

## Sujet

HTML - CSS Création d'un template - 3e partie

# TABLE DES MATIÈRES

## NAVIGATION

- [1. Création d'une barre de navigation](#)
- [2. Effet au passage de la souris](#)
- [3. Les effets de transition](#)
- [4. Des liens vers des ancres](#)

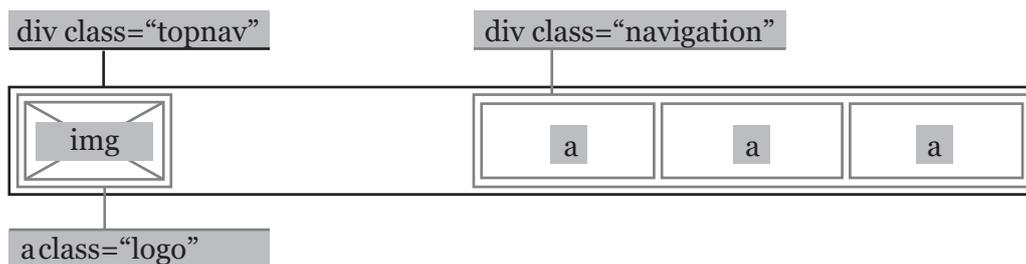
## MEDIA QUERIES

- [5. Responsive vs. Adaptive Design](#)

## NAVIGATION

### 1. Création d'une barre de navigation

Voici sous forme de wireframe, la structure de la barre de navigation que vous allez créer:



Cela donne ceci dans le code HTML. **Attention**, ce code doit être mis en haut de votre code sous la balise d'ouverture <body>.

```
<body>
  <div class="topnav">
    <a href="" class="logo">
      
    </a>
  <div class="navigation">
    <a href="">Section 1</a>
    <a href="">Section 2</a>
    <a href="">bas de page</a>
  </div>
</div>
```

## NAVIGATION (suite)



Dans le code CSS:

«`.topnav`» correspond au bloc qui contiendra votre navigation et votre logo.

- Dans ce cas-ci, «`overflow`» permet à votre bloc d'adapter ses dimensions.
- La position «`fixed`» fixe la position de la navigation.
- Avec «`top:0;`», la navigation est fixée en haut de votre page.
- Le «`z-index`» place la navigation devant le reste du contenu de votre page.

```
.topnav {
  overflow: hidden;
  position: fixed;
  z-index: 1000;
  top:0;
  width: 100%;
  background-color: #333;
}
```

Dans votre «`topnav`», vous avez un lien (a) qui porte la class «`logo`».

- Les liens, par défaut, sont en display «`inline`», il n'est donc pas possible de leur donner de mesure ou de «`margin`». Pour corriger ça, vous leur mettez la propriété «`display: block`».
- La propriété «`float:left`» permet d'aligner les éléments HTML à gauche et côte à côte.

```
.topnav a.logo{
  display: block;
  float: left;
  padding:0;
  margin: 10px;
}
.topnav a.logo img{
  height: 60px;
  width: auto;
  float: left;
}
```

Enfin, dans votre lien «`logo`» se trouve votre image.

- Donnez-lui une hauteur avec la propriété «`height`» et assurez-vous qu'il s'adapte proportionnellement avec la propriété «`width:auto;`».

Pour l'instant vous n'avez customisé que le bloc qui contient votre navigation et votre logo.

Occupons-nous de la barre de navigation à proprement dit.

- Le bloc «`navigation`» s'alignera à droite grâce à la propriété «`float:right;`».
- La propriété «`texte-decoration: none;`» supprime le soulignement des liens.

```
.navigation{
  float: right;
}
.navigation a {
  float: left;
  color: #f2f2f2;
  padding: 20px;
  margin: 10px;
  text-decoration: none;
  font-size: 17px;
}
```

## 2. Donner un effet au passage de la souris, grâce au sélecteur `:hover`

«`:hover`» s'ajoute au sélecteur pour donner un aspect différent lorsque la souris passe sur un élément.

- Pour cela il vous faut ajouter un nouveau sélecteur suivi directement de la condition `:hover`, et changer les propriétés de ce sélecteur.

```
.navigation a:hover {
  background-color: #dddddd;
  color: #000000;
}
```

## NAVIGATION (suite)

### 3. Les effets de transition



Vous pouvez adoucir l'effet du «hover» avec les propriétés de transition. La propriété «transition» permet de réaliser de simples animations à l'aide du CSS. Ajoutez cette propriété au code existant dans votre CSS.

```
.navigation a {  
  float: left;  
  color: #f2f2f2;  
  padding: 20px;  
  margin: 10px;  
  text-decoration: none;  
  font-size: 17px;  
  transition-property: all;  
  transition-duration: 1s;  
}
```

### 4. Des liens vers des ancres

Vous avez vu qu'il était possible d'ajouter un lien vers un autre site.

Exemple: `<a href="https://www.facebook.com/"></a>`.

Vous pourriez également faire un lien vers une autre page de votre site, en mettant tout simplement le nom de la page dans la référence : `<a href="page.html"></a>`

Dans le template que vous réalisez, les liens redirigent vers un endroit de la page. On dit qu'ils redirigent vers une ancre. En HTML une ancre est un «id» que vous devez mettre dans une balise.



Dans votre code HTML vous allez placer 4 «id» dans les «div» principales. Ces «div» existent déjà, vous devez juste ajouter un «id».

Aidez-vous du raccourci [ctrl]+[f] pour faire une recherche dans votre code.

```
<header id="top">  
<section id="section1">  
<section class="flex" id="section2">  
<footer id="footer">
```

Il ne vous reste plus qu'à ajouter dans votre barre de navigation les références vers vos ancras. Pour cela, dans votre lien vous devez appeler votre ancre avec un «#».

Exemple: si votre «id» se nomme «top» `<div id="top">`, vous devrez écrire votre lien comme ceci `<a href="#top">`. Encore une fois, il n'est pas nécessaire de réécrire tout le code, vous l'avez déjà fait. Remplissez seulement les href="".

```
<div class="topnav">  
  <a href="#top" class="logo">  
      
  </a>  
  <div class="navigation">  
    <a href="#section1">Section 1</a>  
    <a href="#section2">Section 2</a>  
    <a href="#footer">bas de page</a>  
  </div>  
</div>
```

# MEDIA QUERIES

## 5. Responsive vs. Adaptive Design

### Responsive design

Un site web se doit, aujourd'hui, d'être responsive. C'est à dire, qu'il peut être vu sur des écrans de tailles et de proportions différentes. Vous l'avez remarqué lorsque vous diminuez la taille de votre navigateur (Chrome...), les éléments de la barre de navigation se rapprochent, l'image en background change de taille ou disparaît partiellement.

La section avec trois colonnes est fixe, cette section ne fonctionne que sur des écrans entre 1024px et 1920px. Cette dernière section pose donc quelques problèmes sur des écrans plus petits.

Par contre la section avec deux colonnes change de taille en fonction du format que vous donnez à votre navigateur. Cette section est **responsive**.



Corrigez le problème de la section avec trois colonnes en changeant la propriété «width:1024px» de «.container» par «max-width: 1024px;». La balise <div class="container"> s'étirera sur toute la largeur de l'écran sauf si celui-ci dépasse les 1024 pixels.

```
.container{  
width: 1024px;  
margin: auto;  
padding: 20px;  
}  
↓  
.container{  
max-width: 1024px;  
margin: auto;  
padding: 20px;  
}
```

### Adaptive design

Les colonnes ne peuvent pas s'écraser indéfiniment, sans quoi le texte devient illisible.

Il va falloir adapter le layout à des écrans encore plus petits, comme ceux des smartphones.

La solution est de mettre les colonnes les unes au-dessus des autres au lieu de côte à côte.

Pour ce faire vous allez utiliser les «**media queries**».



Tout en bas de votre feuille de style, à la dernière ligne ajoutez le code:

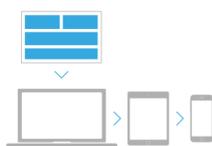
```
@media screen and (max-width: 600px) {}
```

C'est un media query qui s'applique sur les écrans plus petits que 600px.

Toutes les propriétés qui seront dans ce média query ne s'appliqueront que sur des petits écrans. Une fois que vous avez ajouté le code CSS faites le test sur Chrome, diminuez votre navigateur au maximum.

```
@media screen and (max-width: 600px) {  
  .container{  
    width: 100%;  
    padding: 0;  
  }  
  header{  
    height: auto;  
  }  
  .intro{  
    width: 100%;  
  }  
  .flex{  
    display: block;  
  }  
}
```

Le **responsive** est un design qui s'adapte automatiquement à tous les appareils.



L'**adaptive** est un template qui s'adapte à chaque appareil et de manière optimale.

