
Frameworks CSS/HTML(5)

I. Sommaire

II. Framework ?	1
III. Le contenu de Bootstrap	2
IV. La base HTML(5) de Bootstrap	2
IV.1. Exercice 4.....	3
V. Positionnement CSS, et décoration avec un Framework	3
V.1. Utilisation de la grille d'un framework.....	3
V.1.1. Qu'est-ce qu'une grille ?.....	3
V.1.2. Le cas de la grille de Bootstrap.....	4
V.1.3. Exercice 5.....	5
V.2. Quelques classes CSS spécifiques de Bootstrap.....	6
V.2.1. Des classes pour positionner sur la grille.....	6
V.2.2. Des classes typographiques et décoratives.....	6
V.2.3. Exercice 6.....	7
VI. Mise en œuvre de la navigation	8
VI.1. Une barre de menu avec Bootstrap.....	8
VI.1.1. Exercice 7.....	9
VI.2. Un fil d'Ariane avec Bootstrap.....	10
VI.2.1. Exercice 8.....	10
VII. Conception adaptative (Responsive design)	10
VII.1. Les classes « responsives » de Bootstrap.....	10
VII.1.1. Largeur des éléments.....	10
VII.1.2. Montrer/cacher des éléments.....	11
VII.2. Le menu responsive de Bootstrap.....	11
VII.3. Exercice 9.....	12
VIII. Exercice 10	12

II. Framework ?

Le développement web présente de nombreux défis. Ex. :

- faire des sites web vite (industrialiser)
- faire des sites web jolis
- faire des sites web utilisables
- faire des sites web accessibles (handicaps, ordinateurs/smartphones/tablettes, robots d'indexation, etc.)
- réutiliser les éléments déjà conçus

Un des moyens de relever ces défis est de se fonder sur les bonnes pratiques qui existent déjà pour :

- accélérer le développement
- bien faire (joli, utilisable, accessible : navigateurs, écrans, personnes, robots)
- réutiliser ce qui a déjà été réussi

Le framework est un outil qui peut être défini par :

Un framework est un cadre pour organiser et faciliter le développement grâce à des concepts clairs, des méthodes efficaces, des morceaux de code, et des éléments graphiques préconçus qui fonctionnent partout.

Ex. : Bootstrap est un framework CSS (notamment) très utilisé (développé par Twitter et Libre)

Autres ex. : [Foundation](#), [Blueprint](#), [KnackCSS](#), etc.

III. Le contenu de Bootstrap

Le framework est à récupérer ici : <http://getbootstrap.com/getting-started#download>

il contient différents fichiers ventilés dans des répertoires :

- des fichiers CSS (qui décrivent tous les blocs de règles)
- des fichiers d'icônes dans divers formats (pour compatibilité)
- des fichiers JavaScript (pour gérer la compatibilité avec tous les navigateurs et pour ajouter des fonctionnalités). Ces fichiers utilisent la bibliothèque JavaScript jQuery qu'il faut télécharger aussi : <http://code.jquery.com/jquery-2.1.1.min.js>

Certains fichiers contiennent « min » dans leur nom, et d'autres non :

- Les fichiers sans « min » sont des fichiers lisibles et contiennent des commentaires aidant à leur compréhension. Ils sont destinés au développement
- Les fichiers avec « min » sont des fichiers « minifiés », c'est à dire compressés (sans commentaires, ni espaces, etc.) qui sont utilisés pour le site en production (pour diminuer la bande passante nécessaire)

Placez le répertoire « bootstrap » dans votre répertoire de travail, ainsi qu'une copie du fichier de CV sur lequel nous travaillons. Placez dans un répertoire nommé « jquery » le fichier téléchargé ici : <http://code.jquery.com/jquery-2.1.1.min.js>

Nous sommes prêts à utiliser Bootstrap.

IV. La base HTML(5) de Bootstrap

Bootstrap propose un DOM de base pour les fichiers HTML. Il faut l'utiliser pour profiter des avantages du framework.

Voici cette base :

```

<!DOCTYPE html>
<html>
  <head>
    <title>Base de bootstrap</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
    <!-- WARNING : Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
      <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Titre du contenu</h1>
    ... contenu...
    <script src="jquery/jquery-2.1.1.min.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>
  </body>
</html>

```

On y reconnaît le doctype du HTML5 (obligatoire pour Bootstrap), et des éléments particuliers dans le **head** :

- La balise **meta** permet de définir quelle est le type d'écran par défaut
- La balise **link** renvoie vers les CSS de Bootstrap
- Les balises **script** renvoient vers les programmes en JavaScript de Bootstrap qui rendent certains navigateurs compatibles HTML5

On retrouve aussi des balises **script** dans le **body** qui permettent d'activer d'autres programmes en JavaScript de Bootstrap.

IV.1. Exercice 4

Modifiez le CV en y introduisant la base HTML5 de Bootstrap et en plaçant les fichiers (CSS, icônes, jQuery) Bootstrap où il le faut.

V. Positionnement CSS, et décoration avec un Framework

V.1. Utilisation de la grille d'un framework

V.1.1. Qu'est-ce qu'une grille ?

En conception web, comme en typographie et en publication assistée par ordinateur, une grille est un quadrillage invisible sur lequel sont calés les contenus. Elle contient donc des colonnes (col) et parfois des lignes (row).

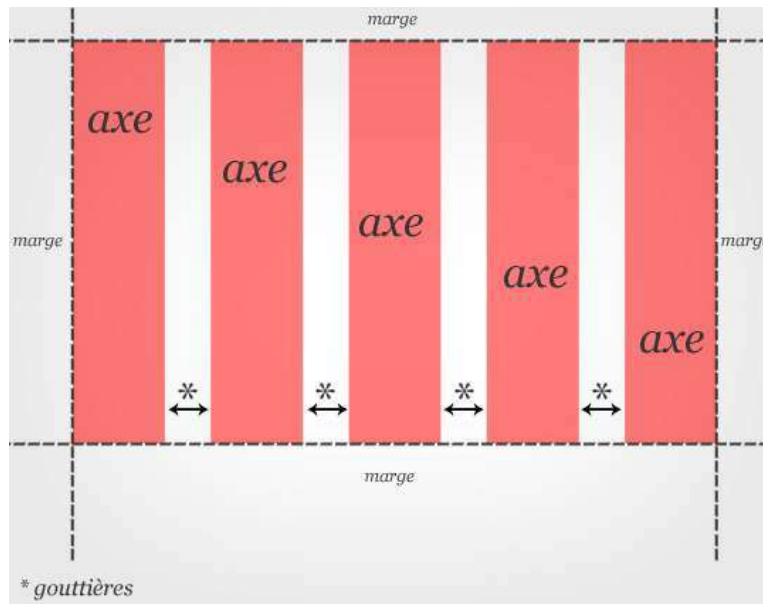


Fig. 1 : Illustration du principe de la grille, issue de <http://www.alsacreations.com/article/lire/1196-grilles-framework-css-webdesign.html> par Simon-K. Licence : [Creative Commons « Attribution - Pas d'Utilisation Commerciale - Partage à l'Identique 2.0 France \(CC BY-NC-SA 2.0\)](#)

Pourquoi utiliser une grille ?

- Elles facilitent l'organisation des contenus
- Elles permettent de respecter des proportions équilibrées entre les contenus
- Elles permettent de respecter les marges qui conviennent à la lecture (rythme des blancs)

voir aussi : <http://www.alsacreations.com/article/lire/1196-grilles-framework-css-webdesign.html> (avec des exemples concrets de types de grilles).

V.1.2. Le cas de la grille de Bootstrap

La grille de Bootstrap est composée de 12 colonnes. On l'utilise grâce à des classes CSS particulières qui déterminent la largeur d'un contenu en nombre de colonnes : `col-md-*` où « * » correspond au nombre de colonnes.

Voici un exemple avec un contenu (`<aside class="col-md-4">`) de 4 colonnes de large et composé de deux lignes (`<section class="row">`) :

```

<div class="container">
  <aside class="col-md-4">
    <section class="row">
      Voici un bloc HTML qui sera affiché sur 4 colonnes. Donc il y aura un espace de 8 colonnes libres
    </section>
    <section class="row">
      seconde rangée de contenu dans ce bloc de 4 colonnes de large
    </section>
  </aside>
</div>

```

Voici un autre exemple dans lequel les colonnes (col) sont aussi imbriquées dans les lignes (row) :

```

<div class="container">
  <aside class="col-md-4">
    <section class="row">
      <div id="zone_gauche" class="col-md-4">
        le contenu de la sous zone de 4 colonnes
      </div>
      <div id="grande_zone_droite" class="col-md-8">
        les contenu de la sous zone de 8 colonnes
      </div>
    </section>
  </aside>
</div>

```

Vous pouvez voir le résultat de ces deux exemples dans jsfiddle : <http://jsfiddle.net/595kqxfv/> (jsfiddle est site très pratique qui permet de tester et partager des morceaux de sites web).

Notez la présence d'une classe **container** qui initialise la grille. Un contenu ayant la classe **container** a une taille fixe qui dépend de la taille de la fenêtre, soit 1170 pixels, soit 970 pixels, soit 750 pixels soit la taille de la fenêtre si elle est plus petite que 750 pixels. Pour avoir un container qui prend toute la largeur, quelle que soit la taille de la fenêtre, il faut utiliser la classe **container-fluid** au lieu de **container**.

V.1.3. Exercice 5

En utilisant la grille de Bootstrap, dans le fichier exemple de CV (voir Fig. 2) :

1. Placez le menu de navigation à droite de l'accueil. Le menu doit être trois fois moins large que l'accueil.
2. Placez côte à côte les zones « Qui suis-je ? » et « Me contacter » (tailles à votre convenance)
3. Complétez le CV avec une nouvelle **<section>** consacrée au portfolio :
 - 3.1. en utilisant le Zencoding pour aller plus vite, ajoutez-y 4 entrées : « Projet IHM-Web », « Fiche de lecture UX », « Maîtrise de Zotero », et « C2i2e »
 - 3.2. Chaque entrée est structurée ainsi : une illustration (**** ou **<figure>**) de votre choix, un titre, un texte descriptif court (2 phrases maxi).
 - 3.3. Le portfolio doit figurer sous « Qui suis-je ? » et « Me contacter », mais à gauche du menu.
4. Modifiez les liens des menus pour qu'ils pointent vers les ancres du portfolio et du haut de la page (Accueil).

NB : pour trouver des images que vous avez le droit de re-publier : <http://search.creativecommons.org/?lang=fr> (n'oubliez pas de citer la licence...)



Fig. 2 : Exemple de modification du CV en utilisant la grille de Bootstrap (exercice 5)

V.2. Quelques classes CSS spécifiques de Bootstrap

Les frameworks CSS fournissent donc des classes CSS qui facilitent le travail des concepteurs. Nous venons de voir que Bootstrap utilise des classes CSS pour positionner les contenus sur la grille.

V.2.1. Des classes pour positionner sur la grille

Voici quelques exemples avec « * » correspondant à un nombre de colonnes.

- **col-md-*** permet de choisir la largeur d'un contenu en nombre de colonnes
- **row** désigne une ligne de la grille : les lignes se superposent dans une colonne
- **col-md-offset-*** permet de choisir à partir de quelle colonne commence un contenu
- **col-md-pull-*** permet de décaler un contenu de « * » colonnes à gauche de sa position normale
- **col-md-push-*** permet de décaler un contenu de « * » colonnes à droite de sa position normale

Les deux dernières classes (**col-md-pull-*** et **col-md-push-***) permettent donc de choisir l'ordre des colonnes.

V.2.2. Des classes typographiques et décoratives

D'autres classes proposées par Bootstrap permettent de s'intéresser à l'aspect des contenus et à la typographie. Voici quelques exemples :

- **h*** où « * » est compris entre 1 à 6 permet de donner le même style qu'un élément titre (<h*>) à n'importe quel contenu
- **lead** permet de mettre en évidence un contenu (il sera notamment en plus gros)
- **text-uppercase**, **text-lowercase**, **text-capitalize** permettent de choisir la casse du texte (majuscules ou non).
- **img-rounded**, **img-circle**, **img-thumbnail** permettent d'encadrer les images de différentes façons (cette classe se réserve donc à des contenus de ce type) : à bords arrondis, dans un cercle, dans une vignette.
- **jumbotron** doit contenir un élément **container** et permet de créer un bandeau avec du texte bien mis en évidence
- Des icônes, obtenues en donnant une classe **glyphicon** accompagnée d'une classe spécifique de l'icône (liste ici : <http://getbootstrap.com/components/#glyphicons>) à un élément, par exemple : `` (à tester dans jsfiddle)

Il y en a beaucoup d'autres que vous pouvez découvrir en anglais dans la documentation de Bootstrap : <http://getbootstrap.com/css/#type> ou en français dans ce tutoriel bootstrap : <http://fr.openclassrooms.com/informatique/cours/prenez-en-main-bootstrap/les-composants-integres#r-1886941>.

Faites quelques essais dans jsfiddle avant de passer à l'exercice..

V.2.3. Exercice 6

1. Sans changer sa place (vers la fin) dans le code source HTML de la page, faites en sorte que la navigation s'affiche plutôt à gauche des contenus.
2. Choisissez une image pour la photo contenue dans « Qui suis-je ? » et faites-en une image contenue dans un rond.
3. Déplacez le titre « CV en ligne de Machin Bidule » dans un **jumbotron** placé en début de page.
4. Ajoutez au moins un **glyphicon** de Bootstrap dans au moins un des **<h1>** ou **<h2>** de la page.

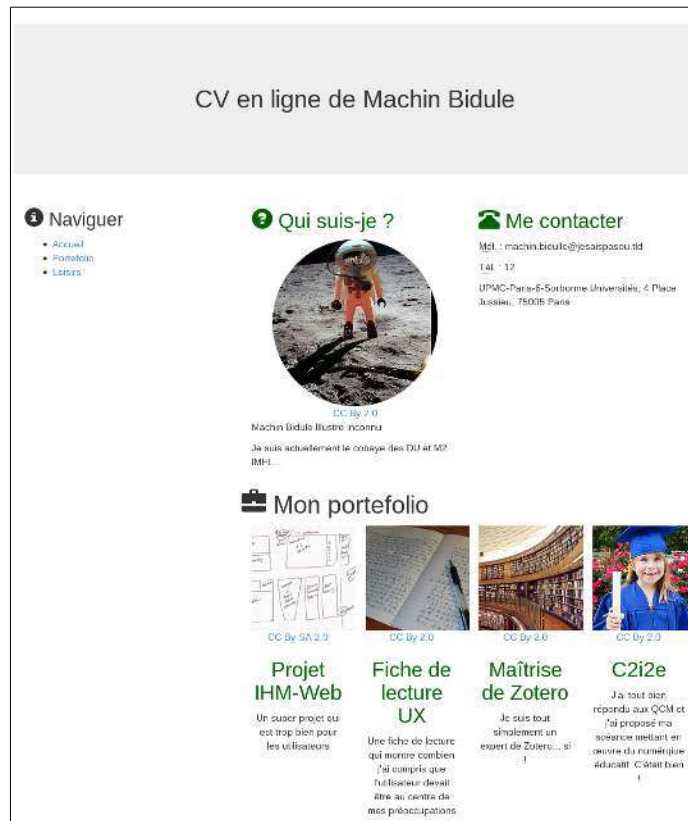


Fig. 3 : Exemple de modification du CV en utilisant les classes de Bootstrap (exercice 6)

VI. Mise en œuvre de la navigation

Sur le web, se répand actuellement une pratique qui est de mettre le menu de navigation en haut de page (contrairement à ce qui se faisait encore il y a quelques années avec des menus latéraux).

VI.1. Une barre de menu avec Bootstrap

Bootstrap, comme la plupart des frameworks CSS, propose des mécaniques pour aider à faire un tel menu.

L'élément qui contient le menu (ex. `<nav>`) devra comporter au moins deux classes :

- `navbar` qui identifie le contenu de cet élément comme une barre de navigation
- `navbar-fixed-*` ou `navbar-static-*` où « * » peut être `top` ou `bottom`. Indique si la barre de menu doit défiler avec la page (« static ») ou rester au même endroit de la fenêtre quand la page défile (« fixed »), ainsi que l'endroit où placer la barre (haut ou bas de la page)

Il faut noter que l'élément qui porte ces deux classes doit contenir une liste portant elle aussi deux classes particulières : `nav` et `navbar-nav` :


```

<ul class="nav navbar-nav">
  <li><a href="lien1">Premier élément du menu</a></li>
  <li><a href="lien2">Second</a></li>
  <li><a href="lien3">Troisième</a></li>
</ul>

```

Quelques autres classes :

- **navbar-brand** permet de titrer la barre de menu (et souvent, permet en fait de titrer le site lui-même, d'où l'appellation « brand »)
- **active** désigne l'item de la liste des éléments du menu qui est visité actuellement

Voir :

- <http://getbootstrap.com/components/#navbar>
- <http://fr.openclassrooms.com/informatique/cours/prenez-en-main-bootstrap/les-composants-integres>
- <http://bootstrapbay.com/blog/bootstrap-tutorial-navbar/>

VI.1.1. Exercice 7

Transformez le menu de navigation du CV en barre de menu qui reste en haut de l'écran même quand on fait défiler la page (voir Fig. 4).



Fig. 4 : Exemple de modification du CV intégrant une barre de menu Bootstrap (exercice 7)

VI.2. Un fil d'Ariane avec Bootstrap

Les utilisateurs ont besoin de se repérer sur les sites comprenant plusieurs pages. On utilise des fils d'Ariane pour les aider. Les frameworks CSS proposent souvent des classes pour faciliter leur mise en œuvre. Avec Bootstrap, il suffit de donner la classe **breadcrumb** à un conteneur de liste pour qu'il se présente comme un fil d'Ariane :

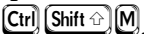
```
<ul class="breadcrumb">
  <li><a href="lien">Page d'accueil</a></li>
  <li class="active">Truc</li>
</ul>
```

VI.2.1. Exercice 8

Créez une nouvelle page HTML présentant brièvement pour votre projet IHM (ou un projet fictif) sous le menu de laquelle vous mettez un fil d'Ariane. N'oubliez pas d'intégrer le lien vers cette page dans le portfolio de la page principale du CV.

VII. Conception adaptative (Responsive design)

Le principe du « responsive design » est de faire des sites capables de s'afficher correctement et lisiblement sur tous les types d'écrans (grands et petits, portrait ou paysage, etc.).

C'est inclus par défaut dans les frameworks CSS récents comme Bootstrap. C'est facilement vérifiable grâce à la fonction dédiée de Firefox qui permet de tester un site avec différents types d'écrans. On accède à cette fonction par le raccourci .

Néanmoins, il est possible de contrôler plus finement cet automatisme pour avoir un affichage choisi pour chaque type d'écran.

VII.1. Les classes « responsives » de Bootstrap

VII.1.1. Largeur des éléments

Nous avons vu que les classes définissant la largeur des contenus sont de la forme **col-md-*** (avec « * » pour la largeur en colonnes, c'est-à-dire de 1 à 12). Il s'agit en fait de la classe qui détermine cela pour les écrans moyens (entre 992 et 1200 pixels de large), mais, il en existe d'autres pour les autres types d'écrans. Il y a :

- **col-xs-*** pour les écrans plus petits que 768 pixels de large
- **col-sm-*** pour les écrans entre 768 et 992 pixels de large
- **col-md-*** pour les écrans entre 992 et 1200 pixels de large
- **col-lg-*** pour les écrans de plus de 1200 pixels de large

Grâce à ces classes, on peut choisir pour chaque contenu de la page la façon dont ils vont s'afficher les uns par rapport aux autres en fonction du périphérique de consultation. Ex. : sur grand écran, on évitera les textes trop larges (illisibles) et donc on privilégiera les contenus rangés les uns à côté des autres. Au contraire, sur un appareil mobile à l'écran peu large, on favorisera les contenus placés les uns au-dessus des autres.

Note : on appelle « xs », « sm », « md », et « lg » des « breakpoints » car ils déterminent pour quelle taille d'écran un changement de comportement a lieu. Nous verrons plus loin que d'autres classes les utilisent.

Voir en détail les 4 types de classes : <http://getbootstrap.com/css/#grid-options> pour les différentes tailles d'écrans

VII.1.2. Montrer/cacher des éléments

D'autres classes permettent d'aller plus loin en donnant la possibilité de montrer ou de cacher certains contenus en fonction de la taille de l'écran. Les classes les plus simples à utiliser sont les classes pour cacher : **hidden-*** avec « * » à remplacer par les breakpoints de Bootstrap (« xs », « sm », « md » et « lg ») pour choisir sur quel type d'écran le contenu sera caché.

Les classes **visible-x-y** jouent le rôle inverse en permettant de montrer des contenus en fonction de la taille de l'écran. Elles prennent deux paramètres. « x » correspond au breakpoint utilisé et permet donc de choisir pour quelle taille d'écran montrer quelque chose. « y » correspond au type d'affichage à produire parmi trois possibilités : « block », « inline » et « inline-block » (voir le cours sur les CSS).

Voir : <http://getbootstrap.com/css/#responsive-utilities-classes>

VII.2. Le menu responsive de Bootstrap

Bootstrap propose aussi des fondations pour rendre les barres de menu plus agréables à utiliser sur les petits écrans : sur les grands écrans, le menu est affiché tel que nous l'avons défini précédemment, alors que sur les petits écrans il est remplacé par une icône, et le menu se déplie lorsque l'on clique (ou appuie avec son doigt) dessus.

Ce mécanisme dans Bootstrap met en jeu un programme JavaScript fourni et qui surveille dans le DOM la présence de contenus portant les deux classes **navbar-collapse** et **collapse**. Si ces classes sont présentes et que la taille d'écran atteint moins de 994px (écrans petits selon les breakpoints de Bootstrap), alors le contenu qu'elles désignent est caché au profit du contenu portant les classes **navbar-toggle** et **collapsed** ainsi que des attributs particuliers **data-toggle="collapse"** et **data-target="#bs-example-navbar-collapse-1"**. Voici un exemple concret :

```

<nav class="navbar navbar-default" role="navigation">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#bs-example-navbar-collapse-1">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Brand</a>
    </div>
    <!-- Collect the nav links, forms, and other content for toggling -->
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Link</a></li>
        <li><a href="#">Link</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">Dropdown <span
class="caret"></span></a>
          <ul class="dropdown-menu" role="menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li><a href="#">Something else here</a></li>
            <li class="divider"></li>
            <li><a href="#">Separated link</a></li>
            <li class="divider"></li>
            <li><a href="#">One more separated link</a></li>
          </ul>
        </li>
      </ul>
    </div><!-- /.navbar-collapse -->
  </div><!-- /.container-fluid -->
</nav>

```

Vous pouvez tester cette barre de menu dans jsfiddle. Vous constaterez qu'elle contient un menu dépliant qui est également rendu possible grâce à un programme en JavaScript fourni avec Bootstrap.

Plus de détails :

- <http://bootstrapbay.com/blog/bootstrap-tutorial-navbar/>
- <http://getbootstrap.com/components/#navbar>

VII.3. Exercice 9

Il s'agit de paramétrer l'adaptativité du CV :

1. Prototypiez sur papier ou avec Pencil l'aspect des contenus du CV pour toutes les tailles de breakpoints de Bootstrap (écrans de moins de 768px, de 768 à 992px, de 992 à 1200px et de plus de 1200px).
2. Modifiez le code HTML du CV pour mettre en œuvre ce que vous avez prototypé
3. Mettez en place une barre de menu responsive dans le CV

VIII. Exercice 10

Pour finir :

1. personnalisez librement le CV. Vous pourrez trouver de nombreuses sources d'inspiration en ligne, notamment : <http://www.freshdesignweb.com/personal-vcard-templates.html>
2. Postez votre travail zippé dans l'espace de dépôt dédié sur Moodle