



Squelette HTML du menu déroulant

Pour représenter nos sous-menus, nous allons simplement imbriquer une nouvelle liste dans les différents éléments de notre liste principale.

Pour représenter nos sous-menus, nous allons simplement imbriquer une nouvelle liste dans les différents éléments de notre liste principale.

```
<nav>
  <ul>
    <li class="deroulant"><a href="#">Exemple1
      &nbsp;</a>
      <ul class="sous">
        <li><a href="#">ss-menu</a></li>
        <li><a href="#">ss-menu</a></li>
        <li><a href="#">ss-menu</a></li>
      </ul>
    </li>
    <li class="deroulant"><a href="#">Exemple2
      &nbsp;</a>
      <ul class="sous">
        <li><a href="#">ss-menu</a></li>
        <li><a href="#">ss-menu</a></li>
        <li><a href="#">ss-menu</a></li>
      </ul>
    </li>
    <li><a href="#">Contact</a></li>
    <li><a href="#">A propos</a></li>
  </ul>
</nav>
```

Ici, nous créons un deuxième niveau de menu sous nos deux premiers onglets de menu.

Notez que j'ai rajouté des attributs **class** et des entités HTML ** ** dans le code (qui servent à créer un espace double). Cela va nous être utile plus tard en CSS.

Mise en forme du menu en CSS

On va donc déjà vouloir par défaut cacher les sous menus et ne les afficher que lorsqu'un utilisateur passe sa souris par-dessus l'onglet de menu correspondant. Nous allons donc appliquer un **display : none** par défaut à nos sous-menus et un **display : block** lorsqu'un utilisateur passe sa souris sur l'onglet du menu principal correspondant.

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

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

nav ul li{
  float: left;
  width: 25%;
  text-align: center;
  position: relative;
}

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

```
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;
}
```

On va également en profiter pour ajouter une couleur de fond aux sous-menus ainsi qu'une ombre légère autour des sous-menus pour qu'ils se distinguent du reste de la page avec **box-shadow**.

```
.sous{
  display: none;
  box-shadow: 0px 1px 2px #CCC;
  background-color: white;
  position: absolute;
  width: 100%;
  z-index: 1000;
}

nav > ul li:hover .sous{
  display: block;
}
```

Ici, notez qu'on peut placer les styles généraux de mes sous-menus soit dans le sélecteur **nav > ul li:hover .sous**, soit dans le sélecteur **.sous**. Par défaut, on préférera placer les styles dans le sélecteur le plus simple et général.

Le sélecteur **nav > ul li:hover .sous** peut sembler complexe à première vue. Il sert à cibler un sous menu **.sous** lorsqu'un utilisateur passe sa souris sur un éléments **li** de la liste représentant notre menu principal.

Ensuite, on va vouloir que les éléments de nos sous-menus s'affichent cette fois-ci les uns en dessous des autres, occupent tout l'espace disponible dans leur conteneur et que le texte soit aligné à gauche. On va donc annuler le flottement hérité des onglets du menu principal.

```
.sous li{
  float: none;
  width: 100%;
  text-align: left;
}
```

On va également déjà en profiter pour mettre en forme les onglets de nos sous-menus et s'appuyant une nouvelle fois sur les styles déjà créés. Ici, on va se contenter de supprimer la bordure basse héritée du menu principal, d'ajouter une marge interne et une couleur de fond lorsqu'on passe la souris sur un élément.

```
.sous a{
  padding: 10px;
  border-bottom: none;
}
.sous a:hover{
  border-bottom: none;
  background-
  color:RGBa(200,200,200,0.1);
}
```

On va aussi montrer aux utilisateurs que notre menu est déroulant en ajoutant une petite flèche à côté des textes des onglets de menu qui contiennent des sous-menus. Pour cela, on va insérer le symbole ▼ après le texte en utilisant le pseudo-élément : **:after**, ce qui va également nous permettre de définir la taille du symbole.

Une fois arrivé ici, il nous reste un détail technique à régler qui est la partie complexe de ce menu : pour le moment, lorsqu'on affiche un sous-menu, le contenu situé sous le menu est poussé vers le bas ce qui n'est pas le comportement souhaité.

On va pouvoir régler cela en appliquant une **position : absolute** à nos sous-menus, ce qui va avoir pour effet de les retirer du flux normal de la page. Les sous-menus n'impacteront plus le contenu suivant le menu.

Pour que les sous-menus ne passent pas derrière le contenu qui suit le menu, nous allons également définir un **z-index** élevé. Nous allons également définir la taille des sous-menus à 100%.

Pour finir, je replace les sous-menus correctement sous leur onglet correspondant et pour qu'ils fassent bien la même taille, on va également appliquer une **position : relative** aux éléments du menu principal.

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