

Premiers pas avec le framework Bootstrap

Découvrez les bases du framework [Bootstrap](#) par la pratique.



Présentation de Bootstrap

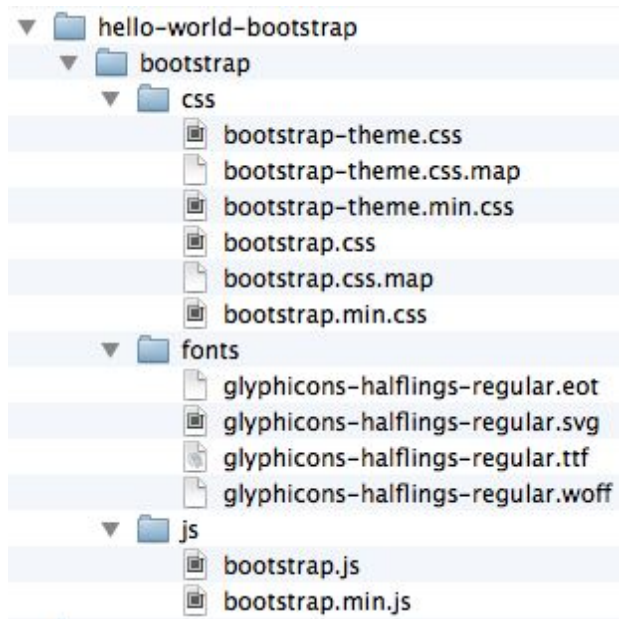
[Bootstrap](#) est un framework destiné à faciliter la création d'applications Web. Il regroupe une collection d'outils fournis sous la forme de classes CSS et de bibliothèques JavaScript. Bootstrap a été créé par deux développeurs du réseau social Twitter.

Prise en main de Bootstrap

Installation

Pour pouvoir utiliser Bootstrap, il est nécessaire que le framework soit accessible pour nos pages Web. Pour cela, on peut y accéder en ligne via un [CDN](#) (*Content Delivery Network*) ou bien l'installer localement. Nous allons utiliser cette seconde solution.

Créez dans votre répertoire personnel un sous-répertoire nommé **hello-world-bootstrap** qui sera le répertoire racine de notre exemple. Dans ce répertoire, créez un sous-répertoire **bootstrap**. Ensuite, téléchargez la version 3.3.7 de Bootstrap sur [cette page](#). Ensuite, décompressez le contenu de l'archive dans le répertoire **bootstrap** créé précédemment. Vous devez obtenir l'arborescence ci-dessous.



On observe que le framework se compose de définitions CSS et JavaScript ainsi que d'images compressées. Les fichiers contenant le mot **min** correspondent aux versions minimales des sources, compressées mais peu lisibles pour le développeur. On les utilise souvent en production afin de réduire le volume de données nécessaire pour afficher une ressource Web.

Une première page Web

Créez dans le répertoire **hello-world-bootstrap** un fichier HTML nommé **index.html**. Recopiez le code ci-dessous dans ce fichier.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <title>Hello world with Bootstrap</title>
  </head>
  <body>
    <h1>Hello world!</h1>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script
>
    <script src="bootstrap/js/bootstrap.min.js"></script>
  </body>
</html>
```

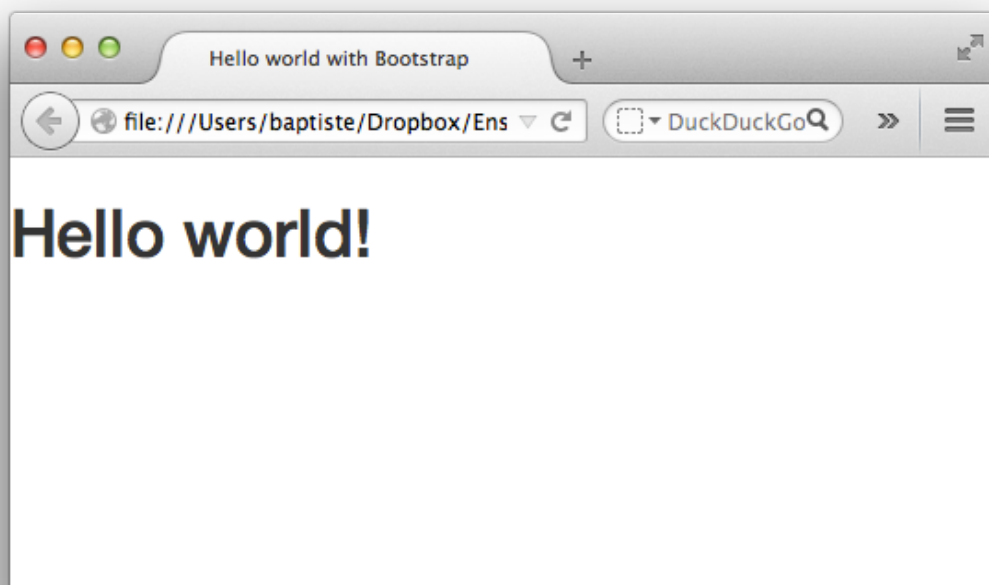
Il s'agit d'une page Web classique avec quelques éléments spécifiques :

- une balise `<link>` vers la feuille de style Bootstrap ;
- une balise `<meta name="viewport">` définissant le *viewport*, c'est-à-dire la surface de la fenêtre du navigateur qui affiche la page. Cela va permettre à la page de s'adapter à la dimension du terminal client ;
- une balise `<meta>` permettant un affichage correct sur le navigateur Internet Explorer ;
- des balises `<script>` qui activent les plugins jQuery de Bootstrap. Ces plugins sont nécessaires pour animer les composants Bootstrap.

Remarque

jQuery est ici inclus en ligne via le CDN Google. Il est également possible de le télécharger et de l'installer localement.

Ouvrez cette page dans votre navigateur Web favori. Vous obtenez le résultat ci-dessous.



Principe de fonctionnement

Bootstrap est avant tout une bibliothèque CSS qui fournit plusieurs classes CSS prédéfinies. Il suffit de réutiliser ces classes pour profiter des fonctionnalités de Bootstrap.

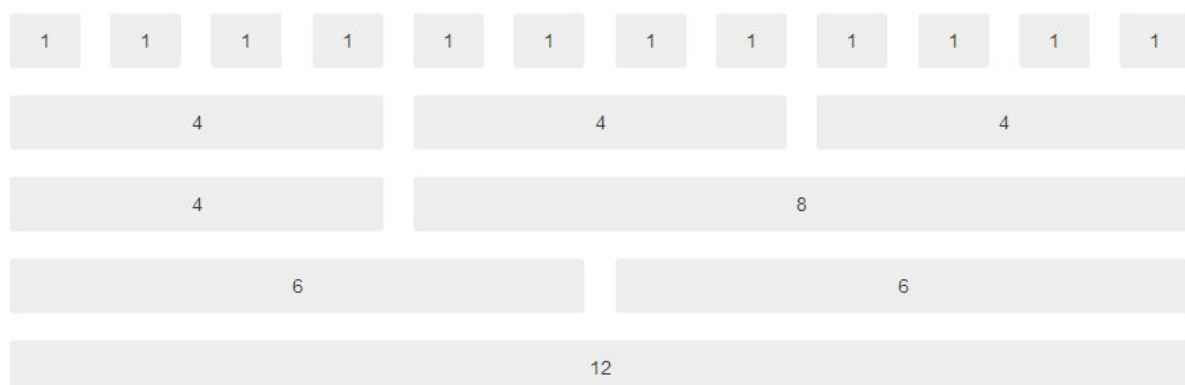
Le conteneur et la grille Bootstrap

Bootstrap fournit une classe CSS `container` utilisée pour regrouper d'autres éléments.

```
<div class="container">
  <!-- ... -->
</div>
```

A l'intérieur d'un `container`, Bootstrap gère la zone d'affichage sous la forme d'une **grille** de 12 colonnes. La classe Bootstrap `row` crée une **ligne** dans cette grille, et chaque classe Bootstrap `col-*` crée une **cellule** qui regroupe * colonnes dans cette ligne.

Dans chaque ligne, la taille totale des colonnes doit être égale à 12. On peut donc choisir de diviser la surface d'affichage en 12 colonnes de taille 1, ou bien en 3 colonnes de taille 4, ou encore en une colonne de taille 4 et une colonne de taille 8.



Démonstration

Créez dans le répertoire `hello-world-bootstrap` un fichier CSS `style.css` contenant le code ci-dessous.

```

.row {
    margin-bottom: 15px;
}
.row [class^="col-"] {
    padding-top: 10px;
    padding-bottom: 10px;
    background-color: #eee;
    border: 1px solid #ddd;
    background-color: rgba(86,61,124,.15);
    border: 1px solid rgba(86,61,124,.2);
}

```

Ce fichier permettra de mieux visualiser les éléments de la grille Bootstrap.

Ensuite, éditez le fichier `index.html` comme indiqué ci-dessous.

```

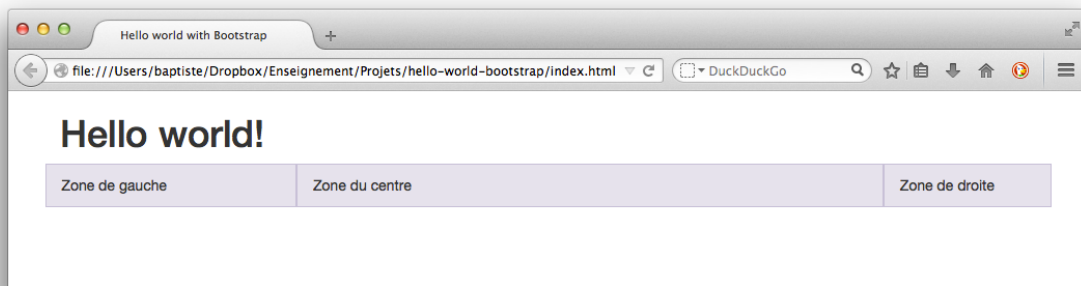
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link href="style.css" rel="stylesheet">
    <title>Hello world with Bootstrap</title>
</head>
<body>
    <div class="container">
        <h1>Hello world!</h1>
        <div class="row">
            <div class="col-md-3">
                Zone de gauche
            </div>
            <div class="col-md-7">
                Zone du centre
            </div>
            <div class="col-md-2">
                Zone de droite
            </div>
        </div>
    </div>

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script
>
    <script src="bootstrap/js/bootstrap.min.js"></script>
</body>
</html>

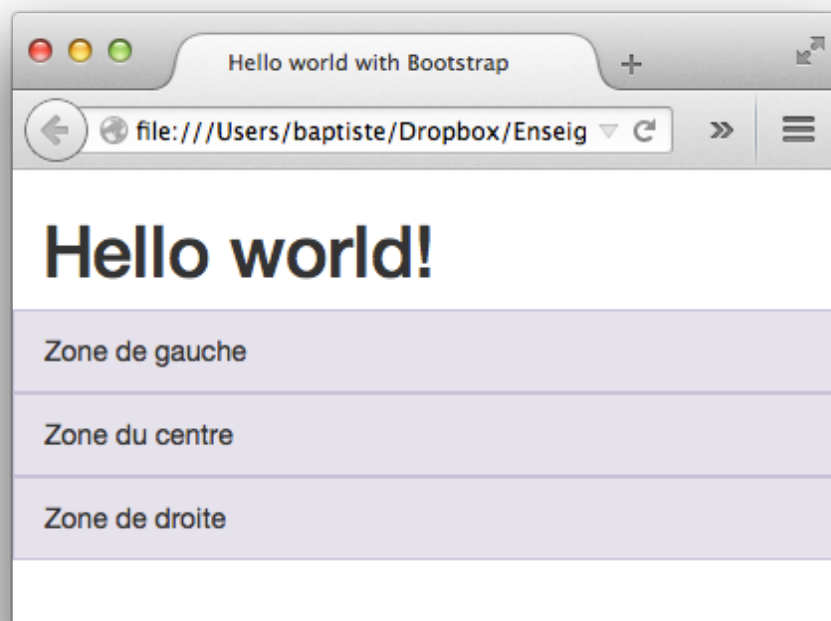
```

On remarque au passage que le total des tailles des colonnes (3+7+2) est bien égal à 12. Ouvrez cette page dans un navigateur Web. Le placement des colonnes s'adapte à la surface d'affichage disponible.

Voici le rendu avec une surface d'affichage importante.



Voici le rendu lorsque la surface d'affichage diminue.



Bootstrap et le design Web adaptatif

D'où vient le comportement que l'on vient d'observer ? Depuis sa version 3, Bootstrap est pensé d'abord pour les terminaux mobiles dont la surface d'affichage est limitée. Il adapte automatiquement la disposition des éléments à la zone d'affichage, détectée grâce à la balise `<meta name="viewport">` vue plus haut.

Pour chaque taille de colonne (de 1 à 12), Bootstrap fournit quatre classes `col-*` dont le comportement diffère en fonction du terminal d'affichage utilisé pour afficher la page Web.

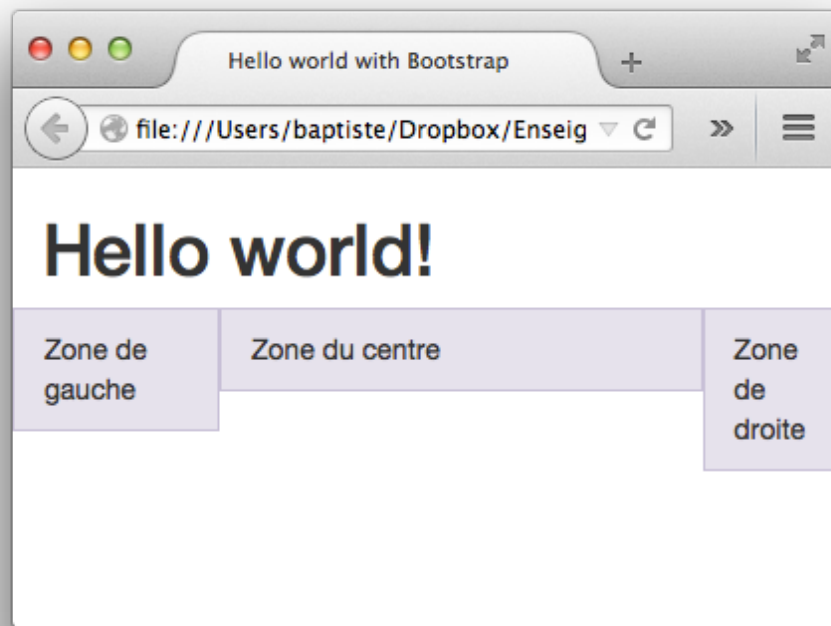
	Smartphones	Tablettes	Desktop	Desktop large
Limite de largeur d'affichage	< 768px	>= 768 px	>= 992 px	>= 1170 px
Préfixe des classes colonne	<code>col-xs-*</code>	<code>col-sm-*</code>	<code>col-md-*</code>	<code>col-lg-*</code>
Placement des colonnes	Horizontal	(*)	(*)	(*)

(*) : Les colonnes seront placées horizontalement (côte à côte) uniquement si la largeur d'affichage est suffisante.

Par exemple, la classe Bootstrap `col-md-6` désigne une colonne de taille 6 affichée horizontalement uniquement si la largeur d'affichage est au minimum celle d'un écran de bureau (desktop). Dans le cas contraire, la colonne sera placée verticalement, c'est-à-dire positionnée sous les autres.

Dans notre exemple, nous avons utilisé des colonnes desktop (`col-md-*`). Le tableau ci-dessus montre que ces colonnes ne sont placées horizontalement que si la largeur d'affichage est supérieure ou égale à 992px. Dans le cas contraire, elles se superposent verticalement. Cela explique le comportement observé.

Modifiez le fichier `index.html` et remplacez les classes `col-md-*` par des classes `col-xs-*` de même taille. Ouvrez ce fichier dans un navigateur. Quelle que soit la surface d'affichage, les colonnes restent placées horizontalement.



Comme indiqué dans le tableau ci-dessus, les colonnes `col-xs-*` ne sont jamais superposées verticalement.

Aller plus loin

Il est possible de contrôler plus finement l'organisation de la page Web en combinant plusieurs classes colonne. Imaginons par exemple qu'on souhaite garder notre design actuel pour un affichage sur écran de bureau, mais agrandir la zone de gauche et diminuer la taille de la zone centrale pour un affichage sur tablette. Voici le code à modifier dans le fichier `index.html`.


```

<!-- ... -->
<div class="row">
  <!-- Taille 4 sur tablettes, 3 sur desktop et plus -->
  <div class="col-sm-4 col-md-3">
    Zone de gauche
  </div>
  <!-- Taille 6 sur tablettes, 7 sur desktop et plus -->
  <div class="col-sm-6 col-md-7">
    Zone du centre
  </div>
  <!-- Taille 2 sur tablettes et plus -->
  <div class="col-sm-2">
    Zone de droite
  </div>
</div>
<!-- ... -->

```

Modifiez `index.html` puis affichez-le dans un navigateur. On constate que la colonne de gauche passe de 25% à 33% de la largeur totale lorsque la surface d’affichage correspond à celle d’une tablette.

Remarque

On vérifie bien que la somme des tailles des colonnes est égale à 12 dans chaque configuration d’affichage (4+6+2 en mode tablette, 3+7+2 en desktop et plus).

Il est même possible de masquer ou de faire apparaître certains éléments en fonction de la surface d’affichage grâce aux classes `visible-*` et `hidden-*`. Par exemple, on souhaite qu’en contexte smartphone, la zone de droite soit masquée et que les zones de gauche et du centre prennent chacune 50% de la place disponible. Voici le code source associé.

```

<!-- ... -->
<div class="row">
  <!-- Taille 6 sur smartphones, 4 sur tablettes, 3 sur desktop et plus -->
  <div class="col-xs-6 col-sm-4 col-md-3">
    Zone de gauche
  </div>
  <!-- Taille 6 sur smartphones et tablettes, 7 sur desktop et plus -->
  <div class="col-xs-6 col-md-7">
    Zone du centre
  </div>
  <!-- Masquée sur smartphones, taille 2 sur tablettes et plus -->
  <div class="col-sm-2 hidden-xs">
    Zone de droite
  </div>
</div>
<!-- ... -->

```

Vous pouvez tester ce code pour observer le comportement décrit plus haut.

Conclusion

En remplaçant la gestion CSS manuelle du positionnement (`float : left;`, etc) par l'utilisation des classes Bootstrap `row` et `col-*`, on rend la mise en page Web à la fois plus simple et plus souple.

Les exemples ci-dessus ne donnent qu'un aperçu des possibilités de Bootstrap. Consultez la [documentation](#) du framework pour plus de détails.

Aperçu de quelques composants Bootstrap

Outre les classes de mise en page déjà étudiées, Bootstrap fournit de nombreux composants prêts à être intégrés dans nos pages Web.

Images et icônes

Bootstrap est livré avec environ 200 mini-images (*glyphicons*) librement utilisables.

```
Je boirais bien un <span class="glyphicon glyphicon-glass"></span> avant  
d'aller écouter de la <span class="glyphicon glyphicon-music"></span> avec ma  
<span class="glyphicon glyphicon-heart"></span>
```

Ce qui donne le résultat suivant :

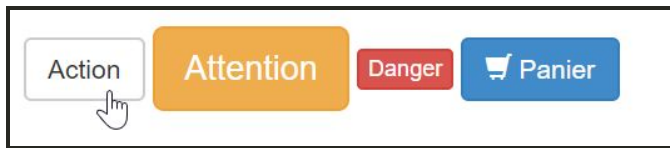
Je boirais bien un 🍷 avant d'aller écouter de la 🎵 avec ma ❤️

De plus, Bootstrap fournit une classe `img-responsive` qui permet aux images (balises HTML ``) de se redimensionner automatiquement à la taille de leur conteneur parent.

Boutons

Bootstrap fournit un moyen simple d'obtenir des boutons élégants et personnalisables.

```
<button type="button" class="btn btn-default">Action</button>
<button type="button" class="btn btn-warning btn-lg">Attention</button>
<button type="button" class="btn btn-danger btn-xs">Danger</button>
<button type="button" class="btn btn-primary"><span class="glyphicon
glyphicon-shopping-cart"></span> Panier</button>
```



Messages

Bootstrap permet de définir des messages faciles à repérer dans une page Web.

```
<div class="alert alert-success">Bravo !</div>
<div class="alert alert-info">Terminé</div>
<div class="alert alert-warning">Attention...</div>
<div class="alert alert-danger">Erreur !</div>
```

Bravo !

Terminé

Attention...

Erreur !

Menus de navigation

Il existe deux possibilités pour concevoir un menu de navigation :

- utiliser la classe `nav` ;
- utiliser la classe `list-group` ;

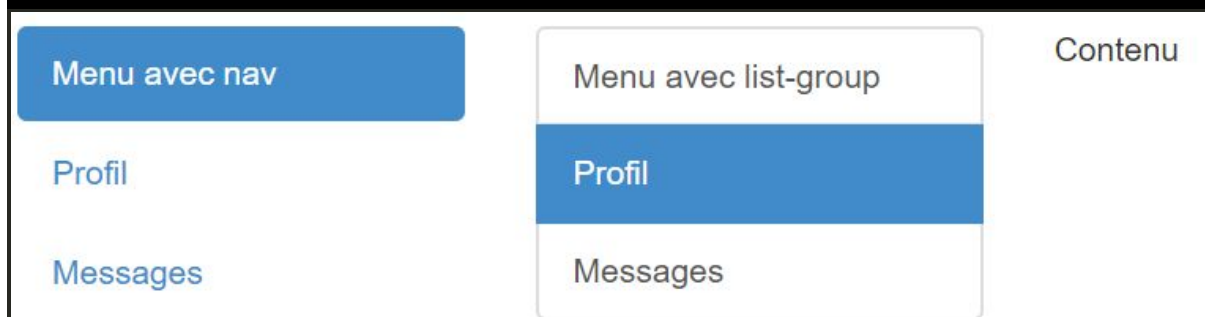
Le code ci-dessous définit une mise en page sur trois colonnes, avec deux menus et une zone de contenu.

```
<div class="row">
  <div class="col-xs-3">
    <ul class="nav nav-pills nav-stacked">
```

```

        <li class="active"><a href="">Menu avec nav</a></li>
        <li><a href="">Profil</a></li>
        <li><a href="">Messages</a></li>
    </ul>
</div>
<div class="col-xs-3">
    <div class="list-group">
        <a class="list-group-item" href="">Menu avec list-group</a>
        <a class="list-group-item active" href="">Profil</a>
        <a class="list-group-item" href="">Messages</a>
    </div>
</div>
<div class="col-xs-6">
    Contenu
</div>
</div>

```



Barre de navigation

On peut ajouter facilement une barre de navigation horizontale grâce à la classe Bootstrap `navbar`.

```

<nav class="navbar navbar-default" role="navigation">
    <a class="navbar-brand" href="#">Mon site Web</a>
    <ul class="nav navbar-nav">
        <li class="active"><a href="#">Rubrique 1</a></li>
        <li><a href="#">Rubrique 2</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
        <li><a href="#">Rubrique 3</a></li>
    </ul>
</nav>

```



Il est également possible d'obtenir une barre de navigation adaptative. Son contenu est masqué et remplacé par un simple bouton d'accès lorsque la surface d'affichage est insuffisante.

```
<div class="navbar navbar-default" role="navigation">
  <!-- Partie de la barre toujours affichée -->
  <div class="navbar-header">
    <!-- Bouton d'accès affiché à droite si la zone d'affichage est trop
    petite -->
    <button type="button" class="navbar-toggle" data-toggle="collapse"
    data-target="#navbar-collapse-target">
      <span class="sr-only">Activer la navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="#">Mon site Web</a>
  </div>
  <!-- Partie de la barre masquée si la surface d'affichage est insuffisante
  -->
  <div class="collapse navbar-collapse" id="navbar-collapse-target">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Rubrique 1</a></li>
      <li><a href="#">Rubrique 2</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
      <li><a href="#">Rubrique 3</a></li>
    </ul>
  </div>
</div>
```

Mon site Web



Conclusion

Nous n'avons fait que survoler les possibilités de Bootstrap. Il existe de nombreuses ressources sur ce framework en ligne, à commencer par sa [documentation officielle](#) ou encore les tutoriels des sites [OpenClassrooms](#) et [Codecademy](#).

