*Formation Bootstrap*

*Notes de cours*



*Johnny Piette*

Table des matières

[I. A propos de ce cours 2](#_Toc64471611)

[II. Prérequis 2](#_Toc64471612)

[III. Présentation de Bootstrap 2](#_Toc64471613)

[1. Brainstorming : Qu’est-ce que Bootstrap pour vous ? 2](#_Toc64471614)

[2. Brève présentation de Bootstrap 2](#_Toc64471615)

[3. Pourquoi utiliser Bootstrap ? 3](#_Toc64471616)

[4. Bootstrap 4.x et Bootstrap 5 3](#_Toc64471617)

[5. Comment Utiliser Bootstrap ? 3](#_Toc64471618)

[IV. Intégration des Fichiers Bootstrap 4](#_Toc64471619)

[V. Fonctionnement de Bootstrap 4](#_Toc64471620)

[VI. Manipulation du texte 7](#_Toc64471621)

[1. Alignement du texte 7](#_Toc64471622)

[2. Alignement du texte fonction de l’écran 7](#_Toc64471623)

[3. Transformation du texte 8](#_Toc64471624)

[VII. Les images 8](#_Toc64471625)

[1. Responsive 8](#_Toc64471626)

[2. Thumbnail 8](#_Toc64471627)

[3. Alignement 8](#_Toc64471628)

[VIII. Le composant Jumbotron 9](#_Toc64471629)

[IX. Lecteur d’écran 10](#_Toc64471630)

[X. Le Composant Carrousel 10](#_Toc64471631)

# A propos de ce cours

Ce document contient les notes de cours sur Bootstrap. En complément, vous avez un autre document qui contient les instructions pour réaliser les différents exercices/manipulations que nous ferons en classe. A ce document d’exercices viendra se greffer un ensemble de répertoires reprenant les différents exercices/manipulations. Si vous vous en sortez, n’allez pas voir le résultat final de ce nos manipulations/exercices.

De plus, il est complémentaire au site <http://eqla.ddns.net> que je vous invite à visiter.

# Prérequis

Pour aborder Bootstrap les prérequis suivants sont nécessaires :

* HTML
* CSS
* Un peu de JS pour certains composants.

En effet, Bootstrap étant un Framework basé sur la présentation, les notions de class et id sur des balises sont nécessaires.

Petit rappel : Quelle est la différence entre les attributs id et class ?

Tous les deux se trouvent sur des balises HTML.

* id : cet attribut doit être unique. Il peut être utilisé pour appliquer un style très précis à une balise.
* class : n’a pas la prétention d’être unique mais peut l’être. Il est utilisé pour appliquer un même style à différentes balises.

L’utilisation des class dans Bootstrap est omniprésente donc il est important de bien comprendre ce qu’est un attribut.

De plus, il est très courant de savoir qu’un élément peut avoir plusieurs classes. Et dans Bootstrap c’est monnaie courante d’ajouter plusieurs classes à un élément.

# Présentation de Bootstrap

## Brainstorming : Qu’est-ce que Bootstrap pour vous ?

On commence par demander aux stagiaires ce qu’est Bootstrap ? Quels sont ses intérêts ?

## Brève présentation de Bootstrap

Bootstrap est le Framework CSS le plus célèbre au monde. Il fournit une liste d’outils qui permet de simplifier le design de sites internet. L’adresse officielle est <https://getbootstrap.com/>

C’est un ensemble qui contient des boutons, des formulaires, des outils de navigations, une typographie et divers composants.

Il est dit « Responsive » ou « Responsive Web Design » : permet la consultation confortable sur des écrans de différentes tailles. Ça peut allez du desktop, smartphone, tablette, télévision, etc.

En avril 2020, il est le 7ème projet le plus populaire sur GitHub.

## Pourquoi utiliser Bootstrap ?

Il n’est pas obligatoire de l’utiliser mais il peut vous faire gagner pas mal de temps dans certaines situations. Il permet d’avoir une certaine cohérence graphique pour les sites. Il est possible de trouver des thèmes gratuits et payants.

Les avantages d’utiliser Bootstrap sont les suivants :

* Gain de temps en développement.
* Cohérence dans le design de votre site.
* Pensé Mobile First : Android, IOS, Windows 10 mobile.
* Compatible à la majorité des navigateurs : tous les navigateurs. Et pour le cas d’IE à partir de la version 10.
* Responsive.
* Open Source.

## Bootstrap 4.x et Bootstrap 5

En ce moment, février 2021, nous sommes au moment où Bootstrap 5 va bientôt pointer le bout de son nez. Il est toujours en phase béta. Comme la version 4.x est la plus déployée, je pense qu’il est plus intéressant pour vous de voir la version 4.x qui est la plus installée et utilisée.

La version 4.x dépend de la librairie JavaScript jQuery qui doit impérativement être chargée avant le fichier JavaScript de Bootstrap.

Mais le passage à Bootstrap 5 ne devrait pas être trop difficile normalement. De plus Bootstrap 5 va se libérer de sa dépendance à jQuery et fera du pur JavaScript (appelé parfois Vanilla JS).

## Comment Utiliser Bootstrap ?

Bootstrap est un Framework qui est composé d’un ensemble de fichiers. Pour utiliser Bootstrap, il nous faut donc utiliser des fichiers que vous pourrez trouver à cette adresse : <https://getbootstrap.com/docs/4.6/getting-started/download/>

Il y a deux manières d’utiliser ces fichiers :

* Soit vous les téléchargez (1 fichier CSS et 1 ou 2 fichiers JS) sur le site Bootstrap. Et vous ajoutez le lien dans votre HTML. Vous voyez que j’ai mis 1 ou 2 fichiers JS :
  + 2 fichiers JavaScript :
    - L’un pour Popper.js qui permet d’avoir des Tooltips (info-bulles) sur des éléments de votre page. Donnant un bel effet.
    - L’autre pour le JavaScript de Bootstrap.
  + 1 fichier JavaScript : c’est un bundle (un paquet) qui contient Popper et Bootstrap.
* Soit vous utilisez des adresses qui pointent sur ce qu’on appelle des CDN (Content Delivery Network). L’avantage des CDN c’est qu’ils sont super rapides mais si vous voulez les utiliser, vous devez ajouter l’attribut integrity pour vérifier que c’est le code javascript que vous voulez et qu’il n’a pas été remplacé par un hacker. Les navigateurs modernes vérifieront grâce à la valeur mise pour l’attribut integrity qu’il s’agit bien du fichier que vous voulez.

Cependant, avec Bootstrap 4.x et versions antérieures, il est impératif d’ajouter la librairie JavaScript jQuery. Elle doit être chargée avant les fichiers JavaScript Popper et Bootstrap. C’est-à-dire que dans votre page HTML vous mettrez la balise <javascript> de jQuery avant celles de Popper et de Bootstrap. Le cas échéant, Bootstrap ne fonctionnera pas. C’est pourquoi Bootstrap 5 n’utile plus jQuery. Le JavaScript moderne permet de s’en affranchir.

Vous aurez des exemples d’intégration à la page suivante : <https://getbootstrap.com/docs/4.6/getting-started/introduction/>

Pour vous simplifier la tâche, j’ai créé dans le répertoire **Eqla-Bootstrap/Pratique/Exercices/** un fichier html modèle pour commencer vos exercices. Il se nomme **Template.html** et les ressources nécessaires pointent sur des CDN. Maintenant, libre à vous de l’utiliser ou non.

Dans ce fichier, j’ai fait pointer vers la dernière version 4.6 de Bootstrap. De plus, j’ai ajouté un fichier CSS supplémentaire (Icones Bootstrap) que nous discuterons plus tard mais comme ça nous avons notre page web modèle Bootstrap déjà prête pour cela.

# Intégration des Fichiers Bootstrap

Allez dans le répertoire ***Eqla-Bootstrap/Pratique/Note-de-Cours/Integration-de-Bootstrap***

Vous verrez dans le fichier exemple1.html comment intégrer les CDN pour utiliser Bootstrap.

Ensuite dans le fichier exemple2.html, vous verrez comment héberger vos fichiers au lieu d’utiliser les CDN. C’est assez trivial mais comme ça vous aurez un exemple de chaque.

Vous pouvez tester et constater que le résultat est le même avec et sans CDN.

# Fonctionnement de Bootstrap

Bootstrap fonctionne principalement avec l’utilisation de classes. Il faut savoir que le fichier CSS de Bootstrap quand il n’est pas minifié (ramené sur une ligne pour qu’il prenne moins de place) fait 10600 lignes … Il n’est pas nécessaire de connaître par cœur toutes les classes. Personnellement j’utilise le site principal et Google.

La première classe que l’on va utiliser est la classe .**container** que l’on applique à un div. Elle permettra d’adapter la largeur du div en fonction de la résolution de l’écran du périphérique utilisé.

Exemple :

|  |
| --- |
| <body>  <div class= "**container**">  <h1>Hello, World !</h1>  <p>  <span class="font-weight-bold">Le Lorem Ipsum</span> est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un imprimeur anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.  </p>  </div>  </body> |

Si l’on regarde le CSS de Bootstrap, on voit que dès que l’on utilise la classe .container, que tout va s’adapter.

Regardons rapidement ce CSS pour la classe .container On voit que tout changera automatiquement en fonction du média utilisé. Vous pouvez en même temps voir qu’il y a d’autres classes **.container**, .**container-fluid**, etc

|  |
| --- |
| .container,  .container-fluid,  .container-lg,  .container-md,  .container-sm,  .container-xl {  width: 100%;  padding-right: 15px;  padding-left: 15px;  margin-right: auto;  margin-left: auto  }  @media (min-width:576px) {  .container,  .container-sm {  max-width: 540px  }  }  @media (min-width:768px) {  .container,  .container-md,  .container-sm {  max-width: 720px  }  }  @media (min-width:992px) {  .container,  .container-lg,  .container-md,  .container-sm {  max-width: 960px  }  }  @media (min-width:1200px) {  .container,  .container-lg,  .container-md,  .container-sm,  .container-xl {  max-width: 1140px  }  } |

**.container** agit différemment en fonction des résolutions des périphériques (d’après l’analyse du morceau de CSS mis plus haut) :

Cas n°1 : résolution inférieure à 576 pixels => 100% de l’écran

Cas n°2 : résolution supérieure ou égale à 576 pixels et inférieure à 768 pixels, l’élément s’affichera au centre de l’écran et sa largeur sera de 540 pixels.

Cas n°3 : résolution supérieure ou égale à 768 pixels et inférieure à 992 pixels, l’élément s’affichera au centre de l’écran et sa largeur sera de 720 pixels.

Cas n°4 : résolution supérieure ou égale à 992 pixels et inférieure à 1200 pixels, l’élément s’affichera au centre de l’écran et sa largeur sera de 960 pixels.

Cas n°5 : résolution ou égale à 1200 pixels, l’élément s’affichera au centre de l’écran et sa largeur sera de 1140 pixels.

.**container-fluid** permet d’utiliser 100% de la taille de votre écran et n’est pas fixée comme l’est .**container**.

Cependant, avec la classe .**container**, comme nous l’avons vu plus haut au cas n°1, si l’écran est inférieur à 576 pixels alors 100% sera utilisé comme **.container-fluid**.

Reprenons l’exemple précédent avec .**container-fluid**

|  |
| --- |
| <body>  <div class= "**container-fluid**">  <h1>Hello, World !</h1>  <p>  <span class="font-weight-bold">Le Lorem Ipsum</span> est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un imprimeur anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.  </p>  </div>  </body> |

Si vous en avez la possibilité, comparez le résultat de la page sur un smartphone et un ordinateur de bureau. Ou encore, jouez sur la largeur du navigateur sur votre desktop, la largeur s’auto adapte.

# Manipulation du texte

## Alignement du texte

**a. La classe .text-justify**

Elle permet de justifier le texte.

**b. La classe .text-center**

Elle permet de centrer le texte.

**c. La classe .text-left**

Elle permet d’aligner à gauche votre texte.

**d. La classe .text-right**

Elle permet d’aligner à droite votre texte.

## Alignement du texte fonction de l’écran

Des abréviations peuvent s’ajouter à certaines classes pour conditionner l’action en fonction de l’écran.

* sm (small) : résolution supérieure ou égale à 576 pixels et inférieure à 768 pixels.
* md (medium): résolution supérieure ou égale à 768 pixels et inférieure à 992 pixels.
* lg (large): résolution supérieure ou égale à 992 pixels et inférieure à 1200 pixels
* xl (extra large): résolution ou égale à 1200 pixels.

Dans la littérature Bootstrap on voit souvent l’utilisation du caractère \* pour certaines classes. Ça veut dire qu’il le faut remplacer par une valeur numérique ou du texte.

Exemples :

.text-\*-right : .text-xl-right

.text-\*-center : .text-md-center

.text-\*-left : .text-sm-left

D’après la feuille de style de Bootstrap, je n’ai pas trouvé ces abréviations pour la classe text-justify.

## Transformation du texte

La classe **.text-lowercase**: texte converti en minuscules.

La classe **.text-uppercase**: TEXTE CONVERTI EN MAJUSCULES.

La classe **.text-capitalize**: Première Lettre De Chaque Mot Est En Majuscule.

Exemples :

|  |
| --- |
| <p>.text-lowercase: <span class="text-lowercase">TEXTE CONVERTI EN MINUSCULES </span>.</p>  <p>.text-uppercase: <span class="text-uppercase">texte converti en majuscules</span>.</p>  <p>.text-capitalize: <span class="text-capitalize">première lettre de chaque mot est en majuscule.</span></p> |

# Les images

Nous allons voir que Bootstrap permet de facilement rendre une image responsive, en faire une jolie vignette, aligner celle-ci.

Evidemment toutes ces classes peuvent être combinées entre elles comme dans le précédent exemple.

## Responsive

Bootstrap permet de rapidement permettre à une image d’être responsive. Ajouter la classe .img-fluid et votre image va s’auto-adapter en fonction de l’écran.

|  |
| --- |
| <img src="Images/logo\_eqla.png" class="**img-fluid**" alt="logo d'Eqla" width="10000px"  /> |

Ici on a ajouté l’attribut width="10000px". L’image ne fera bien sûr jamais 10000 pixels. Bootstrap veille au grain. 😊

## Thumbnail

La classe .img-thumbnail ajoute à l’image un bord blanc arrondi.

|  |
| --- |
| <img src="/Images/paris.jpg" class="**img-thumbnail**" alt="Image de paris" /> |

## Alignement

Permet de mettre des images à gauche **.float-left** ou à droite **.float-right** quelque soit la taille de l’écran.

|  |
| --- |
| <img src="/Images/logo\_eqla.png" class="**img-fluid float-left**" alt="logo d'Eqla" />  <img src="/Images/paris.jpg" class="**img-fluid img-thumbnail float-right**" alt="Image de paris" /> |

On peut aussi définir ces alignements en fonction du périphérique : **.float-\*-left** ou **.float-\*-right** où **\*** peut avoir différentes valeurs.

* sm : small
* md : medium
* lg : large
* xl : extra-large

# Le composant Jumbotron

Le Jumbotron est une boite rectangulaire qui va permettre de mettre en avant un certain message. Il est très grand et donne directement un impact visuel. On ne peut pas passer à côté quand on arrive sur une page.

Pour définir un jumbotron, on va ajouter une classe .jumbotron à un élément conteneur de type div. On va ensuite pouvoir placer plus ou moins n’importe quel contenu HTML à l’intérieur.

Exemple :

|  |
| --- |
| <div class="**jumbotron**">  <h1> Hello, world!</h1>  <p class="lead">Ceci est un exemple du composant jumbotron pour capter l’attention pour son contenu, une information importante.</p>  </div> |

Maintenant, vous n’êtes pas obligés de mettre le jumbotron dans un div avec une class .container mais vous pouvez avoir un div container après le jumbotron. Donnant un effet que le jumbotron prend toute la largeur de la page et que le div ayant la classe .container soit lui centré. Ça dépend de ce que l’on veut mais ce n’est pas mal non plus.

Et pour terminer vous pourriez aussi avoir un div ayant la classe container à l’intérieur de votre jumbotron. Ça n’est pas interdit.

|  |
| --- |
| <div class="**jumbotron** text-white" style="background-color: #563d7c">  <div class="**container**">  <h1> VII. Le composant Jumbotron</h1>  <p class="lead">Ceci est un exemple du composant jumbotron pour capter l’attention par son contenu, une information importante.</p>  </div>  </div> |

Et pour finir, vous avez la class .jumbotron-fluid qui supprime les coins arrondis et les rends carrés. A vous de décider entre « arrondis » ou « carrés ».

# Lecteur d’écran

J’ai découvert cette classe par hasard en me « promenant » sur le site Bootstrap. Il m’a semblé évident de vous la présenter.

La classe .sr-only, elle n’est accessible que par un lecteur d’écran.

Exemple :

|  |
| --- |
| <div class = "container">  <p> Votre formateur s’appelle Johnny </p>  <a class = "sr-only" href="#">Votre formateur doit absolument aller chez le coiffeur mais ça restera entre nous.</a>  <p>Votre formateur est très bien coiffé.</p>  </div> |

La classe .sr-only combinée avec .sr-only-focusable n’est visible que si on utilise le clavier (TAB) pour passer d’un lien à un autre.

Exemple :

|  |
| --- |
| <div class = "container">  <p> Votre formateur s’appelle Johnny </p>  <a class = "sr-only" href="#">Votre formateur doit absolument aller chez le coiffeur mais ça restera entre nous.</a>  <a class = "sr-only sr-only-focusable" href ="#">Votre formateur est très bien coiffé.</a>  </div> |

# Le Composant Carrousel

Bootstrap fournit nativement un composant visuel de diaporama très sympathique que l’on appelle le carrousel. Il est fréquent de le voir sur des pages webs. Il présente par exemple des promos qui défilent via des images, du texte avec un changement cyclique définissable.

On peut y mettre des images, du texte, etc.

Sa construction suit le schéma suivant :

1. Un div contenant notre carrousel ayant les classes **.caroussel** et **.slide**

Ce div doit avoir les attributs suivants data-ride="caroussel"

1. Un second div inclus dans le premier avec la classe **.caroussel-inner**
2. Ensuite nous ajouts des div pour le contenu (slide) à faire défiler. Chaque contenu est inclus dans un div ayant la classe **.caroussel-item**. Notons qu’un des divs de slide doit impérativement avoir la classe **.active**. Sinon le caroussel ne sera pas visible.

Exemple :

|  |
| --- |
| <div id="carouselExampleSlidesOnly" class="**carousel slide**" data-ride="carousel" data-interval="false" data-keyboard="true">  <div class="**carousel-inner**">  <div class="**carousel-item active**">  <img src="Images/la.jpg" alt="Los Angeles, arrivée d'un groupe sur scène."> Los Angeles, pour <a href="">plus de photos</a>.  </div>  <div class="**carousel-item**">  <img src="Images/chicago.jpg" alt="chicago, image d'un chanteur sur scène avec une guitare.">  </div>  <div class="**carousel-item**">  <img src="Images/ny.jpg" alt="New-York, image dans la foule en direction de la scène.">  </div>  </div>  </div> |

Par défaut il n’a pas de boutons de défilement mais on peut les ajouter.