Faire un menu horizontale :

**Code (x)html**

<ul>

<li><a href="#">Item 1</a></li>

<li><a href="#">Item 2</a></li>

<li><a href="#">Item plus long</a></li>

</ul>

Ce code html va donc être stylisé afin d'obtenir un menu horizontal, soit sous forme de [boutons simples](http://css.mammouthland.net/menu-horizontal-en-css.php#simple) sans images, soit avec [des images de fond](http://css.mammouthland.net/menu-horizontal-en-css.php#graphique) qui changeront au survol de la souris par effet rollover.

**Mise en forme de boutons CSS simples**

Le principe est donc de transformer cette liste à puces verticale en liste (sans puces) horizontale. Pour commencer, on va donc déclarer la liste (ul ; list-style-type:none) sans puces et mettre mes marges extérieures (margin) et intérieures (padding) à zéro.

Chaque item (li) devra néanmoins être légèrement écarté de son voisin, on va donc déclarer une marge droite à 2 pixels (margin-left:2px).

Vient ensuite le comportement des liens de cette liste (ul li a). Le display:block va permettre de donner une largeur fixe et identique à chaque item, et pour ne pas qu'ils s'empilent les uns sur les autres, on déclare les différents élément flottants à gauche float:left ce qui va permettre aux items suivants d'aller se mettre... à droite (vous suivez ?).  
Mais comme MSIE ne fait rien comme tout le monde, cet attribut doit être aussi déclaré pour les items seuls.

Reste à déclarer l'aspect du bouton : sa taille (width) ; sa couleur de fond (background-color) ; la couleur du texte, son aspect, et son positionnement dans le bouton (color ; text-decoration ; text-align ; padding) ; l'aspect des bordures (border-width ; border-style ; border-color).

Et pour fignoler le tout, on définit l'aspect que prendra ce même bouton lors du survol de la souris (ul li a:hover).

**Code CSS**

ul {

padding:0;

margin:0;

list-style-type:none;

}

li {

margin-left:2px;

float:left; /\*pour IE\*/

}

ul li a {

display:block;

float:left;

width:100px;

background-color:#6495ED;

color:black;

text-decoration:none;

text-align:center;

padding:5px;

border:2px solid;

/\*pour avoir un effet "outset" avec IE :\*/

border-color:#DCDCDC #696969 #696969 #DCDCDC;

}

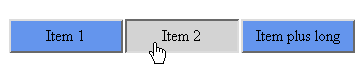
ul li a:hover {

background-color:#D3D3D3;

border-color:#696969 #DCDCDC #DCDCDC #696969;

}

**Résultat attendu**

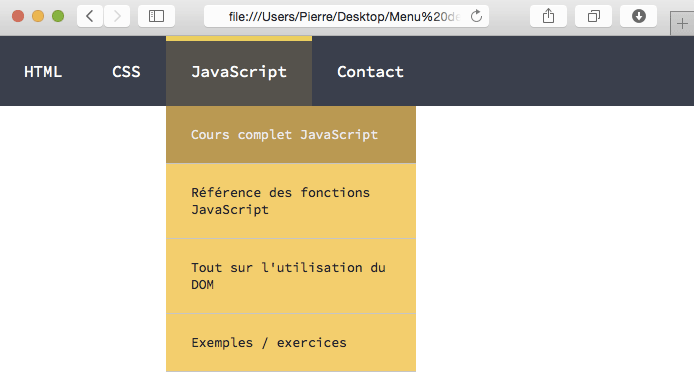


Avec menu défilant :

## **Présentation de l’exercice**

Le but de cet exercice est de **réaliser un menu déroulant entièrement en HTML et en CSS**, sans utiliser d’autres langages comme le JavaScript.

Voilà le résultat auquel on veut parvenir :



Bien entendu, vous pourrez par la suite changer les couleurs ou le style général du menu très simplement en CSS afin que celui-ci vous convienne parfaitement.

## **Etape n°1 : La réflexion**

Comme souvent, avant de se lancer dans un développement, il faut réfléchir à ce qu’on souhaite obtenir au final afin de maximiser son efficacité et d’avoir le meilleur rendu final.

Dans notre cas, nous voulons un menu qui fasse moderne, qui soit aéré avec des couleurs qui se détachent bien les unes des autres, et surtout nous voulons que lorsqu’un visiteur passe sa souris sur l’un des onglets du menu, une liste de sous-onglets cliquables s’affichent.

Pour notre menu, nous allons utiliser des listes imbriquées les unes dans les autres que nous allons ensuite styliser en CSS.

Nous allons faire un menu avec quatre onglets ; les trois premiers onglets comportant chacun quatre sous onglets.

Pour l’affichage et le changement de couleur lors du passage de la souris, nous utiliserons la pseudo classe CSS :hover et les propriétés adaptées.

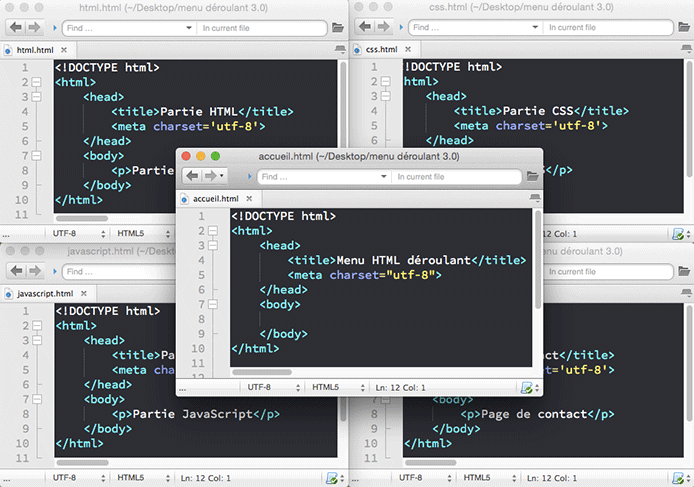
## **Etape n°2 : Le squelette HTML de notre menu déroulant**

Lorsque vous démarrez un travail en HTML / CSS, je vous recommande vivement de toujours commencer par créer le squelette HTML avant de faire une quelconque mise en forme CSS. Cela maximisera votre efficacité.

En HTML, nous allons commencer par créer cinq pages : une page d’accueil et les quatre pages correspondant aux quatre onglets de notre menu.

En théorie, nous devrions également créer les pages pour les sous-onglets mais ce serait un peu long et redondant et cela n’apporterait pas grand chose à l’exercice, donc nous allons nous en passer.

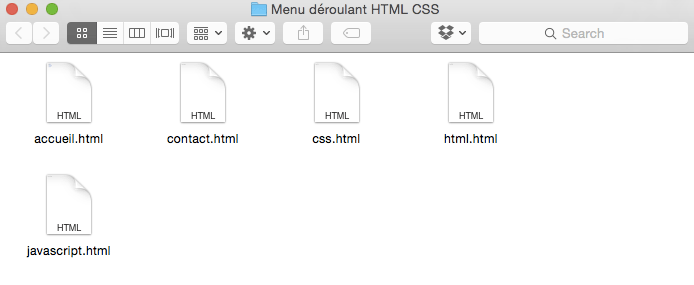
On va donc créer nos cinq pages en indiquant seulement la structure minimale en HTML pour rendre une page valide ainsi qu’un titre et une phrase pour pouvoir différencier chaque page.



Nous allons appeler ces pages :

* accueil.html ;
* html.html ;
* css.html ;
* javascript.html ;
* contact.html.

Pour plus de simplicité, nous allons enregistrer nos cinq pages dans le même dossier.



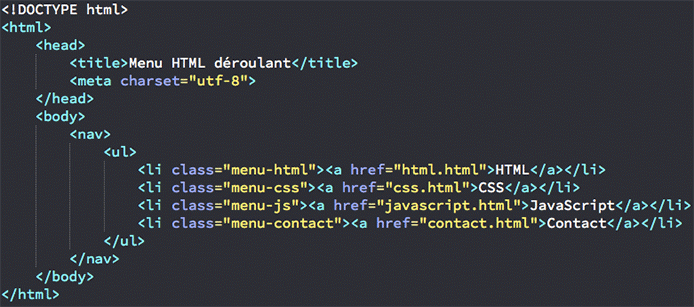
A partir de maintenant, nous allons nous concentrer sur la page accueil.html qui va accueillir notre menu.

Nous allons commencer par enfermer tout notre menu dans l’élément structurant HTML5 nav qui sert à définir le menu de navigation principal d’un site.

Ensuite, nous allons créer le premier niveau de notre menu avec les éléments ul (pour le menu en soi) et li (pour chaque onglet de notre menu).

Comme on sait que chaque onglet aura un style légèrement différent, on pense d’ores et déjà a donner un attribut class à chaque élément li.

Pour le moment, vous devriez donc arriver à cela :



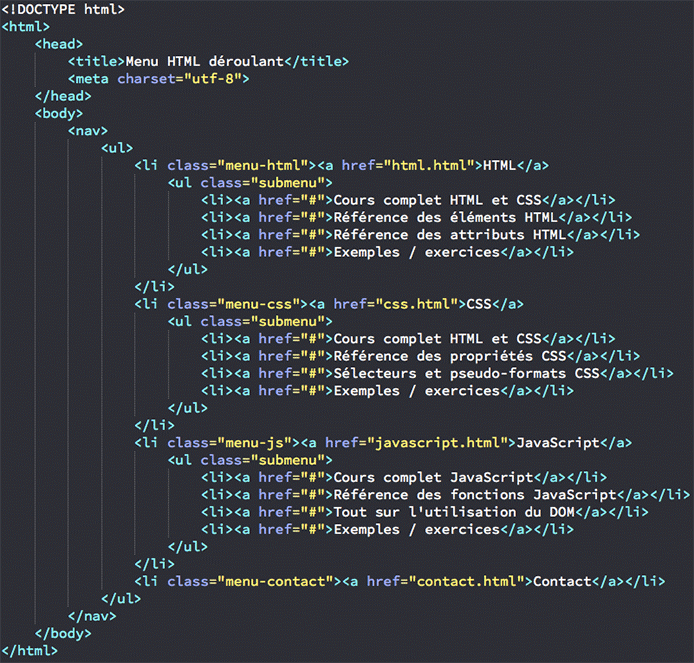
Ici, nous avons donc créé notre liste que nous allons transformer en menu, avec un lien dans chaque élément de la liste représentant chaque onglet du menu qui ramènera vers les pages que nous avons créées.

Comme indiqué, nous attribuons un attribut class à chaque élément li.

Ensuite, nous allons créer le deuxième niveau de notre menu, c’est-à-dire les sous-onglets.

Pour cela, il va nous falloir créer une nouvelle liste dans chaque élément li. Ce n’est pas très compliqué, faites juste bien attention à ne pas vous emmêler avec les balises ouvrantes et fermantes des éléments.

Comme nous comptons donner le même style à tous nos sous menus, nous allons attribuer à chacun d’entre eux un attribut class avec la même valeur.



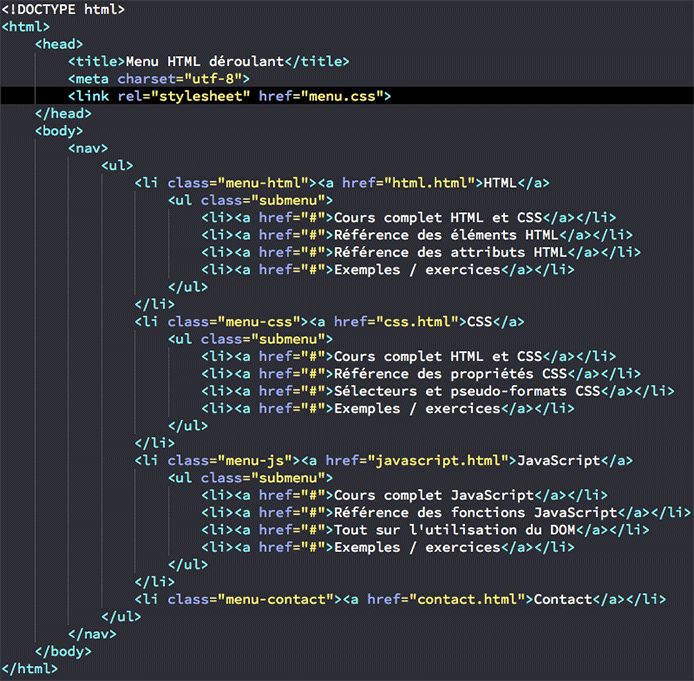
Comme je vous l’ai indiqué, nous créons une nouvelle liste à l’intérieur de chaque élément li (sauf pour la page de contact).

Ici, nous avons utilisé un « # » en valeur de href à la place de créer un véritable lien vers nos sous-onglets. Rappelez vous que nous n’avons pas créé les pages correspondantes, nous utilisons donc le dièse pour indiquer que le lien ne sera pas actif.

Et voilà tout pour la partie HTML de notre menu. Jusque là, vous ne devriez pas avoir eu de grosse surprise ni de gros problème.

Ce menu est parfaitement opérationnel : si l’on clique sur un lien, nous sommes bien amené vers la bonne page. Il a juste besoin d’un minimum de mise en forme CSS, chose que nous allons faire dès maintenant.

Pour cela, je vous invite déjà à créer un nouveau fichier que l’on va appeler menu.css et à l’enregistrer dans le même dossier que nos pages HTML. Ensuite, nous allons lier nos fichiers HTML et CSS grâce à l’élément HTML link.



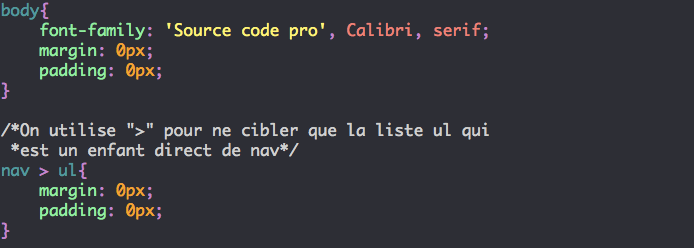
## **Etape n°3 : La mise en forme CSS de notre menu déroulant**

En CSS, il va s’agir de procéder avec un peu de méthode. Nous allons d’abord nous occuper des styles généraux, puis nous appliquerons les styles particuliers à chaque élément.

La première chose que nous allons faire est d’effectuer un reset CSS des marges intérieure et extérieure des éléments body et de notre menu (élément ul) en soi.

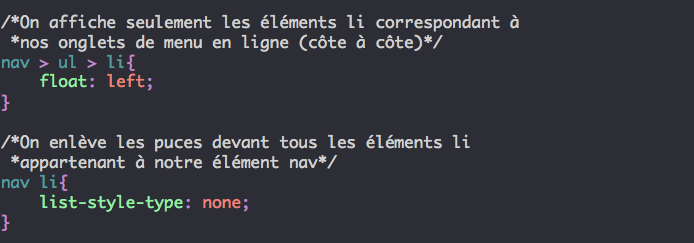
En effet, selon les navigateurs, ces éléments peuvent recevoir automatiquement des padding et margin différents, et nous voulons éviter cela.

Nous allons également en profiter pour changer la police d’écriture utilisée pour notre page avec font-family.

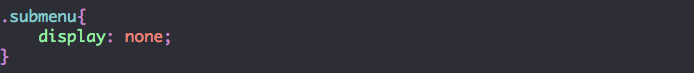


Ensuite, nous allons enlever les puces devant tous les éléments li et mettre les éléments li de notre première liste ul (c’est-à-dire nos onglets) côte-à-côte.

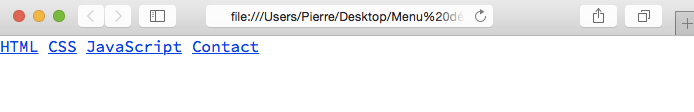
Les éléments li correspondant à nos sous onglets devront rester en colonne. Pour ne cibler que les éléments de notre première liste, nous allons utiliser le signe >, puis nous allons utiliser la propriété float.



Les sous-onglets de notre menu devront être cachés tant qu’un utilisateur ne passe pas sa souris sur l’onglet auquel ils appartiennent. Pour le moment, nous allons nous contenter de les cacher tout simplement avec display:none.



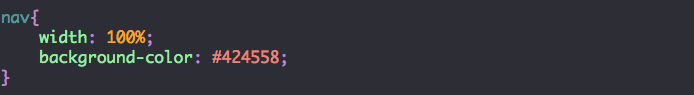
Voilà où nous en sommes pour le moment niveau rendu :



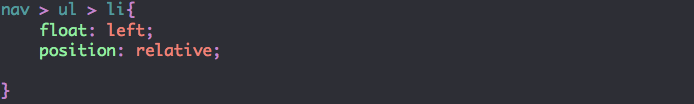
Comme vous pouvez le voir, c’est un peu mieux mais il reste du travail !

Cependant, maintenant que nous y voyons un peu plus clair, nous allons pouvoir nous occuper plus en profondeur du style des éléments composant notre menu.

Nous allons étirer notre élément nav afin que notre navigation prenne tout l’espace disponible dans la page et allons également en profiter pour lui donner une couleur de fond bleu-gris.



Ensuite, nous allons également attribuer une position:relative aux éléments li représentant nos onglets. Cela nous sera très utile plus tard pour positionner exactement les sous-onglets sous les onglets.

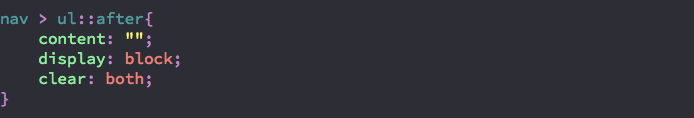


Ici, nous sommes confrontés à un problème en CSS. En effet, vous devez savoir que lorsqu’un élément parent non flottant possède des éléments enfants flottants, le CSS va faire collapser le parent qui va alors avoir une hauteur égale à zéro pixel.

Le meilleur moyen de remédier à cela est d’appliquer un overflow:hidden à l’élément parent.

Seulement, nous n’allons pas pouvoir procéder comme cela ici car cela cacherait par la suite nos sous-onglets.

Nous allons donc devoir ruser un petit peu en utilisant le pseudo élément CSS::after et les propriétés content, display et clear.

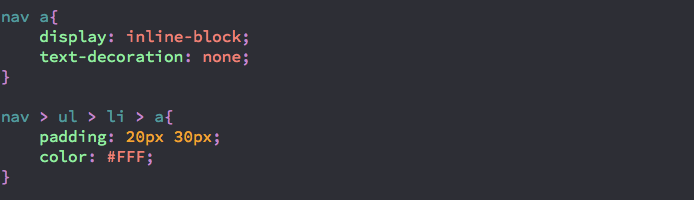


Maintenant que cela est réglé, nous allons nous préoccuper du style de nos liens en soi.

Nous allons commencer par leur ajouter une marge intérieure afin d’étendre la taille de nos onglets tout en faisant en sorte que tout l’onglet reste cliquable.

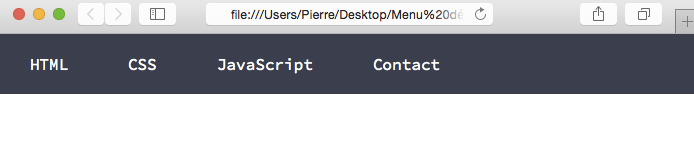
Pour que cela s’affiche bien, il va également falloir que l’on définisse un autre display pour nos éléments a.

Nous allons également changer la couleur du texte de nos liens et supprimer le trait de soulignement.



Faites toujours bien attention à cibler les bons éléments : ici, je veux enlever le trait de soulignement et changer le display de tous les liens de mon menu (onglets et sous-onglets). En revanche, je ne veux appliquer le padding et la couleur qu’aux liens de mes onglets.

Nous en avons fini avec le premier niveau de notre menu. Voilà le résultat visuel pour le moment :

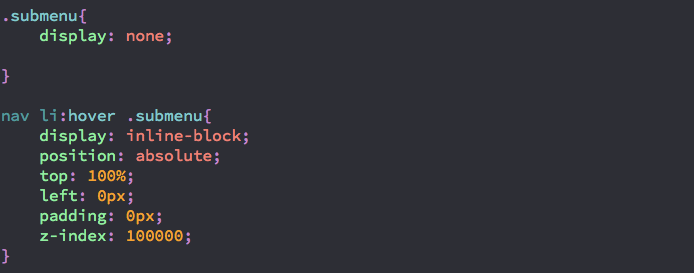


Nous allons commencer par l’affichage de nos sous-onglets. Nous voulons que ceux-ci s’affichent uniquement lorsqu’un utilisateur passe sa souris sur l’onglet correspondant.

Nous allons également positionner nos sous-onglets par rapport à nos onglets.

Pour cela, nous allons positionner nos sous-menus avec position:absolute et utiliser les propriétés top et left.

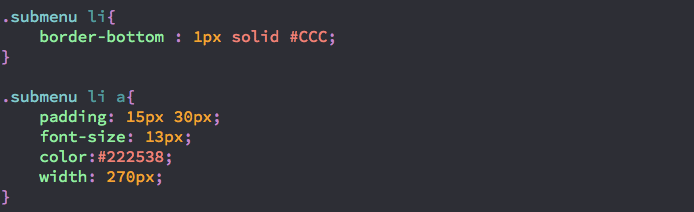
Nous allons également rajouter un z-index à nos sous-menus afin que ceux-ci s’affichent bien au dessus des contenus que vous pourriez rajouter par la suite dans vos pages.



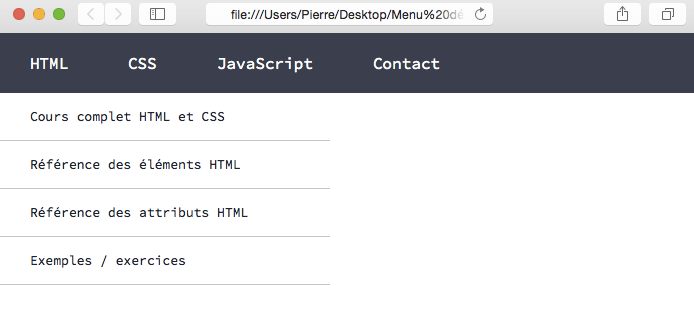
Ensuite, nous allons nous attaquer aux styles généraux de nos sous-onglets.

On va commencer par ajouter une fine bordure entre chaque élément.

En même temps, nous allons définir une largeur fixe pour nos liens de sous-menu, ajouter une petite marge intérieure, changer la taille et la couleur de l’écriture.



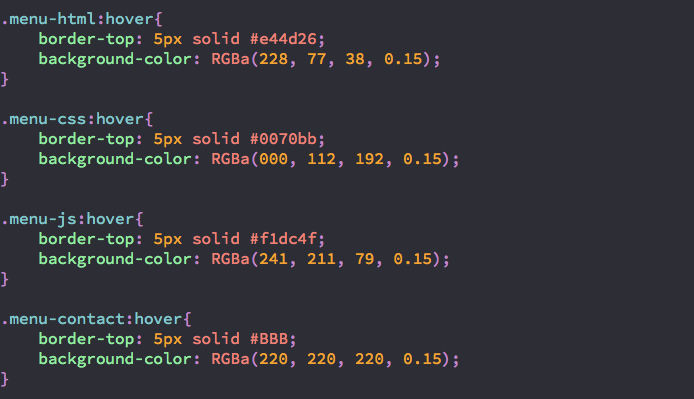
Voilà ce que ça donne :



Il ne nous reste maintenant plus qu’à effectuer un dernier travail d’habillage en attribuant des styles spécifiques à chaque onglet et sous-menu.

Je vous propose donc d’afficher une bordure haute d’une couleur différente pour chaque onglet de notre menu. Cette bordure ne devra apparaître qu’au survol de la souris.

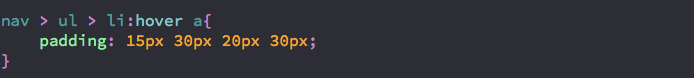
Nous allons également rajouter une couleur de fond pour chacun de nos onglets qui apparaîtra en semi transparence afin de ne pas gêner la lecture pour vos visiteurs.



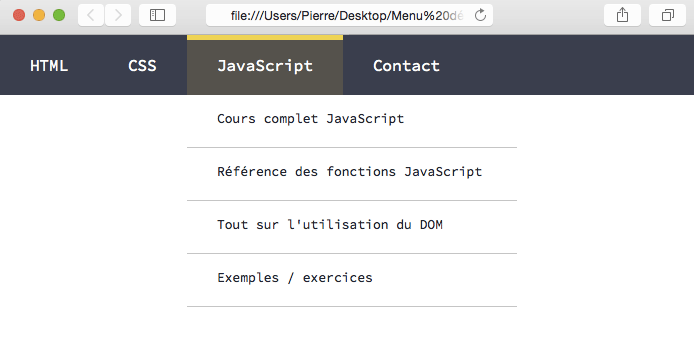
En ajoutant la bordure, on s’aperçoit que cela décale l’écriture vers le bas de la hauteur de la bordure.



Nous allons remédier à cela en réduisant la valeur du padding-top de nos liens lors du survol de la souris d’une hauteur équivalente à notre bordure.



Voilà, notre écriture reste maintenant à la même hauteur même lorsqu’on passe sa souris sur l’onglet :

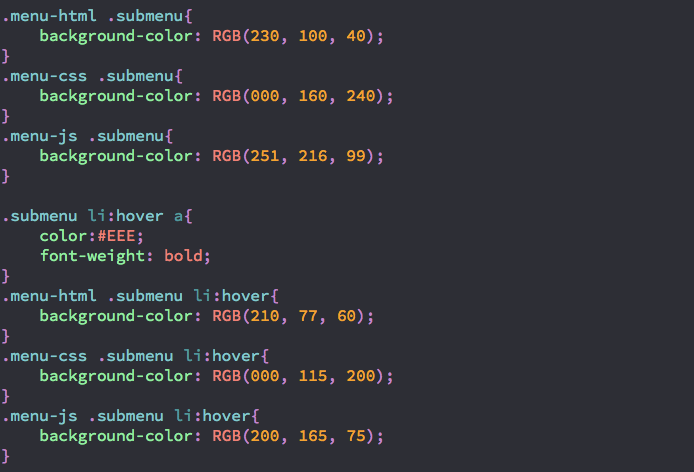


Il ne nous reste alors plus qu’à ajouter de la couleur à nos sous-onglets.

Nous allons ajouter deux nuances de couleurs différentes selon que l’utilisateur passe sa souris sur un élément en question ou non.

Lorsque l’utilisateur passe sa souris sur un élément, on veut que la couleur de fond soit plus foncée et on veut également changer la couleur de l’écriture de nos liens ainsi que le poids de la police.

Voilà l’arrangement de couleurs que je vous propose :



Et voici le résultat final :

