*Formation Bootstrap*

*Notes de cours*



*Johnny Piette*

# 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.

# 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.

# 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,

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 d’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 affichera au centre de l’écran et sa largeur sera de 576 pixels.

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

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

Cas n°5 : résolution ou égale à 1200 pixels, l’élément affichera au centre de l’écran et sa largeur sera de 1200 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. Ca 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 particules pour justifier le texte.

## Transformation du texte

Redimensionnant