

DHTML ou le Html dynamique

1. Définition du DHTML

1.1 Introduction ou ... au risque de vous décevoir

Ne cherchez pas de syntaxe ou de références DHTML, vous n'en trouverez pas. Le DHTML n'est pas un langage de balises, de scripts ou de programmation. Le DHTML n'est pas une quelconque spécification. Il n'est même pas réellement une technologie! Tout simplement,

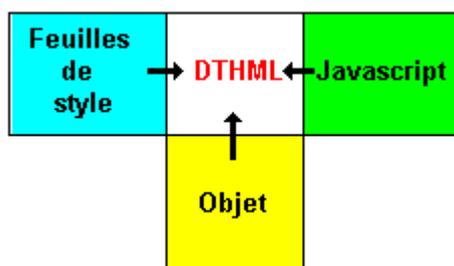
**le DHTML ou Html dynamique est, et reste, du Html
où le concepteur a mis l'accent sur les animations.**

A l'heure de l'importance croissante de l'impact visuel des sites Internet, le Html dit dynamique est un "plus" incontestable et génial sous certains aspects. Mais il faut raison garder! Le DHTML n'est pas le "futur de l'Internet" comme le prétendent certains mais, à n'en pas douter, il y aura certainement une bonne place.

Bien que peut-être excessifs, je suis même tenté de reproduire les mots de Sally Khudairi [à vos souhaits!] du World Wide Web Consortium (W3C) : "DHTML is nothing more than a marketing term for generic animation and manipulation of multimedia events" (DHTML n'est rien d'autre qu'un terme de marketing pour des animations et des effets multimédia).

1.2. Définition

Le DHTML se définit comme un (savant) mélange de trois développements de la publication sur Internet : les feuilles de style (CSS1), les langages de script, principalement le Javascript, ainsi que les objets et leur positionnement en vue de produire des pages dynamiques et interactives.



Ajoutons car cela n'apparaît pas dans la définition :

- l'aspect dynamique réside dans le fait que la page Html peut être modifiée après le chargement de celle-ci par le navigateur.
- une volonté d'effectuer les animations et interactions côté client (c-à-d côté browser) sans faire appel aux ressources du serveur par des applets Java ou des ActiveX.
- cette technique DHTML a un aspect "high-tech" et est réservée aux versions 4.0 (ou plus) de Internet Explorer et Netscape.

1.3. Les feuilles de style

Popularisée par Microsoft avec Internet Explorer 3.0 (bien qu'elles existaient déjà sous d'autres systèmes), les feuilles de style [Cascading Style Sheets] constituent un réel développement dans la conception de pages Web, en séparant la mise en forme du contenu, en proposant des possibilités jusque là inédites en Html strict , etc..

Le concept des feuilles de style est maintenant un standard depuis son officialisation par le World Wide Web Consortium sous la norme CSS level 1 et son intégration dans le Html 4.0. Et les versions 4.0 de Microsoft Explorer et Netscape Navigator reprennent déjà largement les feuilles de style.

Pour une étude plus poussée, les feuilles de style sont largement abordées dans ce site (www.ccim.be/ccim328/htmlplus/idxstyle.htm).

1.4. Les langages de scripts (surtout Javascript)

Introduit par Netscape avec Navigator 2.0, Javascript a bien mûri au fil des versions 3.0 et 4.0 des browsers de la marque (Javascript 1.1, Javascript 1.2). Enfin repris efficacement par Microsoft sous Explorer 4.0, ce langage de scripts s'est définitivement imposé comme un élément incontournable dans l'élaboration de sites Internet.

Javascript permet de programmer et d'exécuter, côté client c-à-d côté browser et donc sans faire appel à des ressources extérieures, de petites applications à l'intérieur de pages Html. Sans être un véritable langage de programmation, Javascript en possède toutes les fonctionnalités ou presque).

VBScript fait également partie des langages de scripts qui peuvent être utilisés dans le DHTML. Mais propriétaire Microsoft, il n'est que peu utilisé en DHTML au détriment de Javascript.

Javascript est aussi un standard Internet depuis la norme ECMA Script. En outre, le Html 4.0 le reprend largement.

Pour vous guider dans l'apprentissage de ces deux langages de scripts, "Apprendre le Javascript" (www.ccim.be/ccim328/js/index.htm) et "Apprendre le VBScript" (www.ccim.be/ccim328/vb/index.htm) du même auteur, sont à votre disposition.

1.5. Les objets et leur positionnement

L'héritage du Html, à l'origine conçu pour faire transiter du texte structuré, est parfois pesant. Positionner précisément, au pixel près, un texte ou une image, tourne vite au cauchemar. De là est née l'idée d'introduire une sorte de cadre, de rectangle dans lequel on pourrait "encapsuler" des "objets" comme précisément du texte, des images ou tout autre élément.

Comme s'est souvent le cas dans l'histoire de la publication sur Internet, nos deux compères Microsoft et Netscape, se sont attelés à la tâche (sans attendre un quelconque standard) et ce en suivant des modèles différents pour cette insertion d'objets dans les documents Html.

Microsoft fournit un accès aux éléments des pages Html par "ses" feuilles de style (CSS-P). Netscape a choisi un modèle d'insertion d'éléments qui lui est spécifique par le concept de "layer" et sa collection de balises. On se retrouve donc avec deux technologies distinctes pour une même approche.

Le W3C tente de réunir de réunir les deux précédentes technologies pour les faire fonctionner ensemble. C'est le DOM [Document Objet Model]. Cette tentative est encore à ce jour (septembre 98), un document de travail et l'on reste donc encore dans l'attente d'un véritable standard.

Cette lacune ouvre la porte aux problèmes de compatibilité pour les applications DHTML selon quelles soient traitées par Netscape 4.0 ou par Explorer 4.0.

1.6. Déceptions

Si visuellement et techniquement, le DHTML est quelque chose de tout à fait génial, on ne peut qu'être un peu déçu :

- par son manque (actuel?) de compatibilité. Un DHTML qui fonctionne pour IE Explorer 4.0 ne tourne que très rarement sous Netscape 4.0. Si le Webmaster est gentil, il aura bien prévu une petite remarque du genre "Il vous faut le browser X". Mais je fais toujours partie de ceux qui n'ont pas de bol et emploient justement le browser Y.
- par sa complexité. Tout en tirant mon chapeau aux concepteurs de DHTML, le Html dynamique n'est pas du tout à la portée d'un débutant et n'est même plus à la portée d'un hobbyiste, fut-il averti. Il faut connaître

sous le bout des doigts, trois techniques : le Html, le Javascript et les feuille de style. Ce qui n'est pas à la portée du premier

- Webmaster venu. A vos cours de programmation... ou à votre "copier/coller" (en respectant le copyright).
- par son temps de chargement plus important bien qu'il ne soit en rien comparable avec la vitesse d'escargot des applets Java.
- par son risque réel de faire planter le browser et même la machine si l'application DHTML est mal écrite.

1.7. Espoirs

Mais terminons par une note positive car :

- le concept DHTML (Feuilles de style - Javascript - Objets) est bien né et n'est pas remis en cause par les actuels problèmes de compatibilité.
- le DHTML est encore jeune et les développeurs ne peuvent que, dans un avenir proche, trouver de nouvelles astuces pour améliorer cette compatibilité.
- après les versions 4.0, Microsoft et Netscape sortiront leurs versions 5.0... Et celles-ci intégreront (peut-être) une gestion commune des objets.

2. Positionner avec Netscape

par la technique des couches [layer]

2.1.La balise <LAYER>

Netscape propose depuis sa version 4.0, une technique pour positionner des éléments. C'est la technique des couches ou des "layers". On peut comparer ces layers aux transparents que l'on utilise avec les rétroprojecteurs.

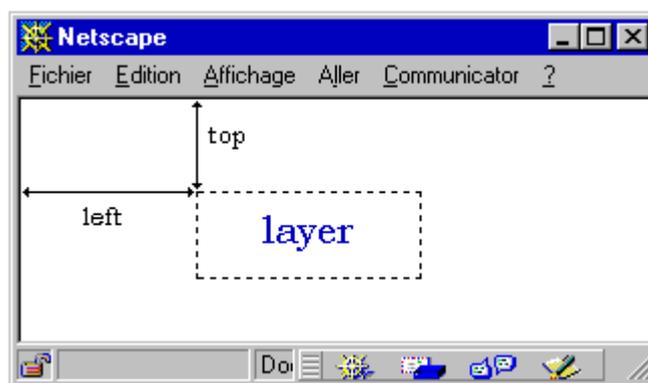
La syntaxe est :

```
<LAYER name="nom" left=x top=y> ... élément ... </LAYER>
```

La balise <LAYER> va demander au browser de réserver une sorte de petit rectangle - vide - ou une sorte de container dans lequel on va placer un élément, par exemple une image.

Il sera indispensable de donner un nom au "layer" lorsque plusieurs sont utilisés dans une même page ou un même document.

Les attributs "left" et "top" vont positionner cette zone par rapport au coin supérieur gauche de la fenêtre du browser. Top est la valeur en pixels du haut vers le bas et left, la valeur en pixels de la gauche vers la droite.



Le browser va afficher les layers et leur contenu quoiqu'il arrive! Idéal pour créer selon vos désirs des effets de superpositions. Idéal aussi pour avoir des superpositions inattendues...

La tag <LAYER> est un tag propriétaire de Netscape 4.0 qui n'a pas été repris par Microsoft. "LAYER" ne fonctionne donc pas sous Eplorer 4.0 et uniquement sous Netscape. Comme les feuilles de style qui permettent aussi de positionner des éléments, sont reprises et par Explorer 4.0 et par Netscape 4.0, cette dernière technique est recommandée pour des raisons de compatibilité.

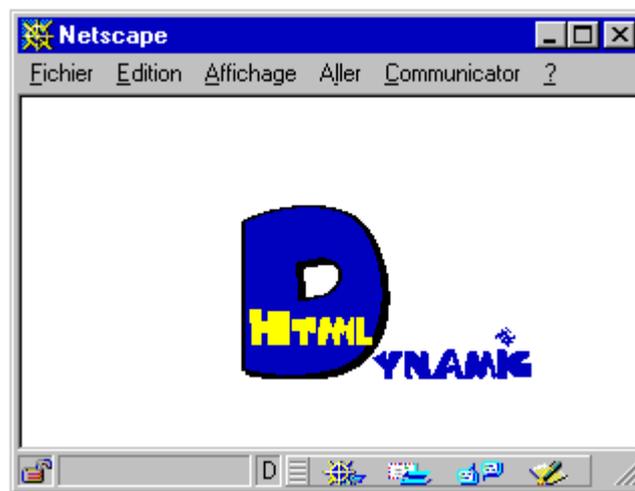
2.2. Exemple

Pourquoi des couches? Tout simplement (les layers étant affichés quoiqu'il arrive), lorsque vous prévoyez plusieurs layers au même endroit, le premier qui apparaît dans le code, sera affiché directement. Le layer suivant viendra se mettre au-dessus et ainsi de suite. Le premier layer est donc celui qui sera "au fond" de l'écran par rapport autres. En jouant avec la transparence des images gif, on peut ainsi créer des effets surprenants.

Nous allons superposer les deux images suivantes :



```
<LAYER name="layer1" left=100 top=50>  
<IMG SRC="dyna.gif" height=97 width=160>  
</LAYER>  
<LAYER name="layer2" left=110 top=75>  
<IMG SRC="html.gif" height=41 width=83>  
</LAYER>
```



La couleur verte a disparue car elle a la particularité gif d'être transparente. Le positionnement au pixel près ne se fait pas sans tâtonnement et il faut souvent plusieurs essais avant d'arriver au résultat souhaité.

Le mouvement quant à lui est obtenu avec une bonne dose de Javascript en faisant varier les valeurs des attributs "top" et "left".

3. Positionner avec Explorer

par la technique des feuilles de style (CSS)

3.1. L'approche par les feuilles de style

Quelques mois après la version 4.0 de Netscape, Microsoft sortait la version 4.0 de Explorer. Ce qui lui a permis d'aller plus avant dans le positionnement et la définition des objets.

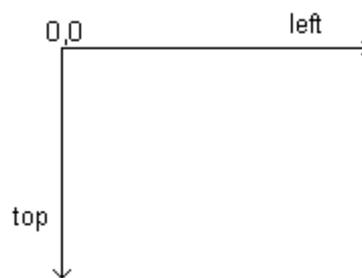
Microsoft a développé "son" concept de feuilles de style (Cascading Style Sheets ou CSS] qui permettait déjà de décomposer une pages Web en blocs avec les balises SPAN et DIV. Restait à pouvoir positionner ces blocs... C'est tout le développement du CSS-P ou du positionnement avec les feuilles de style.

A l'opposé de l'approche de Netscape avec une balise (la balise <LAYER>), les feuilles de style apportent un concept plus vaste, plus complet et plus puissant mais d'un apprentissage (un peu) plus complexe.

Pour les lecteurs qui prennent le site en marche, je rappelle que les feuilles de style sont largement abordées plus avant dans le site (www.ccim.be/ccim328/htmlplus/idxstyle.htm).

3.2. Position absolue ou relative

La position absolue {position: absolute} se détermine par rapport au coin supérieur gauche de la fenêtre du browser. Les coordonnées de ce point sont top = 0 et left = 0. Les coordonnées d'un point s'expriment en pixels, de haut en bas pour top et de gauche à droite pour left.

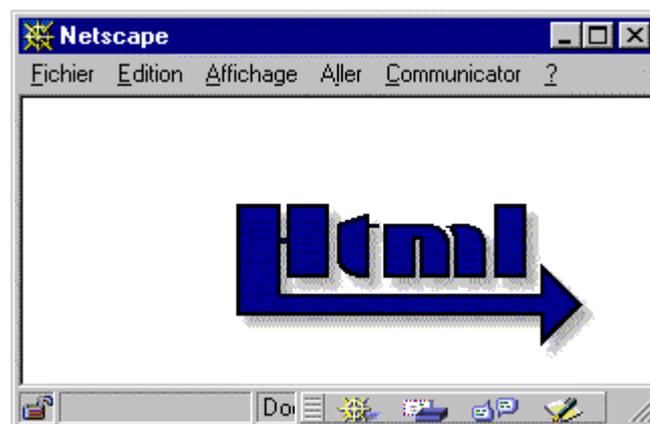


La position relative {position: relative} se détermine par rapport à d'autres éléments de la page, par exemple un élément du code Html.

3.3. Positionner une image

Plaçons l'image htmlplus.gif en position absolue à 50 pixels de haut de la fenêtre (top) et à 100 pixels à gauche (left). Les dimensions de l'image sont width=242 et height=84.

```
<HTML>
<BODY>
<span style="position: absolute; top: 50px; left: 100px; width: 242px; height: 84px;">
<IMG src="htmlplus.gif">
</span>
</BODY>
</HTML>
```



Spécifiez toujours les propriétés width et heigth avec les feuilles de style car par défaut, Netscape 4.0 et Explorer 4.0 ne réagissent pas de la même façon.

3.4. Superposer du texte sur une image

Reprenons l'image htmlplus.gif et on y superposera le nom de l'auteur de ce tutorial, au pixel près dans la barre qui souligne le terme Html.

```
<HTML>
<BODY>
<span style="position: absolute; top: 50px; left: 100px; width: 242px; height: 84px;">
<IMG src="htmlplus.gif">
</span>
<span style="position: absolute; top: 96px; left: 145px; color: yellow; font-size: x-small; font-weight:
bold;">
Van Lancker Luc
</span>
</BODY>
</HTML>
```



4. Le "Document Object Model"

4.1 La notion d'objet.

Conçu dans les années 1980, le Html (HyperText Markup Language) n'est finalement qu'un long fichier de texte auquel on a ajouté des balises (Markups) - prédéfinies et statiques - afin que les navigateurs puissent afficher ce texte dans une mise en forme ou une mise en page identique sous différentes plates-formes. Html n'a par essence aucune structuration et encore moins d'hierarchisation de ses composants.

La perception d'une page Html comme un ensemble d'éléments et qui plus est hiérarchisés, est une véritable révolution dans le domaine de la publication Html. Ces éléments sont aussi repris sous le vocable d'objets au sens des langages de programmation comme C ou C++.

A la mi-97, Netscape a entamé le sujet avec sa version 4.0 en créant une balise propriétaire <LAYER>. Explorer 4.0, qui est apparu quelques mois plus tard, a pu aller plus avant dans le concept objet en exploitant quant à lui, le concept des feuilles de style repris et développé par Microsoft.

Il devenait alors évident que ce concept d'objets devait être standardisé par le World Wide Web Consortium (W3C) avec comme objectif de dégager un modèle commun pour la définition des objets dans un document Html. Ce qui a donné naissance au Document Object Model repris sous l'abréviation de DOM.

4.2 Le Document Object Model ou DOM

Ce "Document Object Model" n'est encore à ce jour (septembre 98) qu'un document de travail et n'est donc toujours pas finalisé.

Si les principaux protagonistes, Netscape et Microsoft, sont unanimement d'accord sur la nécessité d'introduire un principe (commun) de la notion d'objets, leurs avis divergent complètement sur les techniques à mettre en oeuvre pour le réaliser.

Leurs principaux points de désaccord peuvent se résumer ainsi :

- Netscape 4.0, conçu plusieurs mois avant son concurrent, a une conception interne assurément moins élaborée et moins souple que Explorer 4.0.
- Netscape 4.0 n'a accès qu'à quelques éléments de la page alors que IE Explorer 4.0 a accès à tous les éléments de la page.
- A travers cette implémentation de la notion d'objet, Netscape défend (ou tente de récupérer) le concept de Java et Microsoft prône bien sûr ses ActiveX et autres Active Channels.
- Les versions 5.0 de Microsoft et de Netscape se profilent à l'horizon 99.

Constatant que dans l'état actuel des technologies les avis étaient inconciliables, le W3C a préféré jeter des bases plus solides pour l'avenir en rattachant cette notion commune d'objets au XML (eXtensible Markup Language) qui sera compatible avec les futures évolutions 5.0 des navigateurs.

4.3. Le XML en bref

Il faut bien l'avouer, le Html avec toutes ses balises rajoutées au fil des années, fait un peu désordre. Il subit en outre une demande énorme en termes de besoins en publication sur Internet auxquels il ne peut répondre comme les signes mathématiques, les notes de musique ou le braille.

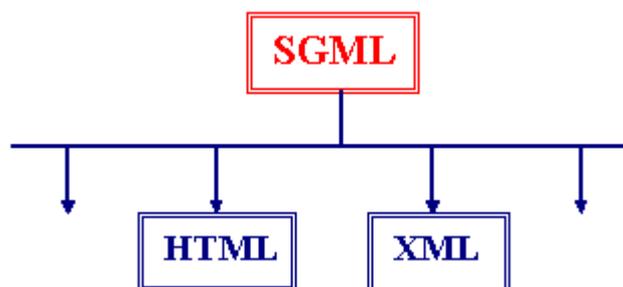
Le XML (eXtensible Markup Language) donne l'impression de repartir avec une feuille blanche. XML n'est en aucune façon, un Html++ mais un nouveau petit frère pour le Html car XML et Html ont la même mère c-à-d le langage SGML.

Pas simple tout cela...

D'abord le SGML (pour Standard Generalized Markup Language), est une norme internationale (ISO 8879), qui régit la structure et le contenu des différents types de documents électroniques. Le SGML est en quelque sorte un "langage-mère" utilisé pour décrire les milliers de documents électroniques que l'on retrouve dans les multiples domaines de l'activité humaine.

Le HTML (HyperText Markup Language) n'est que un de ces types de documents avec ses balises imposées et qui est utilisé pour les documents électroniques qui circulent sur le Web.

Avec le XML, on retourne aux sources du SGML en permettant au "designer" de définir son propre type de document et, côté spectaculaire de la chose, de créer ses propres balises (voir le X de eXtensible).



Tant Microsoft que Netscape annoncent que le XML sera disponible avec les versions 5.0 de leur browser.

Nous ne manquerons pas de revenir plus en détail sur le XML, plus tard dans l'élaboration de ce site consacré à la maîtrise du Html.

4.4 En conclusion

Pour en revenir à la définition d'objets, chaque fabricant de browser pourra avec le Xml implémenter des objets, selon sa propre perception et selon sa propre technologie, définir ses propres balises relatives aux objets et le tout sera géré de façon commune par tous les browsers compatibles avec le XML. Et voilà, le tour est joué.

Le Dhtml ou le Html dynamique
www.ccim.be/ccim328/htmlplus/idxdhtml.htm
© copyright 1998
Vanlancker.Luc@ccim.be