

## Menus Xoops compatibles CSS

Catégorie : Themes et templates

Publié par [Romu](#) le 29/11/2005

Une première approche fut l'intégration du menu DHTML TigrMenu. S'il la présentation est conforme à nos attentes, cette solution comporte un nombre trop élevé d'améliorations possible. Un menu DHTML nécessite l'exécution de code Javascript. Les visiteurs ayant désactivés cette fonctionnalité de leurs navigateurs ne peuvent exploiter les sous menus. Outre le contenu du menu, il est bafoué avec un volume de code important. L'utilisation de DHTML et donc le dessin dans le navigateur nécessite un positionnement fixe et donc difficile dans l'entête. Un défaut d'affichage survient lorsque les options des sous menus recouvrent une image ou un fichier média. Une seconde approche n'utilisait plus de DHTML et se rapprochait de notre objectif. Elle est décrite dans cet [article](#). Ce menu utilise encore un peu de Javascript pour repositionner les éléments des listes. Mais le poids de la solution a été sensiblement réduit. Le positionnement est plus précis mais le défaut d'affichage perdure. Enfin, le troisième opus est un aboutissement : -> pas de javascript -> compatibilité avec les des navigateurs récents (et pas uniquement) -> uniquement du CSS -> Positionnement facile et dimensions adaptatives pour une accessibilité optimale

Etape 1 : préparer le thème

Nous visions un thème XHTML/CSS complet mais avons dû renoncer. Par conséquent, pour l'organisation des colonnes nous continuons à utiliser une table comme le font la majorité des thèmes. Mais, pour l'organisation de l'entête, du conteneur et du pied nous avons utilisé la structure XHTML [décrite dans cet article](#). Le menu comporte sa propre feuille de style (pour plus de clarté) et est intégré au thème par une balise Div « navcontainer ». De la même manière le contenu de cette balise est disposé dans un fichier HTML dédié importé dans le fichier principal. Le fichier CSS

Dans notre feuille de style nous utilisons 100% de la largeur de la fenêtre via les balises #header et #conteneur. #header contiendra le menu. Les dimensions du menu sont fonction de la hauteur de sa police fixée en 'em'. Cela lui permet de s'adapter aux utilisateurs utilisant l'accessibilité (agrandissement des polices par le navigateur). Par conséquent, #header ne comporte pas de hauteur fixe. /\* fixe les polices du thème \*/ body { background-color: #F4F4F4; color: black; font: 0.9em Verdana, "Trebuchet MS", "MS Sans Serif"; } /\* Bloc entête \*/ #header { width:100%; background-color: #F4F4F4; } /\* Logo dans l'entête \*/ #logo { background-image: url("images/logo.gif"); background-repeat: no-repeat; padding-top: 10px; margin-left: 10px; width: 150px; height: 80px; float: left; } /\* Bannière dans l'entête \*/ #banniere { float: right; width: 468px; height: 60px; padding-right: 10px; padding-top: 10px; } /\* Bloc Conteneur \*/ #conteneur { width:100%; margin:0 auto; background-color:white; } /\* Bloc Centre \*/ #centre { } /\* Bloc Pied \*/ #pied { clear:both; height: 30px; color: #CC6666; text-align:center; font-size: 0.9em; } /\* Bloc Contenu \*/ #content { padding: 0; margin-top: 0; } /\* suivent les définitions classiques issues des thèmes 'en tables' \*/

Le fichier HTML : On a séparé la feuille de style du menu de celle du thème. Il faut donc y faire appel dans le Head. Le menu est intégré à la div Header.

```
title="http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
rel="external">http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> http://www.w3.org/1999/xhtml"
xml:lang="" lang=""> - /favicon.gif rel="SHORTCUT ICON" /> /xoops.css" /> " /> menu.css" />
//-->
    Etape 2 : les onglets
```

Notre menu est reparti en deux parties : les onglets et le sous menu. Pour faciliter le positionnement respectif de ces éléments en deux lignes, nous utilisons deux divs successives et indépendantes. Cette partie est réservée à la mise en

place des onglets. Pour cela nous avons tout simplement suivi le travail d'Eric Meyer paru sur Litasmatic (<http://css.maxdesign.com.au/listamatic/horizontal05.htm>). Il existe beaucoup de méthodes pour construire une liste horizontale. L'ingrédient principal est "display: inline", appliqué à une balise "LI".

- Créer une simple liste. Dans une liste non ordonnée, tous les éléments sont actifs, ce qui est essentiel à son fonctionnement. Elle contient des liens pour lesquels des règles CSS définies seront fixées. -> [Accueil](#) -> [modules/icontent/ Documents](#)
- suppression des éléments graphique des listes. On utilise la définition LI de la balise identifiée à « navlist » #navlist li { list-style-type: none; } -> [Accueil](#) -> [modules/icontent/ Documents](#)
- alignement des marges (internes et externes). La liste sera collée sur la gauche et le contenu sera décalé de 3px vers le bas. #navlist { padding: 3px 0px; margin-left: 0; }
- Liste horizontale. Le display: inline va permettre un positionnement horizontal des options de la liste, on modifie le LI. #navlist li { list-style-type: none; display: inline; }
- suppression des décorations du texte. Il s'agit du contenu des menus et donc des balises à l'intérieur des LI. #navlist li a { text-decoration: none; }
- zones utiles des options (marges internes). Pour créer un cadre autour de chaque lien des options du menu il faut lui créer un encadrement au moyen des marges internes (padding). Si la marge haute est fixe, on utilise une valeur proportionnelle pour la marge de droite afin d'adapter le cadre en fonction de la taille de la police. Donner une valeur uniquement pour le haut et la droite équivaut à fixer les 4 valeurs (raccourcis CSS). On crée aussi une marge interne fixe pour séparer les onglets. #navlist li a { text-decoration: none; padding: 3px 0.5em; }
- Ajout de la couleur et des bordures. Le trait en dessous des onglets pour #navlist et le contenu des options (les A) #navlist { padding: 3px 0px; margin-left: 0; border-bottom: 1px solid #6667CC; } #navlist li { margin: 0; list-style: none; display: inline; } #navlist li a { padding: 3px 0.5em; margin-left: 3px; border: 1px solid #6667CC; background: white; color: black; text-decoration: none; }
- Le rollover de la souris. La sous-classe hover des balises A #navlist li a: hover { background: #F4F4F4; }
- centrer le texte. Cela concerne les balise UL grâce à quelques règles de notre feuille de style. On utilise alors l'héritage de la définition existante dans style.css de notre thème. Style.css : ul { list-style: decimal inside; text-align: left; margin: 0px 0px; } .Cet héritage est possible car la feuille style.css est invoquée avant menu.css dans la balise HEAD de theme.html : .. " /> menu.css" /> ...
- un petit plus les arrondis sous Mozilla. Un petit truc ajouté pour un effet onglet plus propre amis qui ne passe que sous mozilla : arrondir les angles des balise A. Là aussi on travaille en dimension proportionnelle pour conserver l'accessibilité. #navlist li a { padding: 3px 0.5em; margin-left: 3px; border: 1px solid #6667CC; background: white; color: black; text-decoration: none; -moz-border-radius-topleft: 0.3em; -moz-border-radius-topright: 0.3em; }
- résultat une barre d'onglets propres et accessibles : navlist { clear: both; padding: 3px 0px; margin-left: 0; border-bottom: 1px solid #6667CC; font-size: 0.9em; } #navlist li { margin: 0; list-style: none; display: inline; } #navlist li a { padding: 3px 0.5em; margin-left: 3px; border: 1px solid #6667CC; background: white; color: black; text-decoration: none; -moz-border-radius-topleft: 0.3em; -moz-border-radius-topright: 0.3em; } #navlist li a: hover { background: #F4F4F4; } -> [Accueil](#) -> [modules/icontent/ Documents](#) ...



Etape 3 : le sous menu s'agit d'une liste beaucoup moins élaborée que la précédente. Généralement le sous menu est défini au sein des options du menu principal. Ses options (LI) contiennent des balises UL. Il en résulte une dépendance de position entre l'option et le sous menu. Ce point pose une difficulté quand on veut un sous menu à position fixe. En outre, pour pallier aux problèmes de chevauchement avec des images des menus précédents, j'ai pensé à placer le sous menu horizontalement dans la seconde ligne. Il devient alors très simple de le réserver dans cet espace fixe. J'ai donc innové en créant, pour le sous menu, une balise UL indépendante de la première.

- Créer une simple liste. Dans une liste non ordonnée, tous les éléments sont actifs, ce qui est essentiel à son fonctionnement. Elle contient des liens pour

lesquels des règles CSS définies seront fixées. -> [Articles](/modules/news/) -> [Téléchargements](/modules/mydownloads/) 2/ suppression des éléments graphiques des listes On utilise la définition LI de la balise identifiée « navlist » #nav2 li { list-style-type: none; } -> [Accueil](#) -> [Documents](/modules/icontent/) 3/ réglage des marges (internes et externes) La liste utilisera tout l'espace #nav2 { margin: 0; padding: 0 0 0; } 4/ Liste horizontale Le display: inline va permettre un positionnement horizontal des options de la liste, on modifie le LI. #nav2 li { list-style-type: none; display: inline; } 5/ suppression des décorations du texte Il s'agit du contenu des menus et donc des balises A à l'intérieur des LI. #nav2 li a { text-decoration: none; } 6/ zones utiles des options (liens) Le sous menu est une succession de liens qui ne surchargent pas la liste. On va donc laisser transparent leur zone utile #nav2 li a { text-decoration: none; background-color: transparent; } 7/ Ajout de la couleur et des bordures Le menu principal ayant un trait bas, on termine l'encadrement du sous menu. #nav2 { margin: 0; font-size: 0.9em; padding: 0 0 0; background-color: #69C; border-left: 1px solid #6667CC; border-right: 1px solid #6667CC; border-bottom: 1px solid #6667CC; } 8/ Le rollover de la souris Pour les options on assure simplement un effet sur les liens par un changement de couleur via les sous-classes des balises A. #nav2 a:link, #nav2 a:visited #nav2 a:hover, { color: white; } 9/ Centrer le texte De la même manière que pour le menu principal, on utilise l'héritage de la définition existante dans le fichier style.css de notre thème pour le centrage du texte des options. 10/ Résultats #nav2 { margin: 0; padding: 0 0 0; background-color: #69C; border-left: 1px solid #6667CC; border-right: 1px solid #6667CC; border-bottom: 1px solid #6667CC; font-size: 0.9em; } #nav2 li { display: inline; list-style-type: none; padding-left: 1em; } #nav2 a:link, #nav2 a:visited { color: white; background-color: transparent; text-decoration: none; } #nav2 a:hover { color: black; } -> [Accueil](#) -> [Documents](/modules/icontent/)

Vers la Forge... Articles Téléchargements FAQ Gabarits Glossaire Liens Plan du site

Etape 4 : exploitation du contexte Xoops Nous sommes parti d'un template Xoops. Nous y avons invoqué les définitions CSS spécifiques du menu et y avons importé le code HTML de ses listes dans une DIV ajoutée à l'entête :



Surlignons les blocs :



Nous remarquons le padding des logo et bannières. Le *clear: both;* et les marges de #navlist permettent un positionnement sous ses éléments. L'encadrement du menu par ses bordures définit bien la zone. Le menu s'adapte si on augmente la taille de la police du navigateur. Tout est en place. Nous allons rendre ce menu plus vivant en tenant compte des choix de l'utilisateur. Il s'agit de mettre évidence l'onglet en cours, adapter le contenu du sous menu et y surligner l'option en cours. Des styles pour les options courantes. Pour chacune des listes, nous définissons un style qui identifie l'option en cours. Les onglets Pour surligner un onglet comme l'onglet en cours, nous changeons simplement sa couleur de fond. Un petit plus est assuré en donnant la couleur de fond du sous menu au cadre bas de l'onglet pour parfaire l'effet visuel. La mise en application est possible en utilisant une classe définie #navlist li a#current { background: #69C; border-bottom: 1px solid #69C; } Les sous menus Dans le même esprit, nous allons cette fois simplement changer la couleur du lien dans la classe définie : #nav2 a#current { color: black; } Code HTML Il s'agit de terminer quel est le contexte Xoops courant pour appliquer la classe adéquate. Pour ce faire, on va exploiter les variables PHP et Smarty dans un script à ajouter au thème avant l'invocation de menu.html. Ce script, en tout début de theme.html va interpréter et terminer le module courant en en

r  cup  rant le r  pertoire et l'envoyer dans une smarty 'xoops\_cmod' : \$mod =  
 \$GLOBALS['xoopsModule']; \$dirname = (isset(\$mod) ? \$mod->getVar('dirname') : 'system');  
 \$GLOBALS['xoopsTpl']->assign( 'xoops\_cmod', \$dirname ); Ainsi, dans menu html, on peut travailler  
 les styles. On va   valuer le contenu de la smarty pour affecter la classe 'current'    l'option  
 concern  e dans la liste. [/" id="current">Accueil](#) -> [/">Accueil](#)  
[/modules/icontent/index.php?page=131" id="current">Documents](#) ->  
[/modules/icontent/index.php?page=131">Documents](#) ...De la m  me mani  re pour le sous menu  
 mais cette fois ci, avec le test dans la construction du lien -> [/modules/news/" id="current">Articles](#)  
 -> [/modules/news/submit.php" class="new">&nbsp;](#) -> [/modules/mydownloads/" id="current"](#)  
[>T     chargements](#) Un plus : le 'bouton' pour les soumission.s Il s'agit g  n  ralement de  
 submit.php dans les modules. On va utiliser une image de 20 pixels de large en fond d'un lien et  
 utiliser alors le padding. Le css #nav2 a.new { background-image: url(images/new.gif);  
 background-repeat: no-repeat; padding-left: 10px; padding-right: 10px; } Le html ->  
[/modules/news/" id="current">Articles](#) -> [/modules/news/submit.php" class="new">&nbsp;](#) ->  
[/modules/mydownloads/" id="current">T     chargements](#) Etape 5 (et finale) : Le menu utilisateur  
 Grand classique de nos r  alisations, le menu utilisateur est inclus au menu g  n  ral sous la  
 forme d'un onglet indiquant le pseudo et d'un sous menu des options adapt  es    son profil. Il ne  
 fonctionne qu'avec le hack du nombre de message dans header.php    la ligne 90 : if (\$xoopsUser  
 != "") { // d  but modification \$pm\_handler =& xoops\_gethandler('privmessage'); \$criteria = new  
 CriteriaCompo(new Criteria('read\_msg', 0)); \$criteria->add(new Criteria('to\_userid',  
 \$xoopsUser->getVar('uid'))); \$xoopsTpl->assign('nbrep', \$pm\_handler->getCount(\$criteria)); //fin  
 modification \$xoopsTpl->assign(array('xoops\_isuser' Le menu utilisateur est r  sum   dans par  
 l'utilisation de quelques scripts du module 'system'. Pour les identifier nous devons alors utiliser, en  
 plus de \$dirname construite dans le script en d  but de theme.html, la smarty xoops  
 \$SCRIPT\_NAME. Elle contient le nom du script en cours en partant de la racine du site. S'il est  
 d  pos      la racine du domaine, la variable ne contiendra que le seul nom du script php.  
 Sinon, elle contient aussi le nom du r  pertoire dans lequel est d  pos   xoops.Exemples  
 accueil (index.php) d'un xoops d  pos   en racine : => index.php accueil (index.php) d'un xoops  
 d  pos   dans le r  pertoire racine/rep1/rep2/ : => rep1/rep2/index.php Il faut imp  rativement  
 tenir compte de cette notion lors de l'envoi du th  me.html sur le serveur. En effet, le chemin de la  
 copie de test peut diff  rer de celle de la version r  elle de votre site. Les scripts du menu  
 utilisateur sont : readpmsg.php, viewpmsg.php, user.php, edituser.php, userinfo.php et register.php.  
 Si nous sommes sur l'un de ces scripts, nous allons donc, dans theme.html affecter la valeur 'user'  
    la smarty : \$mod = \$GLOBALS['xoopsModule']; \$dirname = (isset(\$mod) ?  
 \$mod->getVar('dirname') : 'system'); \$GLOBALS['xoopsTpl']->assign( 'xoops\_cmod', \$dirname );  
 \$GLOBALS['xoopsTpl']->assign( 'xoops\_cmod', 'user' ); Pour finir, le code html d  di   au  
 menu utilisateur. Il affiche le nombre de messages priv  s en attente de lecture et transforme donc  
 le lien Onglets ... 0}> [/viewpmsg.php" id="current">](#) -> [/viewpmsg.php" id="current">](#)  
 -> [/register.php" id="current">S'enregistrer](#) 0}> -> [/viewpmsg.php" >](#) ->  
[/viewpmsg.php" >](#) -> [/register.php" >S'enregistrer](#) Sous menus ... ->  
[/admin.php">Administration](#) -> [/user.php">Votre compte](#) 0}> -> [/viewpmsg.php">Messagerie:](#)  
 -> [/viewpmsg.php">Messagerie](#) -> [/user.php?op=logout">Deconnexion](#)    Conclusion  
 Nous remercions les xoopsiens Christian, TheCat, DenisDlu pour leurs participations diverses    la  
 r  alisation de ce menu. Pour information, nous vous indiquons qu'il est possible de d  cliner le  
 format des onglet    l'aide des tr  s bons guides disponibles (en anglais)    l'adresse suivante  
[Listamatic de maxdesign](#)  
 Pour vos remerciements, nous vous laissons le soins de r     rencer notre site sur les v  tres :

