Les sélecteurs complexes en CSS

(ex : menu horizontal)

Rédigé par: Jean-Luc Colson
Date première rédaction: Janvier 2020

SUIVI DES MODIFICATIONS A LA FICHE			
Série	Date	Page(s) modifiée(s)	Raison

Contents

Exercice : création d'un menu horizontal sticky en HTML et CSS 4
Squelette HTML du menu 4
Mise en forme du menu en CSS 5

Exercice : création d'un menu horizontal sticky en HTML et CSS

Au cours des dernières leçons, nous avons vu beaucoup de nouveaux concepts HTML et CSS. Il est maintenant temps de mettre ce qu'on a vu en application en créant un menu simple horizontal entièrement en HTML et en CSS.

Étant donné que c'est notre premier « vrai » exercice, nous allons progresser ensemble.

Commençons avec un conseil d'ordre général : lorsqu'on se lance dans un projet de code, il est rarement bénéfique de commencer immédiatement à coder. Au contraire, on commencera généralement par clarifier ce qu'on souhaite obtenir et par définir les différents langages, objets, etc. que nous allons devoir utiliser pour arriver au résultat voulu.

Procéder comme cela limite les risques d'avoir à revenir en arrière et à réécrire son code au milieu du projet car on n'avait pas pensé à ceci ou cela et fait au final gagner beaucoup de temps !

Ici, le projet est très simple puisqu'on souhaite simplement créer un menu horizontal simple. Pour créer des menus, nous allons utiliser des éléments de liste ul que nous allons ensuite mettre en forme en CSS.

Squelette HTML du menu

Nous allons commencer par créer le squelette de notre menu en HTML avant de le mettre en forme en CSS.

Ici, nous allons utiliser une liste ul qui va représenter notre menu. Chaque élément de liste li va représenter un onglet de notre menu.

Comme un menu est utilisé pour naviguer entre les pages d'un site, il va falloir que nos onglets de menu soient cliquables. Nous allons donc ajouter des éléments a dans nos éléments de liste. Pour cet exercice, je laisserai mes liens vides en écrivant href="#".

En pratique, nous devrons également placer notre menu principal de navigation dans un élément structurant nav qui sert à indiquer aux navigateurs et moteurs de recherche que ce qui est dans l'élément nav est notre menu principal de navigation.

Voici donc le code HTML de notre menu et le résultat dans le navigateur :

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Cours HTML et CSS</title>
        <meta charset= "utf-8">
        <link rel="stylesheet" href="cours.css">
    </head>

    <body>
        <nav>
            <ul>
                <li><a href="#">Cours Complets</a></li>
                <li><a href="#">Articles</a></li>
                <li><a href="#">Contact</a></li>
                <li><a href="#">A propos</a></li>
            </ul>
        </nav>
    </body>
</html>
```

Pour le moment, il s'agit visuellement d'une simple liste non ordonnée et cela ne ressemble pas à un menu. C'est tout à fait normal : je vous rappelle que la mise en page est du ressort du CSS.

Mise en forme du menu en CSS

Nous allons maintenant transformer notre liste HTML en quelque chose qui ressemble visuellement à un menu en CSS.

Ici, nous allons déjà commencer par effectuer un reset des marges intérieures et extérieures des différents éléments de notre page pour nous assurer d'avoir un affichage cohérent d'un navigateur à l'autre. Profitons-en également pour définir une liste de polices d'écriture pour notre page.

```css
/*Reset CSS*/
*{
    margin: 0px;
    padding: 0px;
    font-family: Avenir, sans-serif;
}
```

Ensuite, nous allons enlever les puces devant nos éléments de liste. Nous allons pouvoir faire cela en utilisant la propriété list-style-type avec sa valeur none.

Ici, nous allons cibler uniquement la liste de notre menu avec le sélecteur nav ul afin de ne pas enlever les puces des autres listes qui pourraient potentiellement être affichées dans notre page.

```css
nav ul{
    list-style-type: none;
}
```

A partir d'ici, nous avons plusieurs solutions en CSS pour mettre nos éléments de liste sur la même ligne. Les deux solutions fonctionnant le mieux ici vont être d'ajouter soit un display : inline-block, soit un float : left à nos éléments de liste.

Ici, je vais plutôt opter pour un float : left pour n'avoir aucune espace entre mes éléments de liste et faciliter la mise en forme.

En effet, je vous rappelle qu'une espace va se créer entre différents éléments possédant un display : inline-block si ces éléments ne sont pas collés dans votre code.

Cette espace, généralement de 4px, est l'équivalent d'une espace insécable. On peut le supprimer en collant les différents éléments dans le code.

Comme on applique un float : left à tous nos éléments de liste, nous allons également utiliser le clearfix avec le pseudo-élément ::after que nous allons appliquer à la liste en soi pour éviter que sa hauteur ne soit nulle.

```css
nav li{
    float: left;
}

nav ul::after{
    content: "";
    display: table;
    clear: both;
}
```

C'était l'étape la plus complexe du menu. Nous allons ensuite espacer nos différents onglets de menu. Pour cela, on va attribuer une largeur égale à 100% divisée par le nombre d'éléments de menu à chaque élément de menu.

Nous allons également pouvoir centrer le contenu textuel de chacun de nos éléments de menu dans l'élément de menu en appliquant un text-align : center aux différents éléments.

```
nav li{
    float: left;
    width: 25%;
    text-align: center;
}
```

Note : Ici, notre élément nav occupe 100% de la largeur disponible par défaut et on dirait donc que le menu est centré dans la page. Cependant, ce n'est pas le cas (il suffit de lui attribuer une largeur plus petite pour s'en apercevoir). Pour centrer le menu, on pourra appliquer une largeur explicite à l'élément nav ainsi que margin : 0 auto.

```
nav{
    width: 100%;
    margin: 0 auto;
}
```

Une fois arrivé ici, on s'aperçoit d'un problème d'ergonomie : seul le texte est cliquable et non pas tout l'espace dans chaque élément de menu. Cela est dû au fait que l'élément a est un élément inline. Changeons donc ce comportement par défaut et profitons-en pour enlever le trait de soulignement et pour changer la couleur de nos textes.

```
nav a{
    display: block;
    text-decoration: none;
    color: black;
}
```

Ensuite, nous allons vouloir mettre en relief un élément de menu lorsque celui-ci est survolé par l'utilisateur. Nous allons pouvoir faire cela en utilisant la pseudo-classe :hover.

On va par exemple pouvoir changer la couleur du texte et ajouter une bordure basse lorsqu'un utilisateur passe sa souris sur un élément de menu.

```
nav a:hover{
    color: orange;
    border-bottom: 2px solid gold;
}
```

Ici, on se retrouve face à un autre problème d'ergonomie : en effet, en ajoutant une bordure durant l'état :hover, la hauteur de l'élément est modifiée de la taille de la bordure et cela va faire bouger les différents contenus sous le menu.

La façon la plus simple de résoudre ce problème est d'ajouter une bordure de même taille mais de couleur transparente à nos éléments a dans leur état normal. Ainsi, les éléments auront toujours la même hauteur.

```css
nav a{
    display: block;
    text-decoration: none;
    color: black;
    border-bottom: 2px solid transparent;
}
```

Finalement, on va pouvoir aérer notre menu en hauteur. Ici, je vous propose d'ajouter des padding haut et bas de 10px à nos éléments a. Cela aura également pour effet de séparer la bordure basse de ces éléments de leur contenu pour un meilleur rendu visuel.

```css
nav a{
    display: block;
    text-decoration: none;
    color: black;
    border-bottom: 2px solid transparent;
    padding: 10px 0px;
}

nav a:hover{
    color: orange;
    border-bottom: 2px solid gold;
}
```

Pour rendre enfin notre menu sticky, il va suffire d'ajouter une position :sticky à notre élément nav avec une propriété top :0px si on souhaite que le menu reste collé en haut de la page. Nous allons également en profiter pour ajouter une couleur de fond au menu.

Notez que pour constater l'effet du positionnement sticky il va falloir pouvoir descendre dans la page. N'hésitez donc pas à ajouter un conteneur div sous le menu et à lui donner une hauteur de 2000px par exemple.

```css
nav{
    width: 100%;
    margin: 0 auto;
    background-color: white;
    position: sticky;
    top: 0px;
}
```

Voilà tout pour ce menu simple créé en HTML et en CSS. J'y ai ajouté deux barres horizontales pour bien marquer la séparation. N'hésitez pas à modifier ou à ajouter des styles à ce menu ! Vous pouvez trouver le code complet du menu ci-dessous :

D:\data\Cours\ISL\HTML CSS\Site\html\selecteurs-15 menu1.html - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

cours08bis.html | selecteurs-15 menu1.html

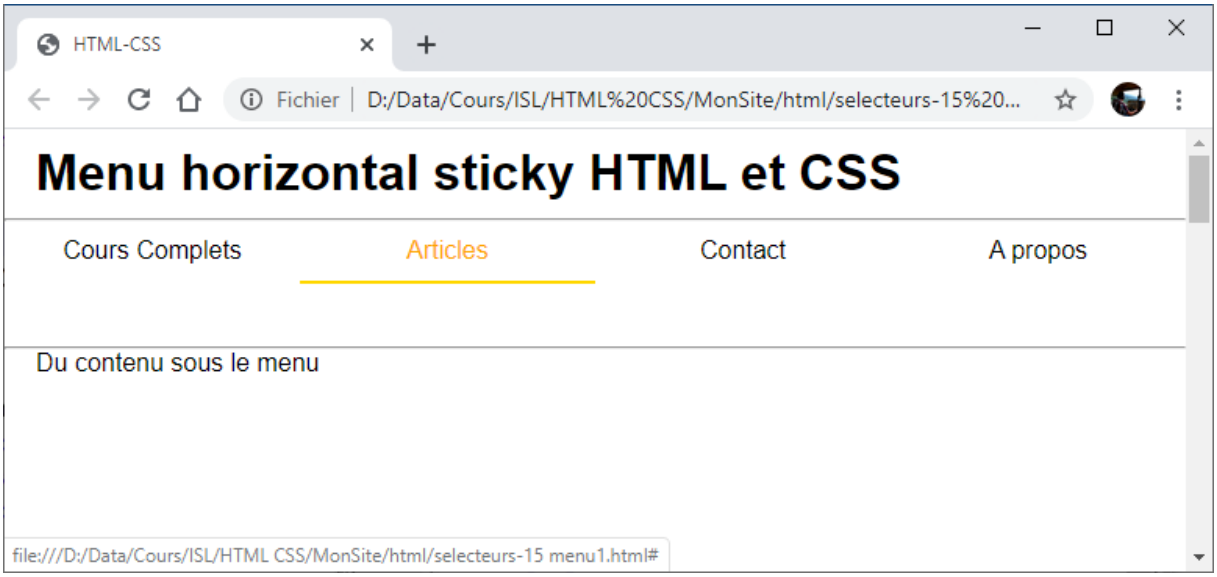
```html
<!DOCTYPE html>
<html>
<head>
    <title>HTML-CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet"
    href="..\css\selecteurs-15 menu1.css">
</head>
<body>

<h1>Menu horizontal sticky HTML et CSS </h1>
<hr>
<nav>
  <ul>
    <li><a href="#">Cours Complets</a></li>
    <li><a href="#">Articles</a></li>
    <li><a href="#">Contact</a></li>
    <li><a href="#">A propos</a></li>
  </ul>
</nav>
<hr>
<div class="conteneur">
  <p>Du contenu sous le menu</p>
</div>

</body>
</html>
```

cours08bis.css | selecteurs-15 menu1.css

```css
/*Reset CSS*/
*{
    margin: 0px;
    padding: 0px;
    font-family: Avenir, sans-serif;
}
nav{
    width: 100%;
    margin: 0px auto 40px auto;
    background-color: white;
    position: sticky;
    top: 0px;
}
nav ul{
    list-style-type: none;
}

nav li{
    float: left;
    width: 25%;
    /*100% divisé par le nombre
    d'éléments de menu*/
    text-align: center;
    /*Centre le texte dans les éléments de menu*/
}

/*Evite que le menu n'ait une hauteur nulle*/
nav ul::after{
    content: "";
    display: table;
    clear: both;
}

nav a{
    display: block;
    /*Toute la surface sera cliquable*/
    text-decoration: none;
    color: black;
    border-bottom: 2px solid transparent;
    /*Evite le décalage des éléments sous
    le menu à cause de la bordure en :hover*/
    padding: 10px 0px;
    /*Agrandit le menu et espace
    la bordure du texte*/
}

nav a:hover{
    color: orange;
    border-bottom: 2px solid gold;
}

.conteneur{
  margin: 0px 20px;
  height: 1500px;
}
h1{margin: 10px 20px;}
```

Hyper Text Markup Language file | length : 506 lines : 27 | Ln : 27 Col : 8 Sel : 0 | 0 | Windows (CR LF) | UTF-8 | INS

Et voici le résultat :


HTML-CSS
Fichier | D:/Data/Cours/ISL/HTML%20CSS/MonSite/html/selecteurs-15%20...
Menu horizontal sticky HTML et CSS
Cours Complets
Articles
Contact
A propos
Du contenu sous le menu
file:///D:/Data/Cours/ISL/HTML CSS/MonSite/html/selecteurs-15 menu1.html#