



NIVEAU DE DIFFICULTÉ :



PRÉREQUIS :





## Table des matières

Présentation .....	4
Un peu d'histoire.....	4
Prérequis.....	4
Développement responsive et mobile first.....	4
Avantages / Inconvénients .....	5
Avantages .....	5
Inconvénients .....	5
Installation.....	6
Utiliser un CDN.....	6
Layout et fonctionnalités principales.....	8
Système de grille : Savoir compter jusqu'à 12 !.....	8
Taille d'écrans et « responsivité » .....	10
Fonctions pratiques .....	12
Largeur du conteneur .....	12
Utiliser plusieurs rangées.....	12
Sauter des colonnes.....	13
Imbrication.....	14
Application de style .....	15
Exemple de typographie.....	15
Modification du texte.....	17
Composants.....	19
Conclusion & alternatives .....	23

## Présentation

---

### Un peu d'histoire

**Bootstrap** a été conçu et élaboré par Mark OTTO et Jacob THORTON, ingénieurs chez **Twitter**, pour être utilisé en interne. L'idée de son développement était de permettre à tous les développeurs de **Twitter** d'utiliser la même méthodologie de codage afin d'éviter que chacun travaille à sa manière, et par conséquent, de diminuer le risque de bugs ou d'incohérences dans le fonctionnement de l'application.

Son utilisation a été un véritable succès, puisqu'elle a permis d'augmenter très largement la productivité des employés de Twitter. Compte tenu de ces résultats et de son potentiel, **Bootstrap** est sorti en août 2011 comme projet open-source sur **Github**.

Le phénomène **Bootstrap** a rapidement pris une incroyable ampleur qui a permis à de nombreux développeurs de contribuer et d'améliorer le framework.

Aujourd'hui, 8 ans après sa sortie, Bootstrap est devenu **LE framework CSS** le plus utilisé au monde pour le développement de sites responsives et orientés mobiles-first. Son succès réside dans sa facilité d'utilisation et surtout par sa capacité à construire et styliser rapidement une page web.

### Prérequis

Bootstrap est donc un **framework CSS**. Il est donc recommandé avant de se lancer dans son apprentissage de connaître les bases de HTML, CSS et avoir si possible quelques notions en Javascript puisque nous utiliserons diverses propriétés liées directement à ces langages.

Ce cours reste une initiation à **Bootstrap**, dont l'utilisation est très large et très complète. Je vous inviterai régulièrement à consulter la documentation officielle pour répondre à vos besoins spécifiques ( <https://getbootstrap.com/docs/4.3/getting-started/introduction/> )

### Développement responsive et mobile first

Le principal intérêt de **Bootstrap**, c'est qu'il permet très facilement et rapidement de produire et de rendre esthétique du contenu web aussi bien sur desktop, tablette ou smartphone : c'est ce que l'on appelle un **site RESPONSIVE**.

Pour permettre ce rendu, Bootstrap se base initialement sur une combinaison de **HTML**, **CSS** et **Javascript**. Au fur et à mesure de son évolution, plusieurs plugins ont été ajoutés avec notamment l'ajout de la **bibliothèque jQuery** qui a augmenté grandement le nombre de fonctionnalités.

Son **système de grille** (nous le verrons plus en détails un peu plus tard) est simple et efficace pour élaborer l'architecture d'une page web.

### Avantages / Inconvénients

#### Avantages

- Prise en main et **apprentissage rapide** pour une utilisation basique.
- **Compatible** avec tous les navigateurs.
- **Large communauté** et par conséquent beaucoup de solutions aux problèmes rencontrés.
- Mise en place rapidement d'une structure de page et d'ajout de style sans feuille **CSS**.
- Permet d'obtenir facilement un site **responsive**.

#### Inconvénients

- Apprentissage relativement long pour une utilisation avancée compte tenue de toutes les possibilités offertes par les **classes Bootstrap**.
- **jQuery** est une bibliothèque lourde à charger et par conséquent, les performances ne sont pas optimales pour la rapidité de chargement d'une page.
- Le style de **Bootstrap** est prédéfini, il peut parfois donner l'impression que tous les sites se ressemblent.

L'utilisation de **Bootstrap** est donc un choix. Un choix qui varie en fonction de son projet et du temps que vous avez à y accorder.

Vous pouvez rapidement développer un site web responsive et stylisé mais qui ne « sortira pas du lot » ou vous pouvez opter pour un développement de votre site en pur **HTML/CSS/JS**, mais cela prendra beaucoup plus de temps, mais le résultat sera unique.

Maintenant passons au concret et regardons comment installer et configurer Bootstrap dans un projet.

## Installation

L'installation de Bootstrap (Version 4 aujourd'hui) est relativement simple. Il y a plusieurs solutions qui comportent chacune leurs avantages et inconvénients. Dans ce cours, nous verrons la méthode la plus simple : l'utilisation d'un CDN.

### Utiliser un CDN

Le CDN (qui signifie « Content Delivery Network ») est un lien qui donne accès directement à une librairie sans avoir à l'installer dans son projet.

Cette solution présente plusieurs avantages :

- Actualisation automatique de la librairie,
- Amélioration du référencement,
- Très simple à installer

En effet, l'installation d'un CDN se fait très facilement. Il suffit d'ajouter une ou plusieurs balises `<link>` pour le CSS et `<script>` pour les fichiers javascripts. Pour trouver les balises correspondantes, il faut de la même manière se référer à la documentation :

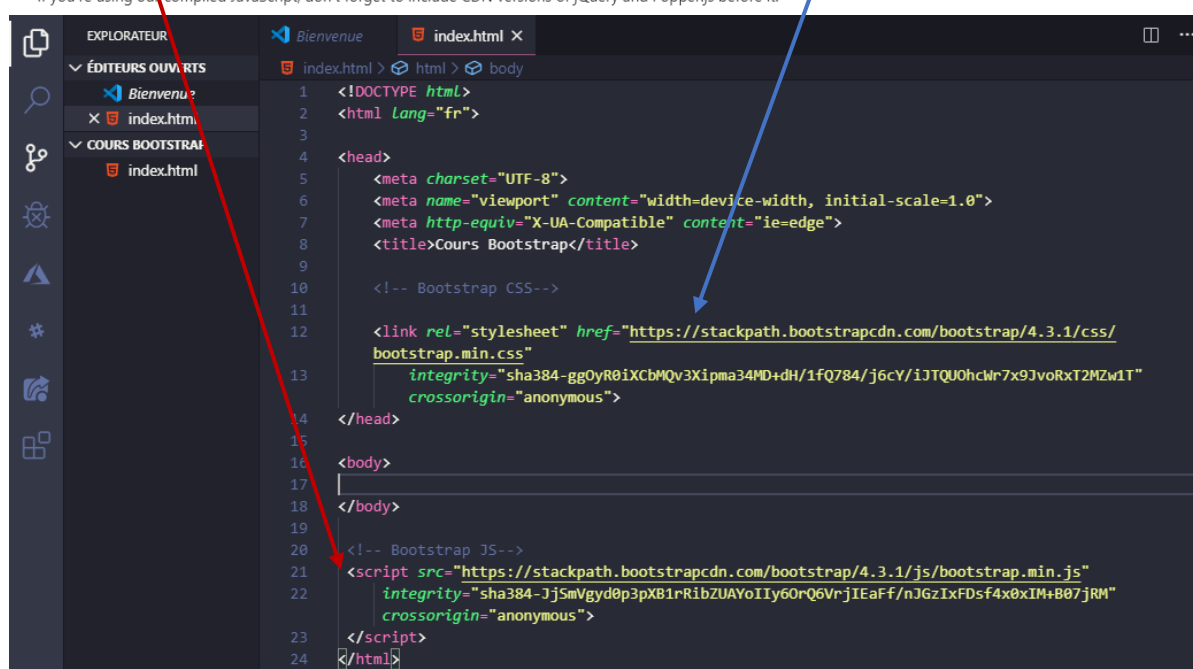
On ajoute donc le CDN CSS de Bootstrap dans la balise `<head>` de votre projet et le CDN Javascript après la fermeture de votre balise `<body>`:

### BootstrapCDN

Skip the download with [BootstrapCDN](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQV3Xipma34MD+dH/1fQ784/j6cY/IJTQUOHCw7x9JvoRxT2MZw1T" crossorigin="anonymous">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pX81rR1bZUAyOIIy60rQ6VrjIEaFf/nJGzIx4x0xIM+B07jRM" crossorigin="anonymous"></script>
```

If you're using our compiled JavaScript, don't forget to include CDN versions of jQuery and Popper.js before it.



Mais cela ne suffit pas. Pour faire activer toutes les fonctionnalités offertes par Bootstrap, il faut ajouter les CDN Javascript de jQuery, et Popper.js :

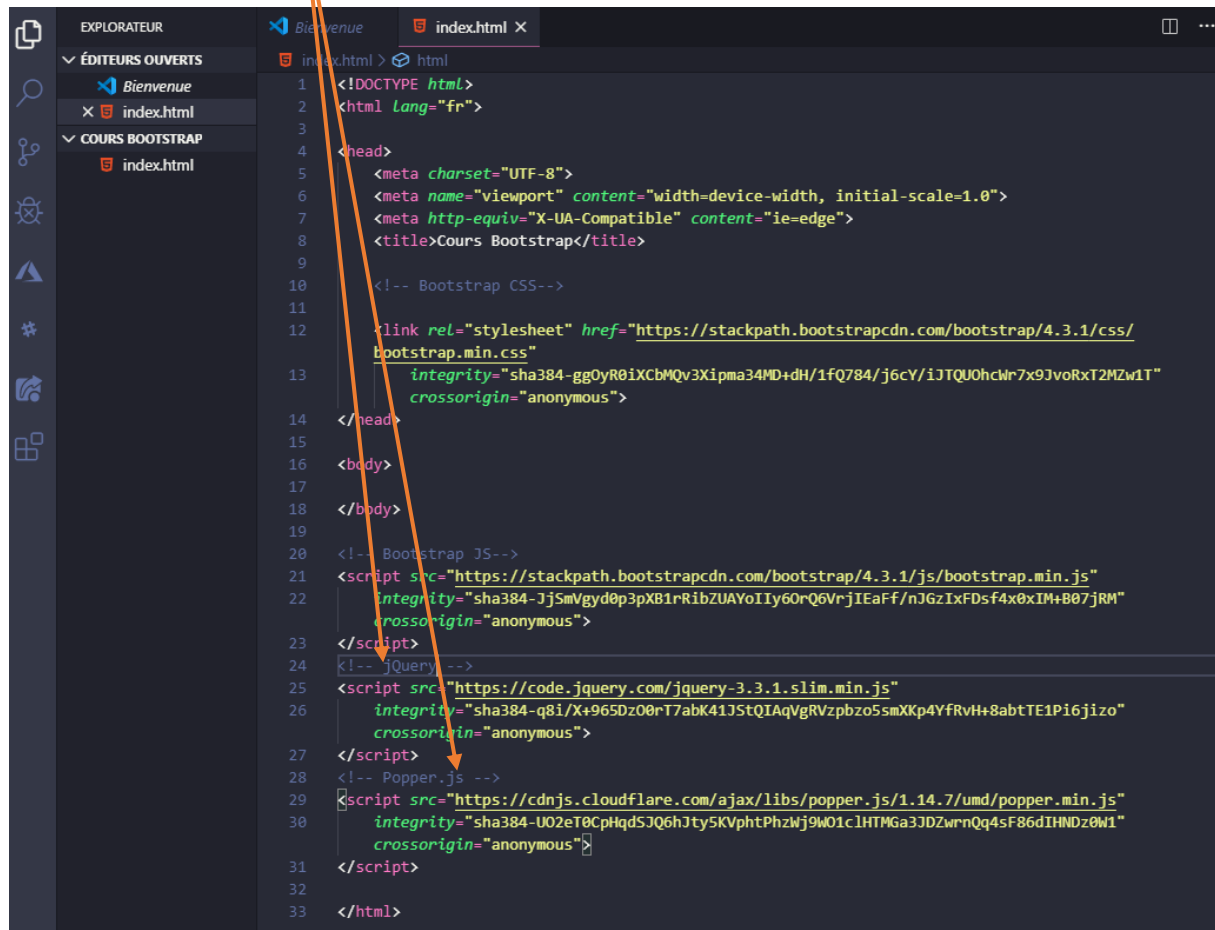
## BootstrapCDN

Skip the download with [BootstrapCDN](#) to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQV3Xipma34MD+dH/1f184" data-bbox="130 212 876 243"/>
```

If you're using our compiled JavaScript, don't forget to include CDN versions of jQuery and Popper.js before it.

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" data-bbox="130 286 876 315"/>
```



Et voilà ! Bootstrap est installé et prêt à l'emploi.

## Layout et fonctionnalités principales

### Système de grille : Savoir compter jusqu'à 12 !

La facilité d'organisation et d'architecture d'une page web est l'un des principaux avantages de Bootstrap. Le concept repose sur découpage simple, il suffit en fait de savoir compter jusqu'à 12 !

12 comme le nombre de colonnes dans le tableau ci-dessus. Voici plus en détail la structure :


- La partie bleue représente ce que l'on appelle le « conteneur » ou « container » en anglais,
- La partie orange, représente, elle, une « rangée » ou « row » de la page
- Et enfin la partie verte, correspond à une des 12 « colonnes » ou « column » de la page.

Les trois parties sont liées entre elles. Il faut toujours un conteneur pour ajouter une rangée, et toujours une rangée pour ajouter une colonne. Pour mettre en place chaque partie dans notre code, il suffit de créer une balise <div> et d'ajouter une classe correspondant à la partie souhaitée :

```
<div class="container">  
  <div class="row"></div>  
    <div class="col-md-6">  
      <h1>Hello Everybody</h1>  
    </div>  
  </div>  
</div>
```

C'est la manipulation de ces différents éléments qui vont nous permettre de construire notre page web.



Si vous avez été observateur, vous remarquerez que nous avons ajouté « md-6 » à la colonne.

Nous verrons la signification des lettres « md » juste après, mais concernant le chiffre « 6 », cela représente le nombre de colonnes que nous voulons donner à la <div>.

Pour rappel, peu importe la largeur de l'écran, il y a toujours 12 colonnes dans Bootstrap ! Dans cet exemple, nous demandons à la <div> de prendre la moitié de la page. En ajoutant quelques propriétés CSS pour distinguer les différentes parties, voici le résultat :



- La zone bleue représente bien le conteneur qui englobe notre page,
- La zone orange la rangée qui couvre la largeur du conteneur,
- Et enfin en vert, les 6 colonnes sélectionnées sur les 12 maximales.

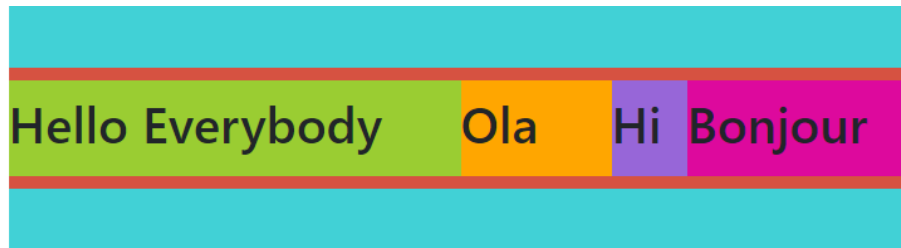
Ajoutons d'autres éléments :

```
<div class="container">
  <div class="row">
    <div class="col-md-6">
      <h1>Hello Everybody</h1>
    </div>
    <div class="col-md-2">
      <h1>Ola</h1>
    </div>
    <div class="col-md-1">
      <h1>Hi</h1>
    </div>
    <div class="col-md-3">
      <h1>Bonjour</h1>
    </div>
  </div>
</div>
```

Maintenant dans cet exemple, nous avons utilisé les 12 colonnes possibles dans la rangée du conteneur :

- Col-md-6
- Col-md-2
- Col-md-1
- Col-md-3

Résultat :  $6+2+1+3= 12$  ! Le compte est bon !



Passons maintenant à la signification de « md » !

### Taille d'écrans et « responsivité »

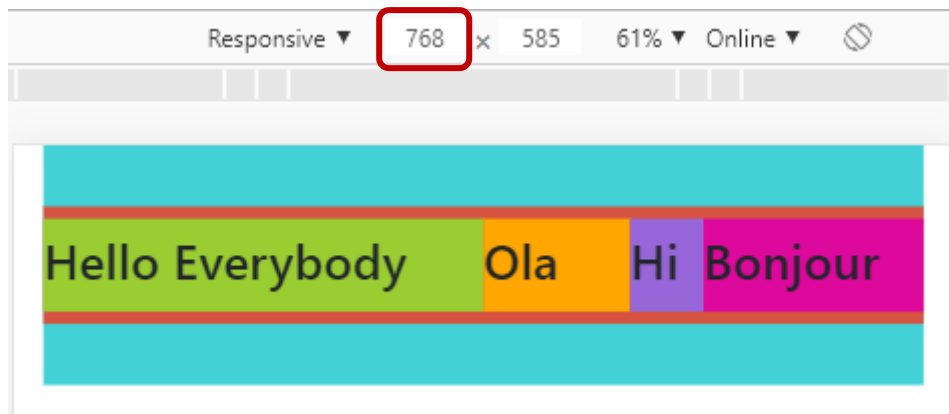
Tout le responsive de Bootstrap repose sur l'ajustement du nombre de colonne occupées par une `<div>` en fonction de la taille de l'écran utilisé pour consulter une page web. Pour « jouer » avec la taille des colonnes nous ajoutons donc deux lettres à notre `<div class= 'col'>`, il en existe trois principaux :

- **COL-XS-\*** : Extra-small pour les petits écrans (smartphone), redimensionnement pour des écrans inférieurs à 768px.
- **COL-SM-\*** : Small, pour les écrans réduits (tablette), redimensionnement pour des écrans supérieurs ou égaux à 768px.
- **COL-MD-\*** : Medium pour les écran moyens (ordinateur portable), redimensionnement pour des écrans supérieurs ou égaux à 992px.
- **COL-LS-\*** : Large pour des grands écrans (ordinateurs de bureau), redimensionnement pour des écrans supérieurs ou égaux à 1200px.

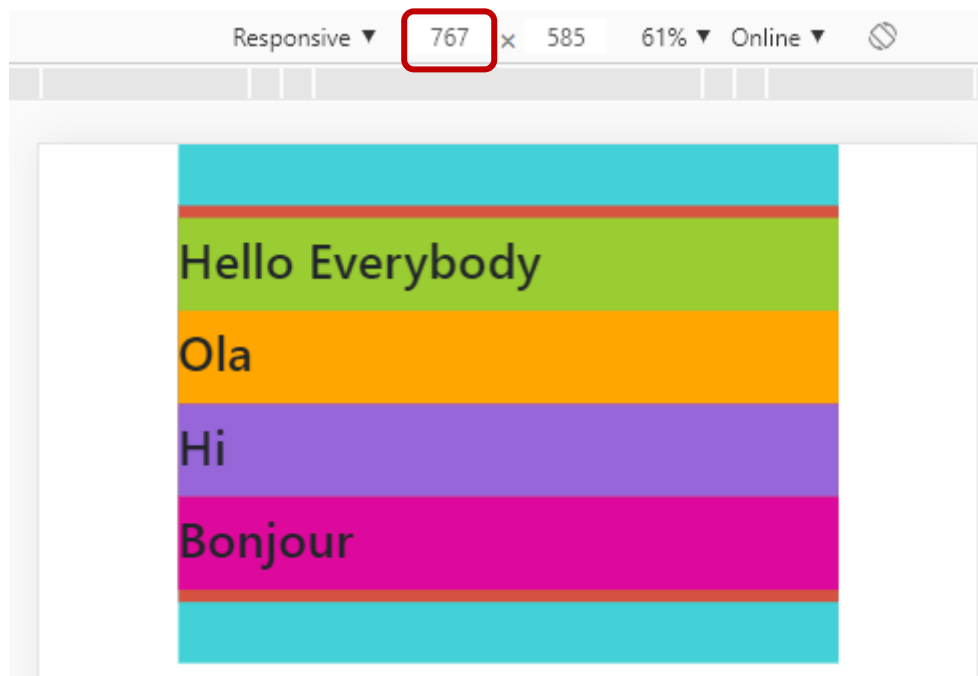
Voyons tout ça avec un cas pratique.

Si nous reprenons notre exemple précédent, nous pouvons constater que nos différentes `<div>` se répartissent sur la largeur du conteneur puisque le total des

colonnes est égal à 12. Cette structure reste telle quelle pour tous les écrans supérieurs à 767px :



Si nous réduisons la largeur d'un seul pixel, les `<div>` se réorganisent pour s'adapter à un écran et petite taille :



Nous pouvons donc remarquer ici, que par défaut, lorsque nous utilisons la classe `COL-MS-*`, et que l'écran se réduit en dessous de 768px, la classe se « transforme » en `COL-SM-12` afin de prendre toute la largeur.

Maintenant que vous avez compris les bases du redimensionnement et de la puissance du responsive de Bootstrap, voyons quelques fonctionnalités plus avancées dans la manipulation de la structure d'une page web.

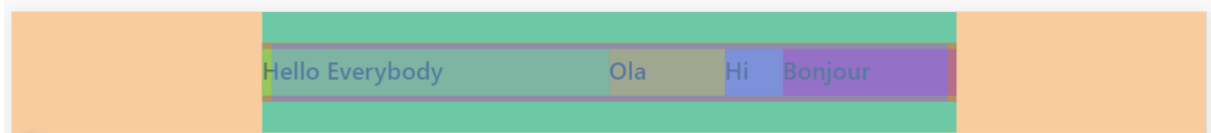
## Fonctions pratiques

### Largeur du conteneur

Vous l'avez peut-être déjà remarqué, mais par défaut, le conteneur possède une marge à gauche et à droite. Pour permettre au conteneur de prendre toute la largeur de l'écran il faut modifier la classe du conteneur :

- **Avant :**

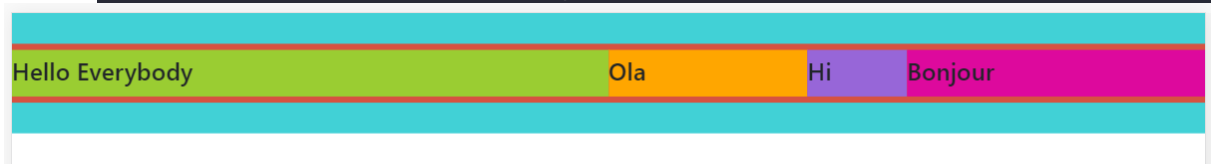
```
<div class="container"></div>
```



Les zones orangées correspondent à ces marges.

- **Après :**

```
<div class="container-fluid"></div>
```



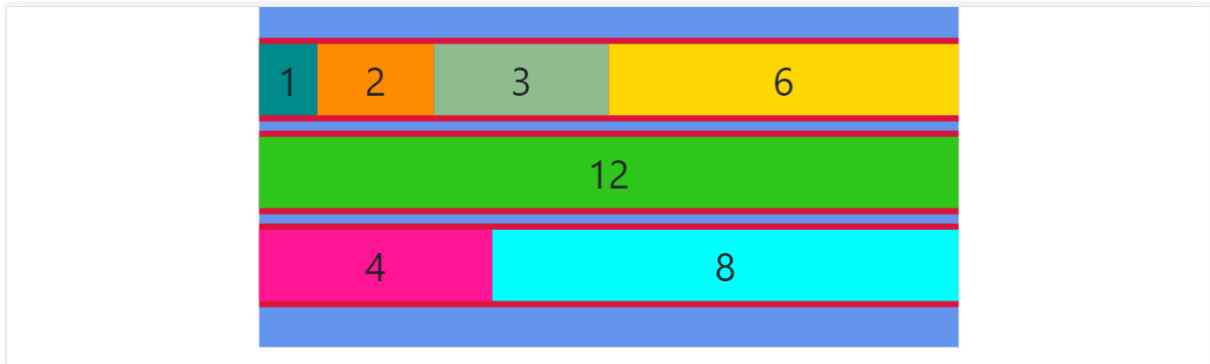
La classe « container-fluid » permet donc d'étirer les <div> sur toute la largeur de la rangée.

### Utiliser plusieurs rangées

Pour l'instant, nous n'avons utilisé qu'une seule rangée dans notre architecture. Mais bien évidemment, nous pouvons ajouter autant de rangées que nécessaires, et modifier la disposition de chaque colonne pour chacune d'entre elles :

```
<div class="container">
  <div class="row">
    <div class="col-lg-1">1 </div>
    <div class="col-lg-2">2 </div>
    <div class="col-lg-3">3 </div>
    <div class="col-lg-6">6 </div>
  </div>
  <div class="row">
    <div class="col-lg-12">12 </div>
  </div>
  <div class="row">
    <div class="col-lg-4">4 </div>
    <div class="col-lg-8">8 </div>
  </div>
</div>
```

Dans les lignes de codes ci-dessus, nous avons un conteneur et trois rangées (« row »). Dans chaque rangée, nous avons ajouté plusieurs colonnes de taille différente, mais dont le total est toujours 12 :

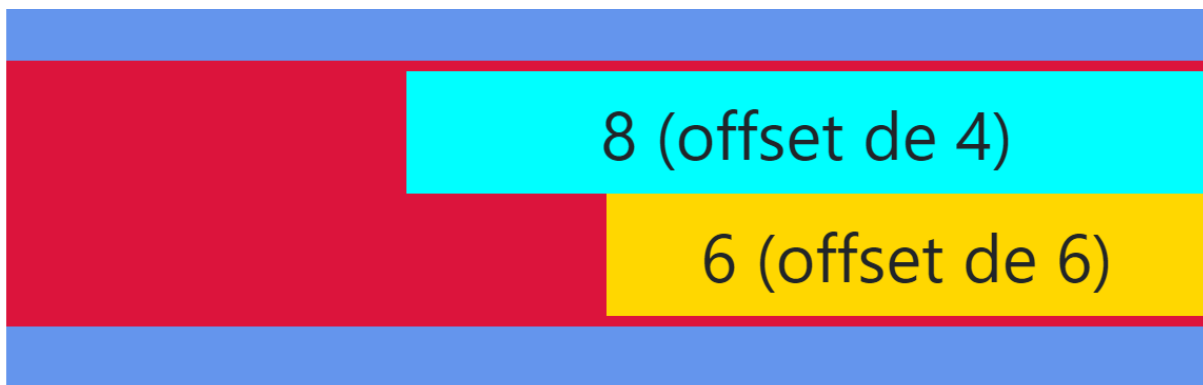


De cette manière, vous pouvez vraiment disposer vos éléments comme vous le souhaitez dans votre page !

### Sauter des colonnes

Pour sauter des colonnes, Bootstrap a créé la classe « offset ». Cette classe permet tout simplement d'ajouter une marge à gauche :

```
<div class="container">
  <div class="row">
    <div class="col-lg-8 offset-md-4">8 (offset de 4)</div>
    <div class="col-lg-6 offset-md-6">6 (offset de 6)</div>
  </div>
</div>
```



Autre exemple, si vous créez une colonne dont l'addition du nombre de colonne avec le nombre de l'offset multiplié par 2 est égale à 12, cela permet de centrer horizontalement l'élément (euuuuuuh ?)

Cas pratique pour illustrer cette torture mathématique :

```
<div class="container">
  <div class="row">
    <div class="col-lg-8 offset-md-2">8 (offset de 2)</div>
  </div>
</div>
```



Vous remarquez ici que nous avons un bloc de 8 colonnes et un offset de 2. Si vous multipliez par deux l'offset, vous obtenez 4.

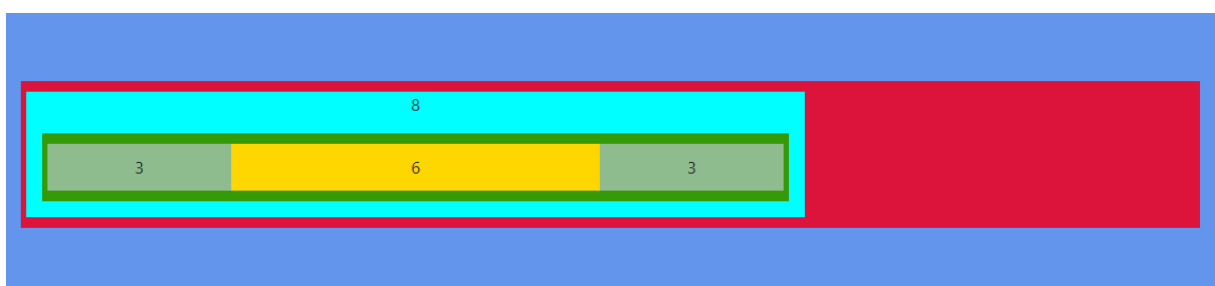
$8+4=12$  !! De cette manière, votre bloc de trouve centré. Il possède 8 colonnes et il est séparé respectivement par 2 colonnes à gauche et 2 colonnes à droite !

### Imbrication

La plupart des structures que nous avons vu pour l'instant restent relativement simples. Pour complexifier un peu l'organisation spatiale d'une page web, nous pouvons imbriquer les éléments les uns avec les autres !

En résumé, dans une colonne, vous pouvez par exemple, ajouter une nouvelle rangée qui comportera elle-même de nouvelles colonnes :

```
<div class="container">
  <div class="row">
    <div class="col-lg-8">8
      <div class="row">
        <div class="col-lg-3">3 </div>
        <div class="col-lg-6">6 </div>
        <div class="col-lg-3">3 </div>
      </div>
    </div>
  </div>
</div>
```



Dans l'exemple ci-dessus, nous avons :

- Toujours le conteneur en bleu qui englobe l'ensemble de la structure
- En rouge, la première rangée
- Dans cette rangée, il y a un bloc de 8 colonnes
- Dans ce bloc, nous avons une nouvelle rangée en vert,
- Et enfin dans cette rangée, il y a 3 blocs (deux de 3 colonnes et 1 de 6 colonnes)

Le nombre de combinaison est infinie, et permet vraiment de personnaliser sa page.

Maintenant que vous connaissez les rudiments de l'organisation d'une page avec Bootstrap, voyons comment appliquer du style sans utiliser de feuille CSS.

## Application de style

Bootstrap permet d'appliquer très rapidement et facilement du style, sans utiliser de CSS personnalisé.

Nous allons voir quelques exemples, mais je vous conseille d'explorer l'étendue des possibilités offertes par Bootstrap à travers la documentation officielle.

## Exemple de typographie

### Display

La classe « display » permet de mettre en avant des titres :

```
<div class="container">
  <div class="row">
    <div class="col-lg-12">
      <div class="display-1">Titre 1</div>
    </div>
    <div class="col-lg-12">
      <div class="display-2">Titre 2</div>
    </div>
    <div class="col-lg-12">
      <div class="display-3">Titre 3</div>
    </div>
    <div class="col-lg-12">
      <div class="display-4">Titre 4</div>
    </div>
  </div>
</div>
```



### Citation

Pour mettre en avant une citation (ou « blockquote » en anglais), il vous suffit d'ajouter la classe « blockquote » de Bootstrap :

```
<div class="container">
  <div class="row">
    <div class="col-lg-12">
      <blockquote class="blockquote">
        <p>Chuck Norris ne ment pas, c'est la vérité qui se trompe.</p>
        <footer class="blockquote-footer">Chuck Norris</footer>
      </blockquote>
    </div>
  </div>
</div>
```





## Modification du texte

### Alignement du texte

Pour aligner le texte d'un élément, rien de plus simple, il suffit d'ajouter les classes :

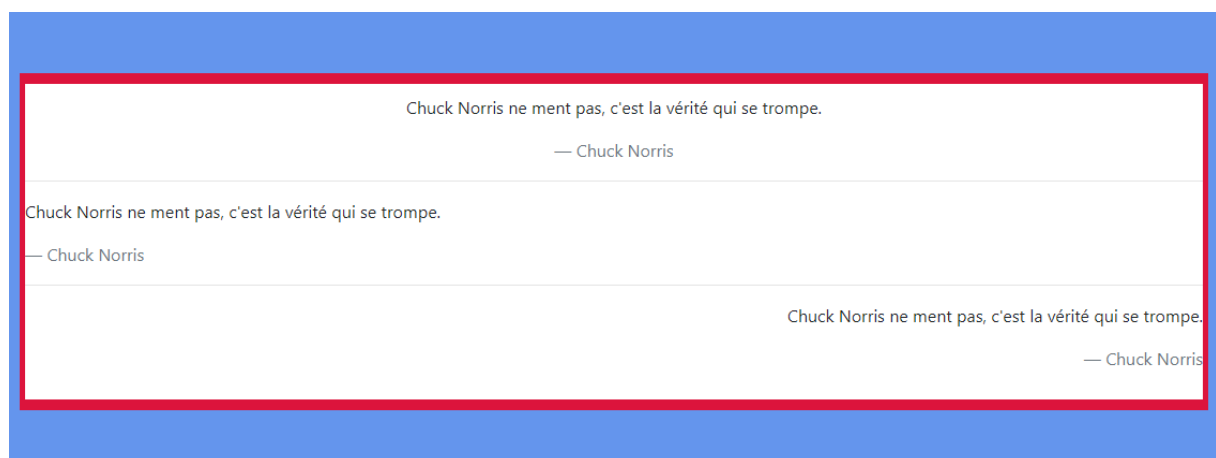
- « text-center » : pour centrer un texte
- « text-right » : pour aligner à droite
- « text-left » : pour aligner à gauche

```
<div class="container">
  <div class="row">
    <div class="col-lg-12">
      <blockquote class="blockquote text-center">
        <p>Chuck Norris ne ment pas, c'est la vérité qui se trompe.
      </p>

      <footer class="blockquote-footer">Chuck Norris</footer>
    </blockquote>
    <hr>
    <blockquote class="blockquote text-left">
      <p>Chuck Norris ne ment pas, c'est la vérité qui se trompe.
    </p>

    <footer class="blockquote-footer">Chuck Norris</footer>
  </blockquote>
  <hr>
  <blockquote class="blockquote text-right">
    <p>Chuck Norris ne ment pas, c'est la vérité qui se trompe.
  </p>

  <footer class="blockquote-footer">Chuck Norris</footer>
</blockquote>
    </div>
  </div>
</div>
```



## Style du texte

Également pour appliquer du style à du texte, vous n'avez qu'à ajouter les différents styles dans la classe de votre élément :

```
<div class="container">
  <div class="row">
    <div class="col-lg-12">
      <p class="font-weight-bold">Ce texte est magnifique en gras</p>
      <p class="font-weight-normal">Ce texte est magnifique en normal</p>
      <p class="font-weight-light">Ce texte est magnifique en police légère.</p>
      <p class="font-italic">Ce texte est magnifique en italique</p>
    </div>
  </div>
</div>
```

Ce texte est magnifique en gras

Ce texte est magnifique en normal

Ce texte est magnifique en police légère.

*Ce texte est magnifique en italique*

Pareil pour la classe « text-transform » :

```
<div class="container">
  <div class="row">
    <div class="col-lg-12">
      <p class="text-lowercase">Ce texte est magnifique en majuscule</p>
      <p class="text-uppercase">Ce texte est magnifique en minuscule</p>
      <p class="text-capitalize">Ce texte est magnifique en capitalisé.</p>
    </div>
  </div>
</div>
```

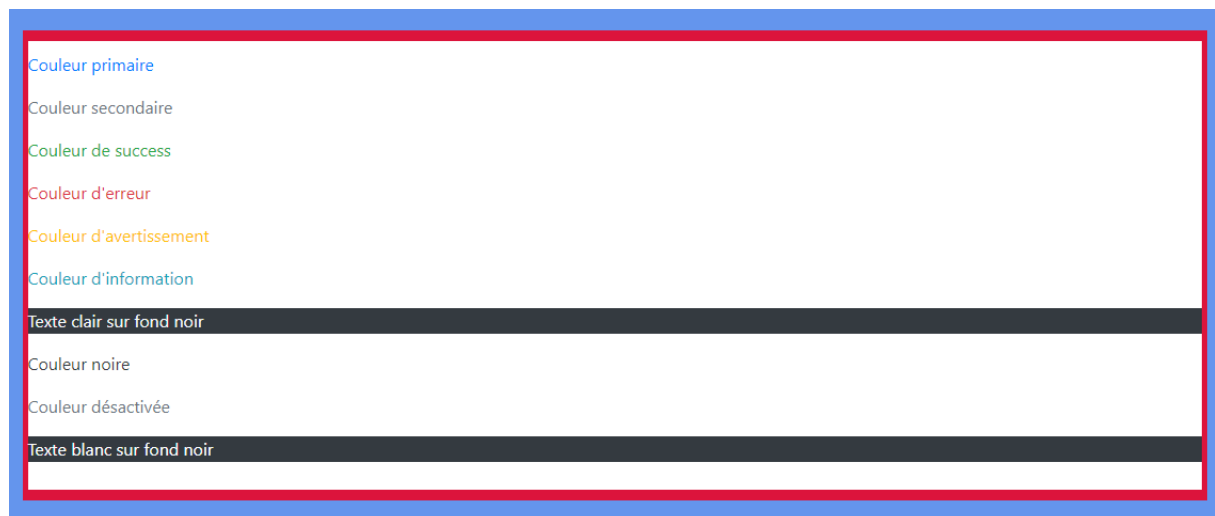
ce texte est magnifique en majuscule

CE TEXTE EST MAGNIFIQUE EN MINUSCULE

Ce Texte Est Magnifique En Capitalisé.

Et enfin, l'application des couleurs est tout aussi facile ! Les couleurs sont prédéfinies par Bootstrap de telle manière à ce que chaque couleur soit associée à une situation ou une action :

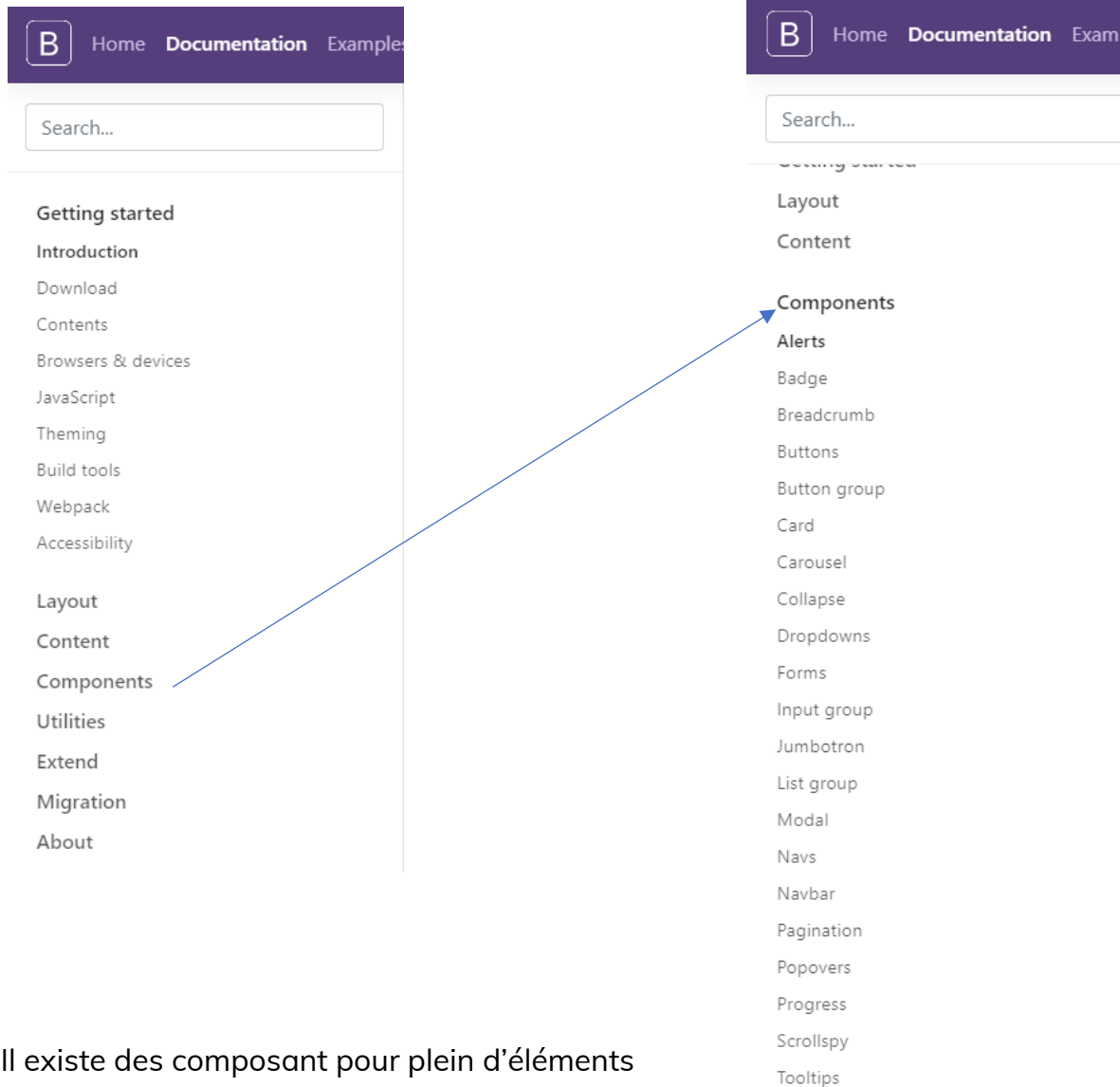
```
<div class="container">
  <div class="row">
    <div class="col-lg-12">
      <p class="text-primary">Couleur primaire</p>
      <p class="text-secondary">Couleur secondaire</p>
      <p class="text-success">Couleur de succès</p>
      <p class="text-danger">Couleur d'erreur</p>
      <p class="text-warning">Couleur d'avertissement</p>
      <p class="text-info">Couleur d'information</p>
      <p class="text-light bg-dark">Texte clair sur fond noir</p>
      <p class="text-dark">Couleur noire</p>
      <p class="text-muted">Couleur désactivée</p>
      <p class="text-white bg-dark">Texte blanc sur fond noir</p>
    </div>
  </div>
</div>
```



## Composants

Le principal avantage de Bootstrap repose sur l'utilisation de ses composants. En effet, avec seulement quelques classes, vous pouvez styliser le contenu de votre page.

Pour accéder aux différents composants, retournez dans la documentation de Bootstrap et dans la section « composants » :

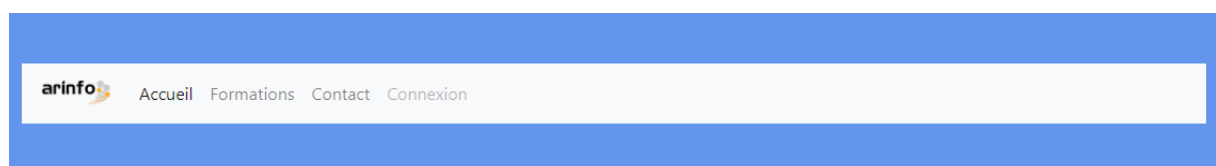


Il existe des composant pour plein d'éléments différents, voici les plus courants :

### Navbar

Il existe plusieurs types de navbar. N'hésitez pas à explorer et tester les différents modèles proposés. Voici un exemple, il vous suffit de copier/coller le morceau de code présenté

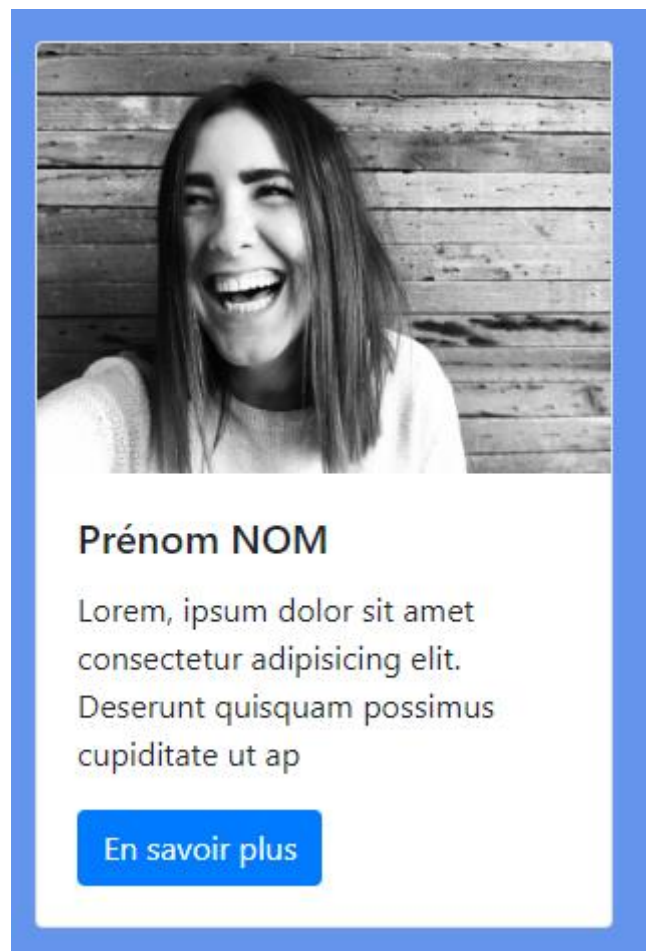
```
<div class="container">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#"></a>
    <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav"
      aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item active">
          <a class="nav-link" href="#">Accueil
            <span class="sr-only">(current)</span>
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Formations</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Contact</a>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" href="#">Connexion</a>
        </li>
      </ul>
    </div>
  </nav>
</div>
```



## Cartes

A travers vos différents projets vous serez amené à présenter des produits, des profils, etc et généralement pour ce faire, vous utiliserez des modèles de cartes (« cards » en anglais) qui permettent d'associer photo, description et bouton en un seul et même endroit :

```
<div class="container">
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title">Prénom NOM</h5>
      <p class="card-text">
        Lorem, ipsum dolor sit amet com
        vur adipisicing elit. Deserunt quisquam
        possimus cupiditate ut ap
      </p>
      <a href="#" class="btn btn-primary">En savoir plus</a>
    </div>
  </div>
</div>
```



## Conclusion & alternatives

---

Comme vous avez pu le constater, en très peu de temps, vous pouvez organiser, structurer, agrémenter et styliser votre page web avec quelques classes spécifiques proposées par Bootstrap.

Ce framework possède de très nombreuses fonctionnalités et subtilités qui rend cet outil facile à utiliser, mais difficile à maîtriser.

Maintenant, que vous êtes initiés, je ne peux que vous inviter à découvrir les classes/composants proposés dans la documentation. Il est inutile de tout retenir, vous irez chercher les composants dont vous avez besoin quand cela sera nécessaire, mais il est important de savoir que tel ou tel élément existe ou que telle ou telle mise en page est possible.

Il existe d'autres alternatives de librairies CSS qui fonctionnent de manière similaire à Bootstrap, et je vous invite à y jeter un œil car le style est différent :

- Bulma : <https://bulma.io/>
- Materialize : <https://materializecss.com/>
- Skeleton : <http://getskeleton.com/>
- Semantic UI : <https://semantic-ui.com/>
- Et bien d'autres...

Pour mettre en application ce que nous venons de voir, vous allez pouvoir suivre le tuto qui suit ce cours, où nous allons créer un site web qui vous permettra d'appliquer les notions que nous avons vu précédemment et bien plus encore. L'objectif sera d'utiliser le moins de CSS possible et le plus de classes Bootstrap dont nous aurons besoin.

Bonne chance !