

# Un menu horizontal

Par neoxx78



**OPENCLASSROOMS**

[www.openclassrooms.com](http://www.openclassrooms.com)

*Licence Creative Commons 6 2.0  
Dernière mise à jour le 20/09/2011*

## Sommaire

Sommaire .....	2
Un menu horizontal .....	3
La structure XHTML .....	3
Avec display : inline .....	4
Avec float : left ou right .....	4
Un exemple de menu .....	6
Un menu style Bluzaz .....	6
Partager .....	8



## Un menu horizontal


Par  neox78

Mise à jour : 20/09/2011

Difficulté : Facile 



Bienvenue dans ce petit tutoriel portant sur la création d'un menu horizontal de base.

Vous avez sûrement suivi les cours sur le [XHTML/CSS](#) de M@teo21, et surtout son [TP : Créons le design de votre site Web](#). Non ?  Allez les lire, alors. Si c'est déjà le cas, prenez le tutoriel qui suit comme un petit supplément à ce TP : nous n'irons donc pas jusqu'au top niveau de la programmation *web*, je ne vous donnerai que l'essentiel. À vous après de laisser libre cours à votre imagination.

Nous allons donc par la suite étudier différents procédés pour coder votre menu horizontal. Un menu horizontal permet une navigation plus simple et intuitive sur votre site. Je trouve aussi que ça donne un aspect plus « pro » au site mais ça, c'est un avis personnel.

En d'autres termes, un menu horizontal est souvent très apprécié des visiteurs. Mais qu'attendons-nous pour démarrer !


Sommaire du tutoriel :



- [La structure XHTML](#)
- [Avec display : inline](#)
- [Avec float : left ou right](#)
- [Un exemple de menu](#)

### La structure XHTML

Nous commencerons par coder notre structure XHTML : c'est le plus facile.

En XHTML, un menu se déclare avec une liste non ordonnée `<ul>` et `<li>`. C'est plus propre dans le code et cela **respecte mieux la sémantique** qu'une flopée de `<p>` alignés.  Si l'envie vous prend, vous pouvez aussi le faire avec une liste ordonnée `<ol>` : chacun ses goûts.

Donc : une liste `<ul>` avec une puce `<li>` pour chaque bouton du menu... Qu'est-ce qu'il manque ? La balise `<a>` pour le lien du bouton, bien sûr. Tout y est !

Ce qui nous donne au final :

#### Code : HTML

```
<ul id="menu_horizontal">
<li><a href="accueil.php">Accueil</a></li>
<li><a href="news.php">News</a></li>
<li><a href="livreor.php">Livre d'or</a></li>
<li><a href="contact.php">Contact</a></li>
<li><a href="admin.php">Admin</a></li>
</ul>
```

Voilà un menu basique, c'est-à-dire avec cinq boutons — ce qui n'est ni trop ni trop peu. Comme nous le voyons, un menu horizontal vous sert à vous diriger vers vos pages principales, **inutile d'y mettre toutes vos pages** : il faut que ça reste clair pour le visiteur, la navigation sur votre site n'en sera que simplifiée.



La navigation est primordiale sur un site, autant que son aspect ! Gardez toujours en tête que votre site doit pouvoir être visité facilement et intuitivement par tous les utilisateurs.

Mais ça y est, c'est déjà terminé ! Heureusement, il reste encore la partie en CSS qui mettra en forme notre menu (car pour l'instant il est encore à la verticale).

## Avec display : inline

Le codage XHTML fini, la question que nous devons nous poser maintenant est :



Comment mettre notre menu à l'horizontale grâce au CSS ?

En CSS, il existe deux propriétés qui nous intéressent pour mettre un menu à l'horizontale ; la première que nous allons étudier est `display : inline`. Cela vous fait penser à `display : block`, non ? En réalité, `display : inline` fait exactement le contraire de `display : block`. Vous commencez à voir l'utilité de cette propriété ?



Une balise `block`, par définition, va toujours à la ligne ; or, si on lui demande de se comporter comme une balise en-ligne, elle restera sur la même ligne.

Il nous suffit donc d'appliquer cette propriété aux puces `<li>` pour qu'elles se placent sur une même ligne. Et voilà le travail :

Code : CSS

```
/* Je sélectionne les <li> du menu horizontal */
ul#menu_horizontal li {
  display : inline;
  padding : 0 0.5em; /* Pour espacer les boutons entre eux */
}
ul#menu_horizontal {
  list-style-type : none; /* Car sinon les puces se placent n'importe où */
}
```

Voici ce que ça nous donne :

[Accueil](#) [News](#) [Livre d'or](#) [Admin](#) [Contact](#)

À vous d'adapter ensuite ce menu à votre design : en quelques coups de CSS, ce n'est plus qu'un jeu d'enfants.



Cependant, cette propriété limite quand même nos possibilités pour le design.

Par exemple, il sera **impossible d'aligner une partie des boutons à gauche et l'autre à droite** — comme c'était le cas sur le Site du Zéro à une certaine époque. Cette disposition peut sûrement vous intéresser, elle n'est pourtant pas réalisable avec le code montré ci-dessus. Heureusement, il existe une autre technique un petit peu plus longue. Ouvrez grand vos oreilles, heu... les yeux en l'occurrence.

## Avec float : left ou right

La technique du `float` est moins claire que la précédente, mais elle permet un affichage des boutons à gauche et à droite grâce à `float : left` et `right`.

Cette technique offre donc une possibilité de plus que la précédente : c'est peut-être pour cela que vous la préférerez à l'autre (c'est mon cas). Mais, tout d'abord, il faudra changer un petit peu notre code XHTML. Ce changement consiste à rajouter une `class "bouton_gauche"` pour les liens qui seront à gauche, et une autre `"bouton_droite"` pour les boutons qui

seront à droite. (Ouf, je reste logique.) 😊

#### Code : HTML

```
<ul id="menu_horizontal">
<li class="bouton_gauche"><a href="accueil.php">Accueil</a></li>
<li class="bouton_gauche"><a href="news.php">News</a></li>
<li class="bouton_gauche"><a href="livreor.php">Livre d'or</a></li>
<li class="bouton_droite"><a href="admin.php">Admin</a></li>
<li class="bouton_droite"><a href="contact.php">Contact</a></li>
</ul>
```



Hé ! attends ! Pourquoi as-tu inversé les puces « Admin » et « Contact » ?

Elles étaient dans l'ordre inverse avec la technique `display : inline ;` or, je veux que mes boutons soient dans le même ordre qu'avant.

Je t'explique. Dans notre CSS, nous allons appliquer la propriété `float` à nos puces `<li>`. La class `"bouton_gauche"` se verra attribuer un `float : left` et l'autre class `"bouton_droite"`, un `float : right`. Par définition, un flottant est un élément qu'on veut faire « entourer », « flotter », « immerger » dans l'élément qui le suit. Retenez l'exemple de la lettrine qui flotte au début d'un paragraphe. 😊

Dans notre cas, cela nous donne :

- la puce « Accueil » flotte à gauche de « News » ;
- « News » flotte lui-même à gauche de « Livre d'or » ;
- « Livre d'or » flotte aussi à gauche, il prend donc place à côté des autres.

Ce qui n'est valable que dans le cas d'un flottant à gauche, si vous regardez bien. Si on applique la même chose pour les flottants à droite :

- la puce « Admin » flotte à droite de « Contact » ;
- « Contact », lui-même flottant, prend place à la suite, c'est-à-dire à gauche d'« Admin ».

Si nous avons gardé le code XHTML d'avant, la puce « Contact » se serait donc retrouvée tout à droite ! Ce n'est pas catastrophique, mais il faut au moins connaître cette petite subtilité.

Rappelez-moi ce que nous allons mettre ensuite dans notre CSS ? Oui, très bien : la class `"bouton_gauche"` avec `float : left`, et exactement le contraire pour l'autre class.

#### Code : CSS

```
ul#menu_horizontal { /*Ici , c'est ma configuration, à vous de la
changer */
width : 600px;
height : 25px;
list-style-type : none; /* Car sinon les puces se placent n'importe
où */
}
ul#menu_horizontal li {
padding : 0 0.5em; /* Pour espacer les boutons entre eux */
}
li.bouton_gauche {
float : left;
}
li.bouton_droite {
```

```
float : right;
}
```

Ce qui nous donne :

[Accueil](#) [News](#) [Livre d'or](#)

[Contact](#) [Admin](#)

Il ne vous manque plus qu'à l'adapter *via* le CSS au design de votre site. Votre imagination sera votre seule aide pour la suite. Vous pouvez aussi vous inspirer des sites que vous voyez tous les jours, et regarder leurs codes : vous allez sûrement apprendre beaucoup.

Pourquoi vous me regardez comme ça ? 🤔 J'ai compris ! Vous voulez un exemple, c'est ça ?

## OUAIS !

On ne dit pas ouais mais oui, c'est plus poli, bande de voyous. D'accord, nous allons voir un exemple uniquement pour vous donner des idées. Mais je le dirai toujours : c'est en s'inspirant des autres et en utilisant son imagination qu'on progresse le mieux.

## Un exemple de menu

Ici, nous allons voir un exemple qui va sûrement vous permettre d'éclairer votre imagination. Il sera très basique, rien de bien compliqué donc, mais ce sont souvent les choses simples qui font le plus d'effet. Je vous donne d'abord les codes que j'ai utilisés, et ensuite les explications point par point.

### Un menu style Bluzaz

Autrefois, à l'époque de la création de ce tutoriel, le Site du Zéro était habillé par le thème Bluzaz qui possédait un beau petit menu horizontal. C'est ce même menu que je vais vous montrer ici. J'espère que cela plaira aux nostalgiques.

Premièrement, nous allons utiliser la technique du `float` — c'est-à-dire que nous allons prendre la même structure XHTML que précédemment et définir des liens à gauche et à droite. De plus, nous ajouterons une image pour illustrer chaque bouton.

#### Code : HTML

```
<html>
<head><title>FLOAT</title><link rel="stylesheet" media="screen"
type="text/css" title="index" href="tuto_menu_float.css" /></head>

<body>

<ul id="menu_horizontal">
<li class="bouton_gauche"><a href="#">Accueil</a></li>
<li class="bouton_gauche"><a href="#">News</a></li>
<li class="bouton_gauche"><a href="#">Livre d'or</a></li>
<li class="bouton_droite"><a href="#">Admin</a></li>
<li class="bouton_droite"><a href="#">Contact</a></li>
</ul>

</body>
</html>
```

Ensuite, le CSS. Essayez d'abord de décrypter le code vous-même et, si l'utilisation de certaines propriétés vous paraît étrange, vous pourrez toujours vous autoriser à lire les explications écrites plus bas.

#### Code : CSS

```
ul#menu_horizontal {
width : 600px;
height : 30px;
margin : 2em 0 0 0;
padding : 0;
background-color : #f4f9fd;
border : 1px dashed black;
list-style-type : none;
}

ul#menu_horizontal li {
padding : 0 0.5em;
line-height : 30px;
}

ul#menu_horizontal li.bouton_gauche {
float : left;
border-right : 1px dashed black;
}

ul#menu_horizontal li.bouton_droite {
float : right;
border-left : 1px dashed black;
}

ul#menu_horizontal a {
color : black;
text-decoration : none;
padding : 0 0.5em;
font : 0.8em "Trebuchet MS";
}

ul#menu_horizontal a:hover { text-decoration : underline; }

ul#menu_horizontal a img {
border : none;
padding : 0 0.3em;
}
```

- Premièrement, pour le code de `ul#menu_horizontal`, il ne devrait pas y avoir de problème. Nous définissons la taille du bloc, sa hauteur (c'est toujours mieux), sa couleur de fond, une bordure grise en pointillé de 1 px de largeur tout autour et une marge en haut pour éviter que le menu ne colle au haut de la fenêtre. Si nous nous arrêtons là, nous obtiendrions un simple bloc avec des bordures en pointillé et un fond.  
Deuxièmement, nous définissons le `ul#menu_horizontal`, c'est-à-dire que nous sélectionnons toutes les listes qui ont l'id `#menu_horizontal`. Nous utilisons les propriétés `list-style-type` avec comme valeur `none` pour enlever les puces noires qui se mettent automatiquement. Puis `margin` et `padding`, avec comme valeur 0 **pour éviter l'indentation de la liste** (automatique en XHTML) et les marges qui ne seraient pas souhaitées. (Histoire d'éviter que le menu soit carrément décalé.) 🤔
- Ensuite, nous définissons chaque bouton de la liste grâce à `ul#menu_horizontal li`. Nous leur ajoutons des marges intérieures pour qu'ils ne soient pas trop collés au bord, et enfin nous utilisons `line-height`, avec comme valeur la hauteur totale du menu. Cette technique nous permet de centrer verticalement le texte dans sa case. C'est une astuce à retenir, bien qu'elle ne marche pas avec tout malheureusement.

- Pour les codes suivants avec `li.droite` et `li.gauche`, vous savez exactement ce qu'il faut mettre : un `float` avec `left` ou `right`, suivant que le bouton flotte à gauche ou à droite.



Mais pourquoi mets-tu des bordures uniquement à gauche ou à droite ?

En réalité, c'est la même subtilité que nous avons vue plus haut avec l'histoire de l'ordre des boutons ; je ne vais donc pas m'y attarder. Si je fais ça, c'est pour éviter les doubles bordures, car, par exemple, un flottant à gauche n'aura besoin que d'une bordure à droite, puisque à gauche il y aura déjà la bordure du menu. **Chaque bouton flotte donc à côté de la bordure du bouton précédent.**

- Après, nous définissons le style des liens : leur police, leur taille, leur décoration. Nous rajoutons aussi une marge intérieure pour augmenter la « surface » de contact du lien et aérer le menu.
- Enfin, les deux dernières parties de CSS sont banales : nous rajoutons un petit effet au survol du lien et nous enlevons les bordures des images-liens.

Tout ce boulot nous donne le joli menu que vous voyez ci-dessous.



C'était facile, hein ? 😊 (Voire même tout bête.) Les menus horizontaux n'ont maintenant plus de secret pour vous. Cependant, vous pouvez toujours continuer à enrichir vos connaissances sur les menus par l'étude des menus dits graphiques (menus découpés sur une image de fond) et des menus déroulants. C'est encore une autre histoire qui n'a malheureusement pas sa place ici. Ces derniers sont certes plus ardu à concevoir, mais je suis sûr qu'avec du temps vous en viendrez à bout.

Sur ce, bon codage !

### Partager



Ce tutoriel a été corrigé par les [zCorrecteurs](#).