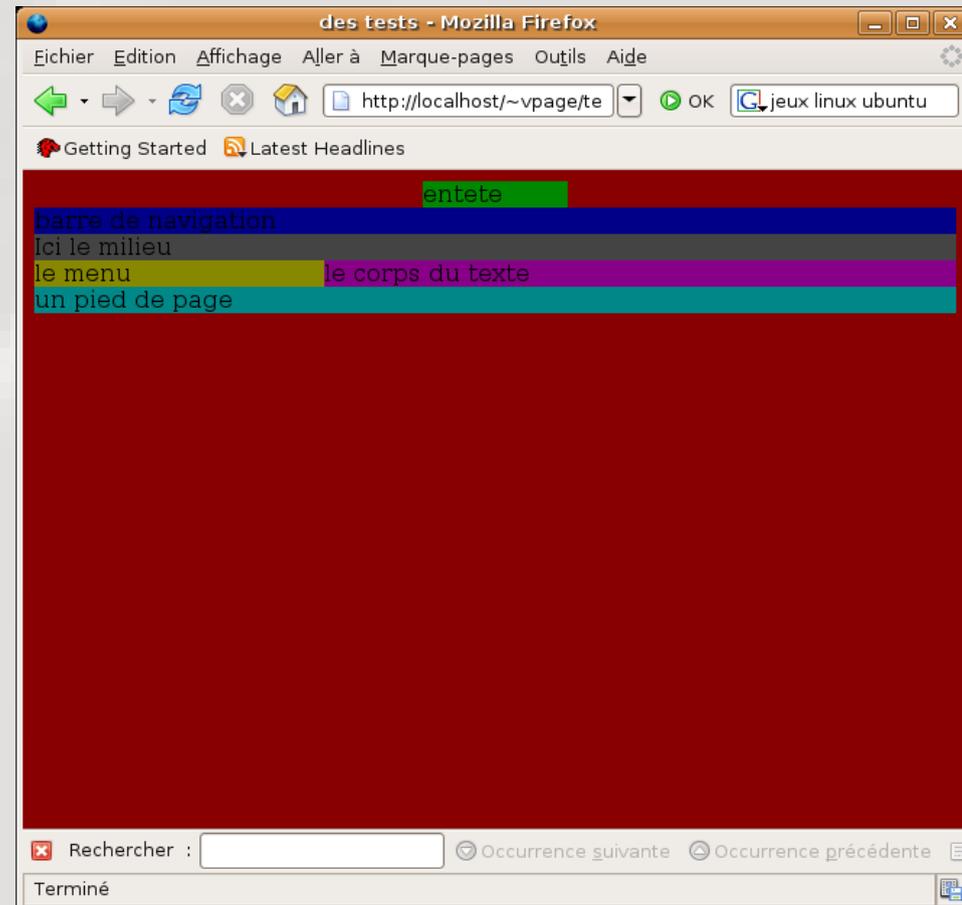
Positionnement et CSS

Positionnement float

on rajoute les lignes suivantes

```
#menu{ float:left;}
```

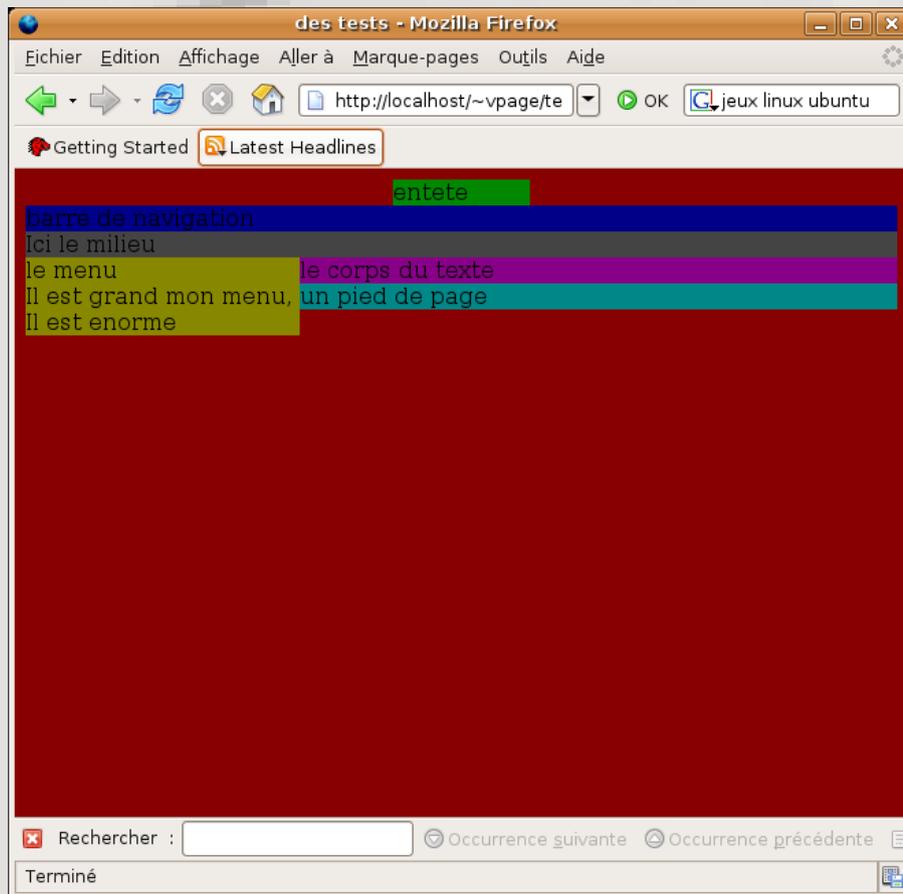
Pour ce résultat :



Positionnement et CSS

Positionnement : float

Mais si on allonge le menu, on a ceci :



La taille de la div de menu n'est pas prise en compte pour le calcul de la taille de la div du milieu...

Notez que le pied de page n'appartient pas a la div milieu....

Positionnement et CSS

Positionnement : float

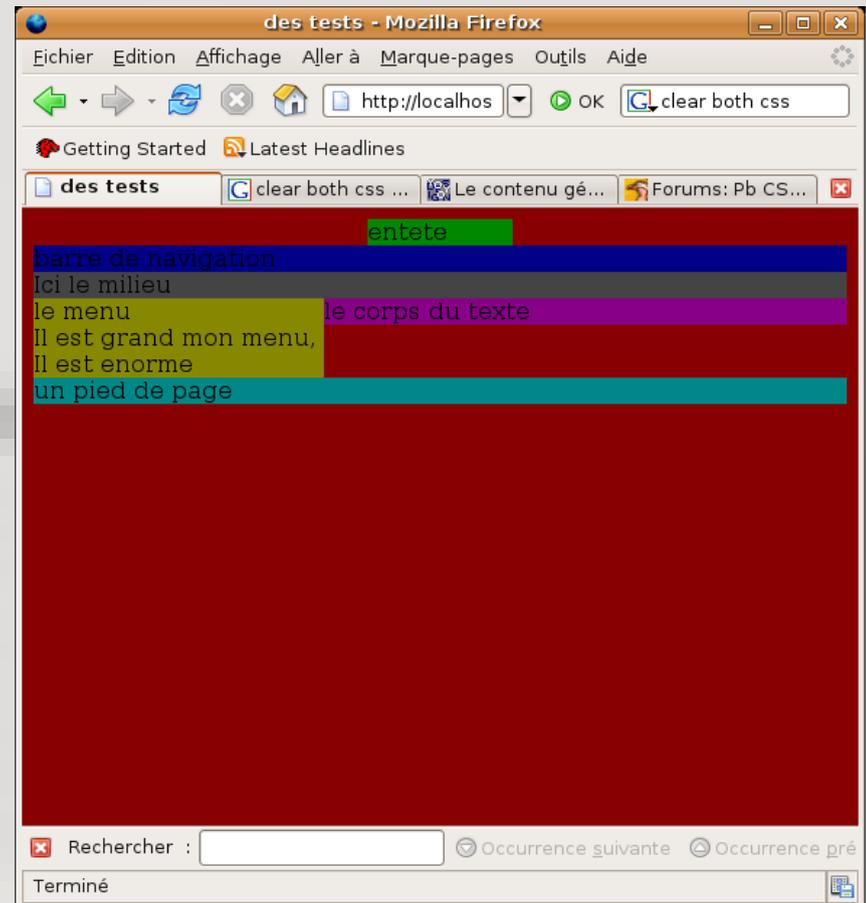
Le **clear** : oblige la boite a avoir un coté libre. La boite se place alors sous les autres boites flottantes floats du coté spécifié.

on rajoute les lignes suivantes

```
#footer{ clear:left;}
```

Notez le vide rouge entre le corps de texte et le footer : la div milieu ne s'étend pas jusqu'au pied de page !

On peut avoir clear:right ou clear:both



Positionnement et CSS

Positionnement absolu

Positionner un élément "en dur".

Pour spécifier la position d'un élément, on donne la position de son coin supérieur gauche. On peut aussi donner l'indiquer comme suit :

- haut et droite (+ largeur)

L'élément est retiré du flux. Son positionnement est donc le même quel que soit sa position dans le fichier html...

Positionnement et CSS

Positionnement absolu

Ici on repositionne quelques éléments en absolu.

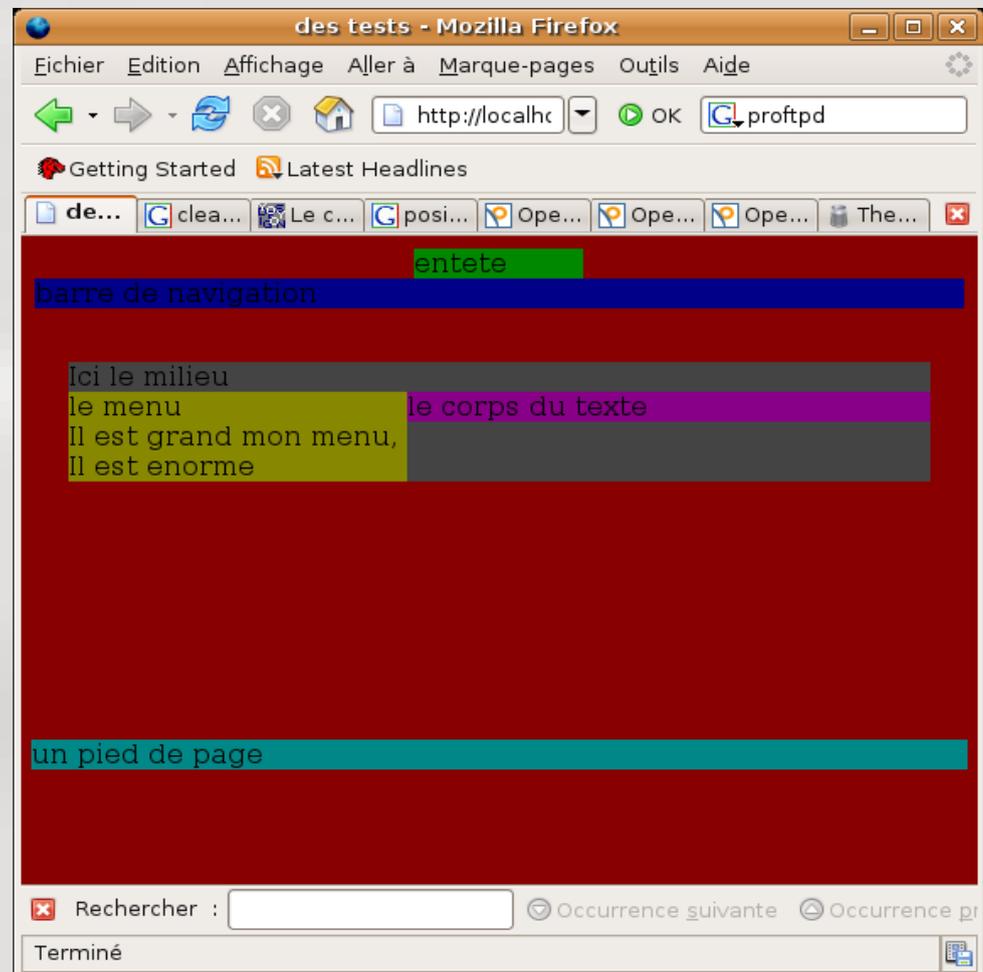
on rajoute les lignes suivantes

```
#milieu{  
    position:absolute;  
    top:5em;  
    left:5%;  
    right:5%;  
}
```

```
#footer{  
    position:absolute;  
    top:20em;  
    left:1%;  
    right:1%;  
}
```

Notez la taille de la div milieu.

Attention aux chevauchements...



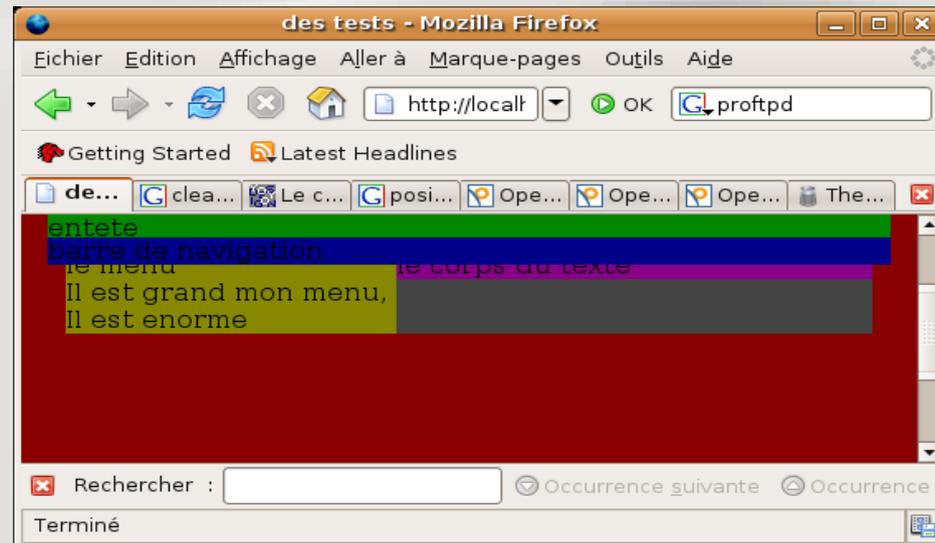
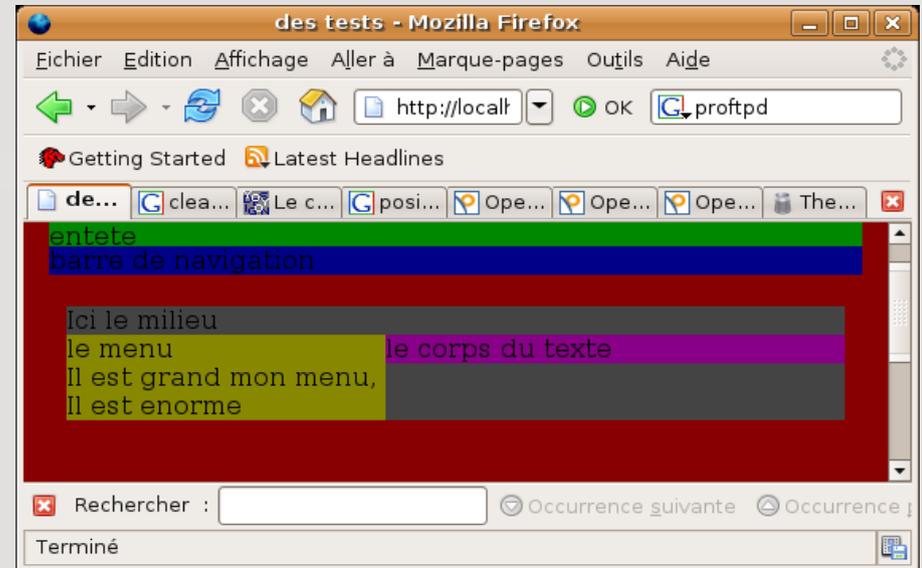
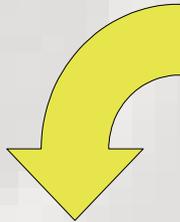
Positionnement et CSS

Positionnement fixe

Les éléments fixes ne bougent pas quelque soit la position du défilement dans la fenêtre du navigateur... Ici, la barre et l'entête sont fixes.

```
#entete#entete{  
    position:fixed;  
    top:0;  
    left:1em;  
    right:1em;  
    z-index:2;  
    margin : auto;  
    background-color : #080;  
}
```

```
#navigation{  
    position:fixed;  
    top:1em;  
    left:1em;  
    right:1em;  
    z-index:2;  
    background-color : #008;  
}
```



Positionnement et CSS

Positionnement Relatif

Non abordé ici !

Positionnement et CSS

Quelques remarques

- Une balise bloc peut être traitée comme une balise inline et inversement avec la propriété display (:inline ou :bloc)
- Attention à la fusion de marges.
- Certains navigateurs sont récalcitrants :
 - évaluation des tailles, comportement fixed, floats, ils n'interprètent pas les choses de la même façon...
 - Trouvez un gabarit éprouvé (alsacreation)
 - Testez vos sites avec plusieurs navigateurs, plusieurs versions des navigateurs.....

Pratiquez !

PHP

(Hypertext Pre-Processor)

Qu'est ce que PHP ?

Un langage de scripts généraliste (perl ?)

Pour nous : un langage interprété par Apache pour générer une page HTML standard qui est transmise au client.

Il s'agit alors de script exécuté « **coté serveur** ».

Pour nous : php4

Similaire : ASP (Active Server Pages), de chez Microsoft.

Syntaxe proche langage C.

PHP

(Hypertext Pre-Processor)

Qu'est ce que PHP ?

Un langage de scripts généraliste (perl ?)

Pour nous : un langage interprété par Apache pour générer une page HTML standard qui est transmise au client.

Il s'agit alors de script exécuté « **coté serveur** ».

Pour nous : php4

Similaire : ASP (Active Server Pages), de chez Microsoft.

Syntaxe proche langage C.

PHP

(Hypertext Pre-Processor)

- Différence avec des scripts « cgi » :
 - Script cgi : exécutable lancé par apache pour générer du HTML.
 - Php : code html dans lequel on insère des directives php. (gain de temps...)
- Coté client : réception du résultat, sans connaissance du code (sécurité)

PHP fonctionnement

Ecrire du PHP :

- Un fichier php porte l'extension .php
- C'est un fichier html classique contenant des balises pour indiquer quelles parties sont « en php »
- On passe du « mode texte » au « mode php » avec la balise `<?php ?>`

```
<?php  
echo "bonjour";  
echo "le monde";  
?>
```

PHP instructions

Php est très semblable au langage C :

- Instructions séparées par des virgules;
- Commentaires par // et /* ... */
- Beaucoup de mots clefs communs (if, for, while, return, switch, ==)

Variables PHP

- Déclaration explicite du type d'une variable non nécessaire. (type d'une variable déterminé par le contexte d'utilisation.)
- Forcer un type :

```
int settype ( string var , string type)
```
- Transtypage (cast) :

```
$mon_double = (double) mon_int;
```
- Obtenir le type d'une variable :

```
echo gettype($ma_variable);
```
- Vérifier le type d'une variable :

```
if is_int($ma_variable) {}
```
- Noms de variables sensibles à la casse.

Variables PHP

- Types scalaires

- Booléen
- Entier
- Nombre à virgule flottante
- Chaîne de caractères

```
$mon_bool=true;  
$mon_entier=0;  
$mon_float = 0.123;  
$ma_chaine = "on bitin";
```

- Types composés :

- Tableau
- Objet

```
$mon_tab = array(1,2,3);
```

Similaire : classes C++ (voir plus loin)

- Types spéciaux :

- Ressource
- Null

Choses bizarres (connexions ftp... voir plus loin)
Absence de valeur.

```
$var = NULL;
```

Variables PHP

Le relâchement sur le type des variables peut poser problème lors de comparaison : utiliser l'opérateur d'égalité

« === »

Chaines de char PHP

- Spécifications de chaînes :

- Guillemets simples : nom de variables non développés.

```
$chaine = 'toto $var \n';
```

- Guillemets doubles : noms de variables développés.

```
$chaine = "toto $var \n";
```

- Syntaxe HereDoc : comme guillemets doubles sans guillemets...

```
$chaine = <<<EOD bonjour,  
je m appelle "$nom"  
EOD;
```

Tableaux PHP

Définition :

- Un tableau php est une association ordonnée : chaque valeur est associée à une clef.
 - La clef est un entier ou une chaîne
 - La valeur peut être de tout type.

Cette implémentation permet de faire des dictionnaires, tables de hachage, des piles...

Tableaux PHP

Création de tableaux : la fonction array

```
$tab = array(15);  
$tab = array(12); // Ecrase le tableau précédent  
$a1 = array("un", "deux", "trois", "nous irons au bois");  
$a2 = array(1 => "un", "deux", "trois", "nous irons au bois");  
$a3 = array(1 => "un", 2 => "deux", 3 => "trois", 4 => "nous irons ...");
```

Tableaux PHP

Manipulation d'éléments de tableaux

```
$stab = array(15);  
$stab[]=12;           // On ajoute 12 à la fin du tableau (index 1)  
$stab[1]=24          // on remplace 12 par 24  
  
unset($arr[0]);      // supprime l'élément de clef 0  
unset($arr);        // supprime le tableau
```

Tableaux PHP

Afficher un tableau : mot clef « foreach »

```
<?php  
$tab = array('rouge','bleu','vert','jaune');  
  
foreach ( $tab 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? */  
?>
```

Tableaux PHP

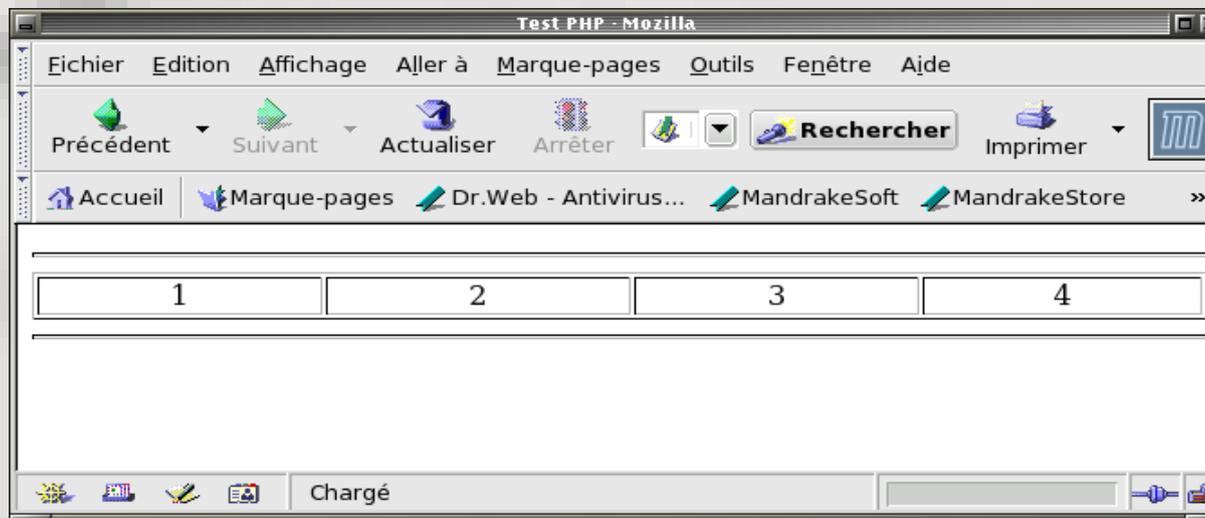
modifier tous les éléments d'un tableau :

```
<?php  
foreach( $stab as $cle => $couleur )  
{  
    // ne marche pas  
    //$couleur = strtoupper($couleur);  
    //marche :  
    $couleurs[$cle] = strtoupper($couleur);  
}  
?>
```

Tableaux PHP

Exercices :

- *Ecrire une page html qui affiche la page suivante :*



- *Ecrire une page php qui affiche un tableau 1D sous la forme précédente.*
- *Ecrire une page php qui fasse la même chose pour des tableau 2D.*

Tableaux PHP

Fonctions utiles de manipulations de tableaux :

- `reset($array)` Remet le pointeur interne de tableau au début.
- `pos($array)` Retourne la valeur de l'élément courant d'un tableau.
- `key($array)` Retourne l'indice de l'élément courant d'un tableau.
- `current($array)` Retourne la valeur de l'élément courant d'un tableau.
- `next($array)` Avance le pointeur interne d'un tableau.
- `prev($array)` Recule le pointeur courant de tableau.
- `end($array)` Positionne le pointeur de tableau en fin de tableau.
- `each($array)` Retourne la paire clé/valeur courante et avance le pointeur de tableau.

Chacune de ces fonctions renvoie false en cas de problème.

Tableaux PHP

Le mot-clef `list` : Transforme une liste de variables en tableau.

```
$info = array('rhum', 'transparent', 'alcool');
```

```
// Liste de toutes les variables
```

```
list($drink, $color, $power) = $info;
```

```
echo "$drink est $color et l'$power le rend particulier.\n";
```

```
// Liste de certaines variables
```

```
list($drink, , $power) = $info;
```

```
echo "$drink contient de l'$power.\n";
```

```
// Ou bien, n'utilisons que le troisième
```

```
list( , , $power) = $info;
```

```
echo "J'ai besoin d'$power!\n";
```

Tableaux PHP

Utilisation de list : parcours de tableaux

```
$info = array('rhum', 'transparent', 'alcool');  
  
while (list ($key, $val) = each ($info))  
{  
    echo "$key => $val \n";  
}
```

Tableaux PHP

Fonctions et opérateurs utiles pour les tableaux :

- Opérateur + Ajoute deux tableaux. (clefs égales..)
- `array_merge($stab1, $stab2...)` Ajoute deux tableaux (clefs égales...)
- `count($stab)` Compte les éléments d'un tableau.
- `array_keys($stab)` Retourne toutes les clés d'un tableau
- `array_values($stab)` Retourne les valeurs d'un tableau
- `array_key_exists($key, $stab)` Regarde si une clef existe.
- `array_search($val,$stab)` Recherche la clé associée à une valeur.
- `array_flip($stab) :` Retourne un tableau : clefs <-> valeurs

Chacune de ces fonctions renvoie false en cas de problème.

Tableaux PHP

Exemple de fusions de tableaux :

```
<?  
+ et merge...  
?>
```

Tableaux PHP

Fonctions utiles de tris de tableaux :

- `sort ($array)` Trie un tableau par valeurs croissantes.
- `rsort ($array)` Trie un tableau par valeurs décroissantes.
- `ksort($array)` Trie un tableau par valeurs de clefs croissantes.
- `krsort($array)` Trie un tableau par valeurs de clefs décroissantes.
- `shuffle ($array)` Mélange les éléments d'un tableau.

Chacune de ces fonctions renvoie false en cas de problème.

Fonctions PHP

Définir et appeler une fonction

```
function somme ($arg1, $arg2,$arg3)
{
    print "$arg1";
    $retval = $arg1+$arg2+$arg3;
    return $retval;
}
$toto=5;
$resu = somme(1,2,$toto);
print "$resu";
```

Fonctions PHP

Passage des arguments d'une fonction

Passage par valeurs : Une fonction ne modifie pas ses arguments.

```
function bidon ($arg1)
{
    $arg1=0;
}

$toto=15;
bidon($toto);
echo "toto = $toto";
```

Fonctions PHP

Passage de paramètres par référence

Pour pouvoir modifier une variable tout le temps

```
<?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.'  
?>
```

Fonctions PHP

Passage de paramètres par référence

Pour pouvoir modifier une variable 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.'  
?>
```

Fonctions PHP

paramètres par défaut des fonctions

```
<?php  
function servir_apero ($type = "ricard")  
{  
    return "Servir un verre de $type.\n";  
}  
  
echo servir_apero();  
echo servir_apero("whisky");  
?>
```



Valeurs par défaut : constantes

Paramètres avec valeurs par défaut sont passés après les autres

Fonctions PHP

Nombre de paramètres variables

- Passer un tableau de variables (le plus simple)
- Utiliser les fonctions suivantes (en TP).
 - `func_num_args` ,
 - `func_get_arg`
 - `func_get_args` .

Inclusion de fichiers PHP

Pour inclure des bibliothèques de fonctions :

- Mots clefs
 - « require » « require_once »
« include » « include_once »
- Ne se différencient que par leur gestion d'erreur :
- Inclut et exécute un fichier php.
 - « require » génère une erreur (le script s'interrompt)
 - « include » génère une alerte (le script continue).

Inclusion de fichiers PHP

Exemple

```
<?php  
// Avec une chaine constante :  
require 'prepend.php';  
  
// Avec une variable :  
require $somefile;  
  
// En pseudo fonction :  
require ('unfichier.txt');  
?>
```

Variables prédéfinies PHP

Liste des variables prédéfinies (auto globales / super globales)

```
$GLOBALS // Ref sur chaque var globales. Clef = nom de variable.  
$_SERVER // variables fournies par httpd ou relatives au script courant.  
$_GET // variables fournies par http en méthode get.  
$_POST // variables fournies par http en méthode post.  
$_COOKIE // variables fournies par http dans les cookies.  
$_FILES // variables fournies par http suite à un chargement de fichier.  
$_ENV // variables fournies par l'environnement.  
$_REQUEST // variables fournies par tout mécanisme d'entrée.  
$_SESSION // variables enregistrée dans la session.
```

Connaître certaines variables prédéfinies: `phpinfo()`;

Portée des variables en PHP

- Une variable utilisée dans une fonction est locale !
- La portée d'une variable concerne la totalité du script.
- Pour récupérer une variable globale dans une fonction :
 - utiliser le mot clef « global ».
 - utiliser le tableau \$globals
- Variables statiques : comme en C

Formulaires HTML

Pourquoi faire ?

On utilise des formulaires dans le cas suivant :

- L'utilisateur doit fournir des infos au serveur.
- Le serveur doit traiter ces infos.
- Le serveur renvoie une réponse au client.

Un formulaire est mis en forme en HTML.

Formulaires HTML

Début d'explication

Les informations que fournit le formulaire doivent être identifiées pour le traitement. Elles portent un nom.

- Le contenu de l'information dépend du type d'information.
- Le serveur traite et renvoie une réponse au client.

Un formulaire est mis en forme en HTML.

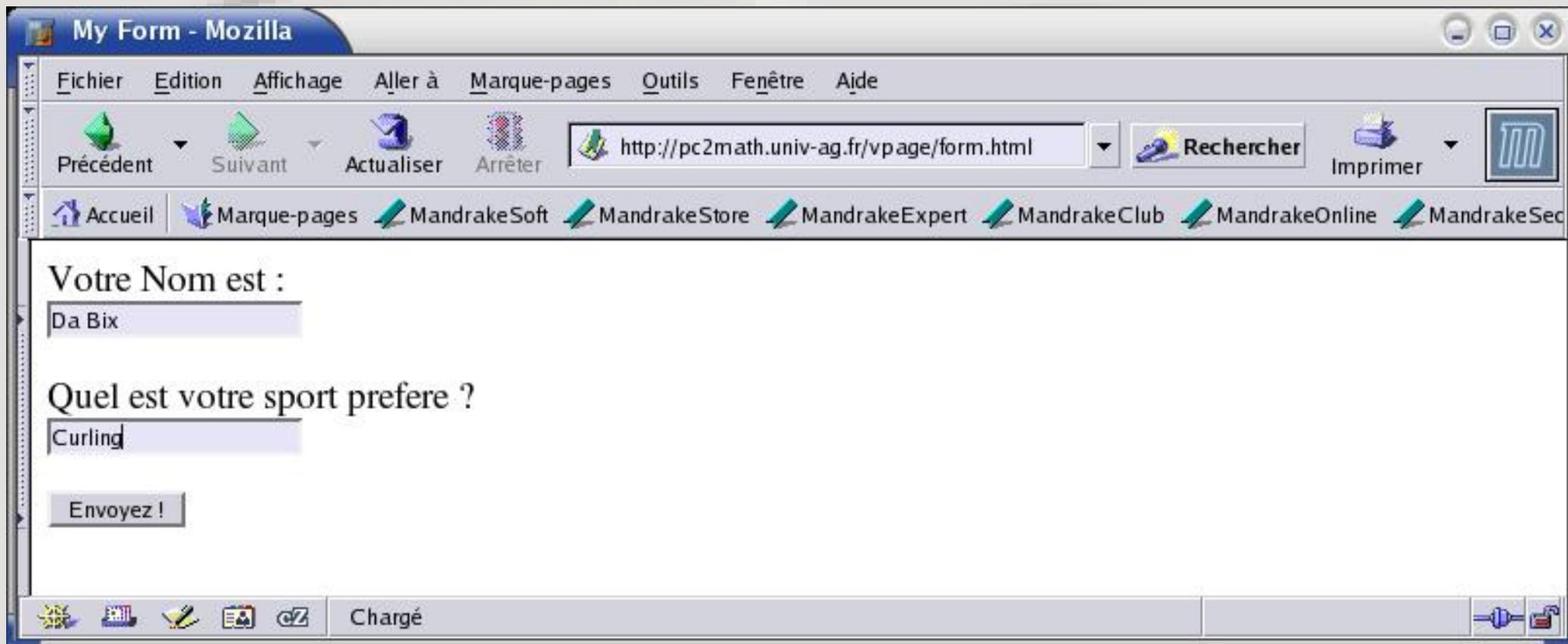
Formulaires HTML

Ecriture d'un formulaire HTML :

```
<form action= "mon_script.php method= "post">  
Votre Nom est : <br>  
<input type="text" name="YourName"> <p>  
  
Quel est votre sport prefere ? <br>  
<input type="text" name="FavoriteSport"> <p>  
  
<input type="submit" name="submit" value="Envoyez !">  
</form>
```

Formulaires HTML

Résultat du formulaire précédent :



Formulaires HTML

Les `<inputs>` qui existent

Ils se distinguent par l'attribut type

- `text` : une ligne de texte
- `password` : idem mais caché.
- `hidden` : invisible (pour passer des variables).
- `checkbox` : selection multiple.
- `radio` : selection d'une possibilité parmi X.
- `submit` : bouton d'envoi de formulaire
- `reset` : bouton de remise dans l'état par défaut
- `file` : saisie d'un nom de fichier (et bouton parcourir...)

Formulaires HTML

Autres balises de formulaires

Attention, ce sont des balises

- `<select></select>` : liste de sélection déroulante.
 - `<option></option>` : options de liste déroulante.
- `<textarea></textarea>` : zone de texte multilignes...
- `<button></button>` : un bouton.

Formulaires HTML

Traitement du formulaire : Script PHP

- Le formulaire est envoyé à un script défini par l'attribut action de la balise <form> (ici mon_script.php)
- Deux méthodes d'envoi : GET et POST défini par l'attribut method de la balise <form> (ici post)
- Le contenu du formulaire est accessible via une variable meta-globale (ici \$_POST)
- Chaque champ du formulaire crée une variable portant son nom (attention aux checkbox...)
- Le résultat du traitement est renvoyé au client.

Utiliser `print_r($_POST)` pour debugger...

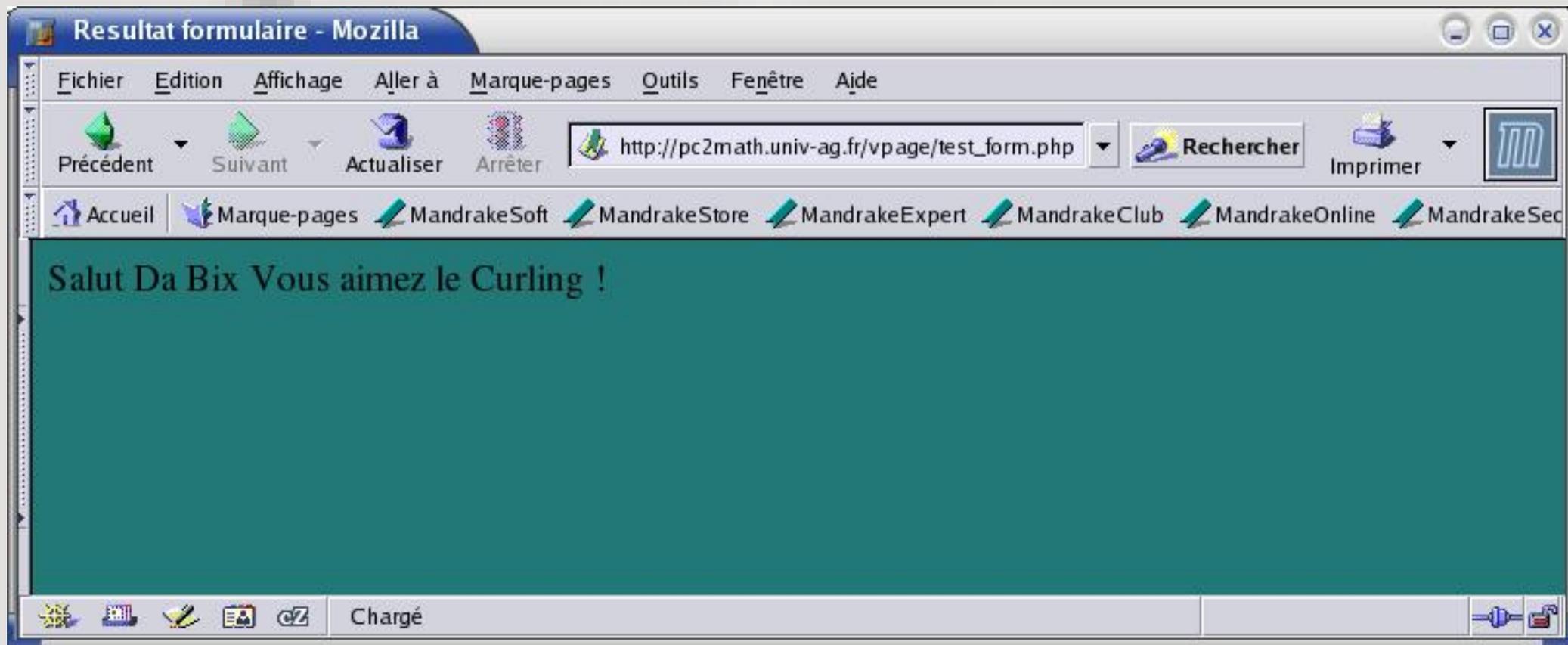
Formulaires HTML

Traitement du formulaire : Script PHP

```
<?php  
$name = $_POST["YourName"];  
$sport = $_POST["FavoriteSport"];  
printf ("Salut %s\n", $name);  
printf ("Vous aimez le %s !\n", $sport);  
?>
```

Formulaires HTML

Résultat du traitement du formulaire précédent :



Formulaires HTML

Méthode Post et Méthode Get

- Méthode GET
 - Méthode par défaut.
 - Le formulaire est passé par ajout à l'url.
 - La requête a la forme : "GET /doc.html?var1=val1&var2=val2.... HTTP/1.0"
 - Le script php peut les récupérer avec la méta-globale `$_GET`
 - Bookmarkable.
 - Petits volumes de données.

Formulaires HTML

Méthode Post et Méthode Get

- Méthode POST
 - Le formulaire est passé après l'url.
 - Les données sont transmises sur l'entrée standard du script.
 - Un script php peut les récupérer avec `$_POST`
 - Non bookmarkables
 - Petits ou gros volumes de données.

Sessions et Cookies

But : Conserver des informations lors de la navigation d'un client sur le serveur.

Exemples d'utilisation :

- Identification (accès contrôlé)
- Panier d'achat
- Affichage adapté au contexte.

En effet : http fonctionne en Mode non connecté.

Cookies

- Qu'est ce ?
 - Un fichier créé par le navigateur sur le poste client, à la demande du serveur.
 - Sur la mdk9.2, les cookies de mozilla sont tous placés dans un fichier cookies.txt.
- Fonctionnement :
 - Le navigateur qui se connecte à un domaine envoie au serveur les cookies rattachés au domaine.
 - Le cookie a une durée de vie limitée. Il est supprimé lorsqu'il est périmé.

Cookies

- Les Cookies sont créés coté client :
 - Problème d'acceptation.
- Php gère les cookies de façon simple.
 - Les cookies sont envoyés comme en-tête du document (avant tout autre chose).
- Envoyer un cookie :

bool setcookie (string name, string value, int expire, string path, string domain, int secure)

- Regarder les cookies : `$_COOKIE`



Attention : cookies envoyés avant toute chose (doctype).

Cookies

Exemple :

```
<?php  
$duree = 24*60*60; // 1 journee en secondes !  
$value=$_SERVER["REMOTE_ADDR"];  
setcookie("moncookie",$value,time()+$duree,"/","127.0.0.1");  
?>  
  
<html> </body>  
  
vous venez de recevoir un cookie <br>  
<a href="lire_cookie.php"> Lire le cookie </a>  
  
</body></html>
```

Cookies

Si vous avez besoin des cookies :

- vérifiez que vous pouvez les utiliser : Envoyez en un et regardez si il est reçu.

Sessions

- Qu'est ce que c'est :
 - Un identifiant SID associé à un navigateur accédant à une page.
 - Le SID est défini par le serveur.
 - Coté serveur, on stocke pour chaque SID un ensemble de variables.
- Mode de passage du SID :
 - Par url. Peu sécurisé.
 - Par cookie. Risque de non acceptation des cookies.
 - Choix du mode de passage : php.ini
- Accès aux variables de sessions : `$_SESSION`

Sessions

- Utiliser les sessions :

```
<?php  
session_start();  
if (!isset($_SESSION['compteur']))  
{ $_SESSION['compteur'] = 0; }  
else  
{ $_SESSION['compteur']++; }  
?>
```

Cookies et sessions

- Authentification sécurisée (ma banque) :
 - Authentification par session.
 - limite les transmissions.
 - Passage du SID par cookie.
 - pas par l'url.
 - Le cookie à une durée de vie de 0
 - détruit quand on ferme le navigateur.
 - https : tous les échanges de données sont cryptés.
 - l'url passe quand même en clair

Manipulation de fichiers en PHP

Script exécuté par le serveur : les droits sur les fichiers sont ceux d'apache ou nobody.

- Obtenir des informations sur les fichiers locaux
 - `file_exists`, `is_dir`, `is_executable`, `is_file`, `is_link`, `is_readable`, `is_writable`
 - `stat`, `lstat`, `readlink`
 - `fileowner`, `filesize`, `filegroup`, `fileinode`, `fileperms`
 - `fileatime`, `filemtime`
 - `disk_free_space`, `disk_total_space`
 - `basename`, `dirname`,

Manipulation de fichiers en PHP

Script exécuté par le serveur : les droits sur les fichiers sont ceux d'apache ou nobody.

- Quelques fonctions bash-like :
 - Créer : touch, mkdir, link, symlink, copy, tempnam
 - Déplacer, effacer : rename, rmdir
 - Modifier les droits : umask, chmod, chmod, chgroup

Manipulation de fichiers en PHP

Quelques fonctions C-like

- les « pointeurs de fichiers » utilisés sont de type *resource*.
 - Ouverture fermeture : fopen, fclose, feof, fflush
 - Répertoires : opendir, readdir;
 - Déplacement : rewind, fseek
 - Ascii : fscanf, fprintf, fgets, fgetc, fgetss (sécu tags)
 - Binaire : fread, fwrite.

Manipulation de fichiers en PHP

Quelques fonctions typiques php

- glob : Trouve les noms de fichiers correspondant à une règle.

```
foreach (glob("*.txt") as $filename) {echo $filename;}
```

- file : Met le contenu d'un fichier dans un tableau

```
foreach (file ("toto.txt", "./") as $ligne) {echo $ligne;}
```

- file_get_contents : idem file mais dans une chaîne.

```
echo file ("toto.txt", "./");
```

Fichiers PHP

Afficher le contenu d'un répertoire

```
<?php  
$handle=opendir('.');  
echo "Pointeur de dossier: $handle <br>";  
echo "Fichiers: <br>";  
while ($file = readdir($handle))  
{  
    echo "$file <br>";  
}  
closedir($handle);  
?>
```

Exemples PHP

Comment faire :

- Une page affichant toutes les images d'un répertoire.
- Une page affichant
 - un tas d'aperçus d'images.
 - Liées aux images.
 - commentaire pour chaque aperçu.
- Comment faire pour que seules certaines personnes ne puissent voir ces images. Images non confidentielles, juste privées.

Php – Postgress

Fonctions utiles

- `pg_connect` : se connecter sur un serveur postgress
- `pg_connection_status` :
- `pg_query` : faire une requete.
- `pg_fetch_array` : Lit une ligne de resultat dans un tableau.
- `pg_fetch_row()` : Lit une ligne dans un tableau (indices colonnes)
- `pg_close` : se déconnecter

Exemple PHP – Postgress

Exemples de connections

```
<?php  
$dbconn = pg_connect("dbname=marie");  
$dbconn2 = pg_connect("host=localhost port=5432 dbname=marie");  
$dbconn3 = pg_connect ("host=sheep port=5432 dbname=marie  
user=mouton password=baaaa");  
$conn_string = "host=sheep port=5432 dbname=test user=lamb  
password=bar";  
?>
```

Exemple PhP – Postgress

Exemples de requete

```
$db = pg_connect("...");  
$query = "SELECT * FROM friends";  
$result = pg_query($db, $query);  
  
$row=0;  
$numrows = pg_numrows($result);  
do {  
    $myrow = pg_fetch_row ($result,$row);  
    echo $myrow[0].$myrow[1]. $myrow[2];  
    $row++;  
} while ($row < $numrows);
```

Génération d'images en lignes

Faut remplir ici... bibliothèque GD.

Sécurité

Considérations de sécurité : Rappels

- Droits des scripts.
- Conservation de renseignements (cookies/session)
- Moyen de se loguer sur un SGBDR.
- Sécurisation des échanges client / serveur.