BOOTSTRAP

Bootstrap, **kit CSS** créé par les développeurs de Twitter.

C’est un framework CSS côté client, c’est à dire un ensemble d’outils et de [composants logiciels](https://fr.wikipedia.org/wiki/Composant_logiciel) structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d’une partie d'un [logiciel](https://fr.wikipedia.org/wiki/Logiciel).

Bootstrap est un framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript.

Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web.

**Les intérêts:**

Gain de temps

Présentation homogène

Système de grille pour faciliter le positionnement des éléments.

Responsive.

**Un framework ?**

Il s'agit d'un ensemble de composants structurés qui sert à créer les bases et à organiser le code informatique pour faciliter le travail des programmeurs

Il existe des frameworks côté serveur (désignés backend en anglais), et d'autres côté client (désignés frontend en anglais). Bootstrap fait partie de cette deuxième catégorie.

**Installation et Utilisation**

Aller sur le site de Bootstrap et le télécharger :

<https://getbootstrap.com/docs/3.3/getting-started/>

Pour que Bootstrap fonctionne il faut le déclarer dans les pages HTML, qui doivent être impérativement au format HTML 5, il faut donc prévoir le bon DOCTYPE :

<head>

…

<link href="chemin du dossier bootstrap/css/bootstrap.min.css" rel="stylesheet">

</head>

Ou

<head>

…

<link href="chemin du dossier bootstrap/css/bootstrap.css" rel="stylesheet">

</head>

Si vous utilisez des composants JavaScript, vous devez également référencer la librairie de Bootstrap ainsi que jQuery (la librairie jQuery ne fait pas partie des fichiers téléchargés avec Bootstrap et doit être récupérée indépendamment [sur http://jquery.com/](https://jquery.com/)) puis le télécharger dans le dossier js de bootstrap :

<script src="bootstrap/js/jquery.js"></script>

<script src="bootstrap/js/bootstrap.min.js"></script>

**Le template de base**

<meta http-equiv="X-UA-Compatible" content="IE=edge">

Cette déclaration ne concerne que le navigateur Internet Explorer. Elle permet de s'assurer qu'il utilise la dernière version du moteur de rendu. Notez que cette ligne ne passe pas la validation W3C.

<meta name="viewport" content="width=device-width, initial-scale=1">

Cette ligne concerne uniquement les mobiles. On demande que l'affichage occupe tout l'espace disponible avec une taille de 1, autrement dit sans zoom

On trouve ensuite la déclaration du fichier CSS dans sa version minifiée :

<link href="css/bootstrap.min.css" rel="stylesheet">

En fin de page (pour ne pas ralentir le chargement) se trouvent les appels JavaScript

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

**Les CDN**

CDN est l'acronyme de « Content delivery network » ; c'est un réseau de serveurs qui met à disposition des librairies. Il devient ainsi inutile de prévoir ces librairies sur son propre serveur, il suffit de « pointer » vers eux.

Mais si vous modifiez le fichier CSS de Bootstrap pour l'adapter à vos besoins, vous ne pourrez plus bénéficier des avantages d'un CDN.

On peut aussi mettre les appels Javascript dans le head, ainsi voici la structure de base avec des CDN :

<!DOCTYPE html>

<html lang="fr">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

</head>

<body>

</body>

</html>

**Les Media Queries**

Les Media Queries sont destinées à simplifier la création de pages web pour les rendre consultables sur des supports variés (tablettes, smartphones…).

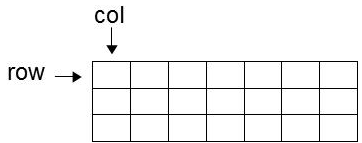
**Une grille**

**Le principe d'une grille**

Une grille est tout simplement un découpage en cellules.

On peut alors décider d'organiser du contenu en utilisant pour chaque élément une ou plusieurs cellules :

La grille de Bootstrap comporte 12 colonnes

Une grille est découpée en rangées (appelées row, parce que tout est en anglais) et colonnes (col)

**Organisation de la grille**

Bootstrap est essentiellement un fichier CSS. Il comporte de nombreuses classes que l'on peut utiliser directement dans les balises HTML.

La première classe à connaître estrow, qui représente une rangée. Cette classe établit des marges négatives à droite et à gauche :

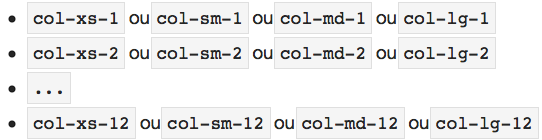
.row {

margin-right: -15px;

margin-left: -15px;

}

Il faut ensuite définir le nombre de colonnes pour chaque élément en sachant qu'il y en a au maximum 12. Pour définir le nombre de colonnes utilisées pour chaque élément, on dispose de quatre batteries de 12 classes :



Bootstrap considère 4 sortes de médias : les petits, genre smartphones (moins de 768 pixels), les moyens, genre tablettes (moins de 992 pixels), les écrans moyens (moins de 1200 pixels) et enfin les grands écrans (plus de 1200 pixels).

Le nom des classes est intuitif :xs pour x-small, sm pour small, md pour medium et lg pour large.

Pour avoir un élément de 4 colonnes de large accouplé avec un élément de 8 colonnes de large sur la même ligne sur un smartphone, on doit utiliser xs:

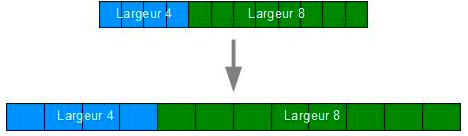
<div class="row">

<div class="col-xs-4">Largeur 4</div>

<div class="col-xs-8">Largeur 8</div>

</div>

La figure suivante illustre ce code avec l'effet quand on passe sur un écran plus grand.



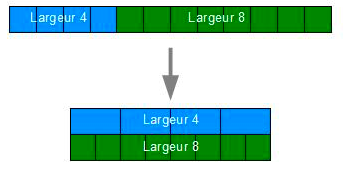
Sur une tablette :

<div class="row">

<div class="col-sm-4">Largeur 4</div>

<div class="col-sm-8">Largeur 8</div>

</div>



**La grille en pratique**

**Le conteneur**

La grille de Bootstrap doit être placée dans un conteneur. Bootstrap propose les classes container et container-fluid.

La classe container contient et centre la grille sur une largeur fixe, qui s'adapte en fonction de la largeur de l'écran. La classe container-fluid permet à la grille d'occuper toute la largeur.

Dans ce premier exemple, on va déclarer une seule rangée avec deux éléments qui occupent tout l'espace :

<div class="container">

<div class="row">

<div class="col-lg-4">4 colonnes</div>

<div class="col-lg-8">8 colonnes</div>

</div>

</div>

Ce qui nous donne la figure suivante.

On a mis un CSS affichant en vert afin de mieux illustrer.

**Plusieurs rangées**

Pour obtenir plusieurs rangées, il suffit d'utiliser plusieurs fois la classerow. Un petit exemple avec 3 rangées pour voir l'effet :

<div class="container">

<div class="row">

<div class="col-lg-1">1 col</div>

<div class="col-lg-2">2 colonnes</div>

<div class="col-lg-3">3 colonnes</div>

<div class="col-lg-6">6 colonnes</div>

</div>

<div class="row">

<div class="col-lg-12">12 colonnes</div>

</div>

<div class="row">

<div class="col-lg-4">4 colonnes</div>

<div class="col-lg-8">8 colonnes</div>

</div>

</div>



**Sauter des colonnes**

Bootstrap permet aussi de sauter des colonnes. Il y a quelques classes d'offset pour ça :

* col-\*-offset-1
* col-\*-offset-2
* ...

Ces classes se contentent d'ajouter une marge gauche. Par exemple :

.col-lg-offset-6 {

margin-left: 50%;

}

Dans ce cas, 6 colonnes représentent la moitié de la fenêtre donc 50%. Voici un exemple :

<div class="container">

<div class="row">

<div class="col-lg-3">3 colonnes</div>

<div class="col-lg-6">6 colonnes</div>

<div class="col-lg-3">3 colonnes</div>

</div>

<div class="row">

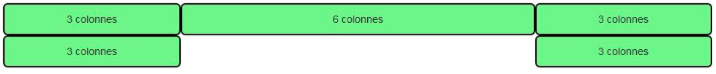
<div class="col-lg-3">3 colonnes</div>

<div class="col-lg-offset-6 col-lg-3">3 colonnes</div>

</div>

</div>

Voici le résultat à la figure suivante:



On peut évidemment multiplier les sauts :

<div class="container">

<div class="row">

<div class="col-lg-2 col-lg-offset-1">2 colonnes</div>

<div class="col-lg-4 col-lg-offset-2">4 colonnes</div>

<div class="col-lg-2 col-lg-offset-1">2 colonnes</div>

</div>

</div>



**Imbrication d’éléments**

On doit souvent imbriquer des éléments dans une page web, autrement dit inclure un row dans un un autre row.

<div class="container">

<div class="row">

<div class="col-lg-8">8 colonnes

<div class="row">

<div class="col-lg-3">3 colonnes</div>

<div class="col-lg-6">6 colonnes</div>

<div class="col-lg-3">3 colonnes</div>

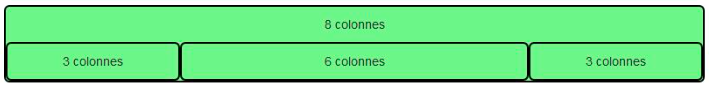
</div>

</div>

</div>

</div>

Voici le résultat à la figure suivante.



Petit TP à faire :

Reproduire la figure suivante :



Correction:

<!DOCTYPE html>

<html>

<head>

<link href="assets/css/bootstrap.css" rel="stylesheet">

<link href="assets/css/tuto.css" rel="stylesheet">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-lg-6">6 colonnes

<div class="row">

<div class="col-lg-4">4 colonnes</div>

<div class="col-lg-offset-4 col-lg-4">4 colonnes</div>

</div>

</div>

<div class="col-lg-6">6 colonnes

<div class="row">

<div class="col-lg-4">4 colonnes</div>

<div class="col-lg-8">8 colonnes</div>

</div>

</div>

</div>

</div>

</body>

</html>

**Mise en page**

L'intérêt principal d'une grille est de réaliser une mise en page.



Code:

<body>

<div class="container">

<header class="row">

<div class="col-lg-12">

Entete

</div>

</header>

<div class="row">

<nav class="col-lg-2">

Menu

</nav>

<section class="col-lg-10">

Section

</section>

</div>

<footer class="row">

Pied de page

</footer>

</div>

</body>

**Combiner les formats**

**Mettre une image**

<div class="col-lg-2"><img src="images/t1.jpg" alt="Tigre"></div>

<div class="col-lg-2"><img src="images/t2.jpg" alt="Tigre"></div>

<div class="col-lg-2"><img src="images/t3.jpg" alt="Tigre"></div>

<div class="col-lg-2"><img src="images/t4.jpg" alt="Tigre"></div>

Si je veux mettre 12 images avec le format lg-2 j’en aurais 6 par lignes mais lorsque je vais rétrécir l’écran elles vont se placer l’une au dessus de l’autre mais l’image sera floue.

L'idéal serait d'avoir :

* Sur moyen et grand écran : 6 images sur la largeur.
* Sur tablette : 4 images sur la largeur.
* Sur smartphone : 3 images sur la largeur.

Comment réaliser cela ? Tout simplement en combinant les classescol-\* :

<div class="col-xs-4 col-sm-3 col-md-2"><img src="images/t1.jpg" alt="Tigre"></div>

<div class="col-xs-4 col-sm-3 col-md-2"><img src="images/t2.jpg" alt="Tigre"></div>

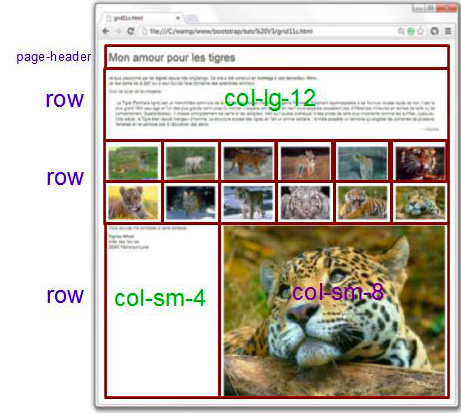
<div class="col-xs-4 col-sm-3 col-md-2"><img src="images/t3.jpg" alt="Tigre"></div>

<div class="col-xs-4 col-sm-3 col-md-2"><img src="images/t4.jpg" alt=« Tigre"></div>

Etc…

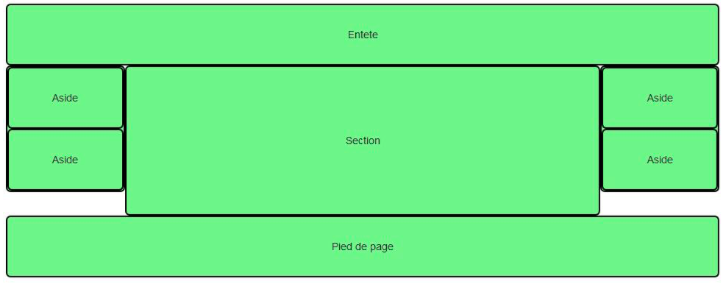
On aura donc :

* Sur un écran de plus de 992 pixels de large : classescol-md-2 actives.
* Sur un écran entre 768 et 992 pixels de large : classescol-sm-3 actives.
* Sur un écran de moins de 768 pixels de large : classescol-xs-4 actives.

Un exemple de mise en page complète :

TP 2

Mise en page à obtenir:

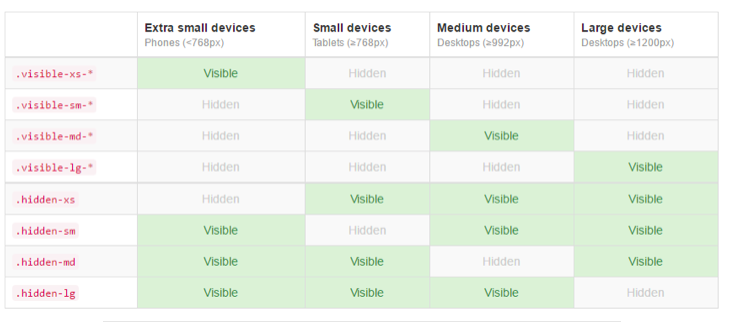


La classe clearfix

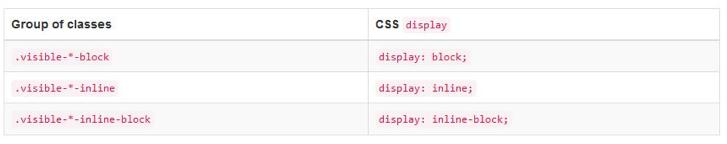
Elle réinitialise les éléments flottants qui suivent pour les ramener dans le flux normal, et éviter qu'ils viennent recouvrir les éléments précédents

**Les classes « responsives »**

Elles vont nous permettre non plus de positionner différemment les éléments, mais carrément de les faire disparaître ou apparaître .



Les classes de visibilité se déclinent en trois sortes selon le type de visualisation :



Correction du TP2

<!DOCTYPE html>

<html>

<head>

<link href="assets/css/bootstrap.css" rel="stylesheet">

<link href="assets/css/tuto.css" rel="stylesheet">

<!-- Un peu de style pour la visualisation -->

<style type="text/css">

.col-lg-8 { line-height: 200px; }

.col-lg-12 { line-height: 80px; }

</style>

</head>

<body>

<div class="container">

<header class="row">

<div class="col-lg-12">

Entete

</div>

</header>

<div class="row">

<div class="col-lg-2">

<div class="row">

<aside class="col-lg-12">

Aside

</aside>

<aside class="col-lg-12">

Aside

</aside>

</div>

</div>

<section class="col-lg-8">

Section

</section>

<div class="col-lg-2">

<div class="row">

<aside class="col-lg-12">

Aside

</aside>

<aside class="col-lg-12">

Aside

</aside>

</div>

</div>

</div>

<footer class="row">

<div class="col-lg-12">

Pied de page

</div>

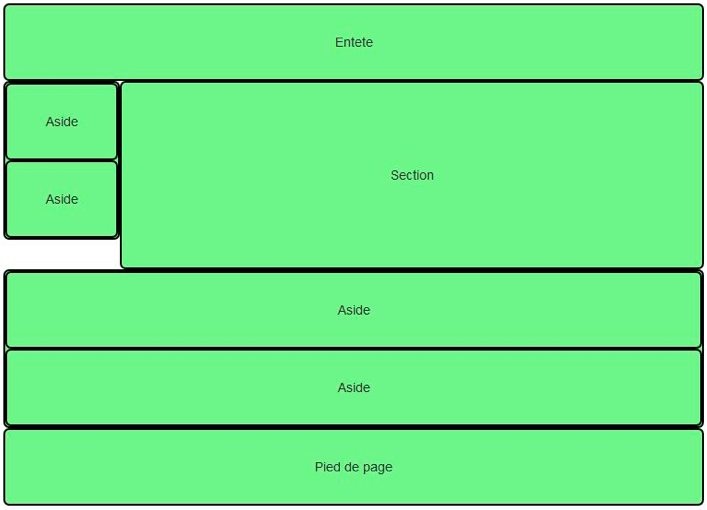
</footer>

</div>

</body>

</html>

Comme on a mis des classes lg, si on passe en mode tablette tout va s’empiler l’un au dessus de l’autre.

Comment faire pour garder cet aspect ci dessous même sur tablette :

Remplacer les classescol-lg-\* par des classescol-md-\* pour avoir un comportement de base pour écrans moyens et grands. J'ai ensuite introduit des classescol-sm-\* pour avoir le comportement désiré sur tablette.

**Listes, descriptions et tableaux**

**Liste sans style**

Il existe la classe list-unstyled, à appliquer à la balise<ul>, ce qui donne donc <ul class="list-unstyled">, qui supprime les puces.

**Les tableaux**

**Classe « table »**

C'est la classe de base pour les tableaux, elle fixe quelques styles et crée des lignes de séparation horizontales :

<table class="table">

**Classe « table-bordered »**

C'est la classe pour avoir des bordures :

<table class="table table-bordered">

**Classe « table-striped »**

Cette classe ajoute un fond avec un gris très léger pour les lignes impaires (1, 3, …) :

**Classe « table-condensed »**

Cette classe permet de condenser le tableau pour qu'il occupe moins de place verticalement :

**Un peu de couleur dans les tableaux**

Vous avez la possibilité de colorer le fond des lignes des tableaux avec les classes info, success, danger, warning et active.

<tr class="success">

**Formulaires**

Un formulaire classique :

Le même formulaire avec la classe form-control :

<form class="col-lg-6">

<legend>Légende</legend>

Text : <input type="text" class="form-control">

Textarea : <textarea id="textarea" class="form-control"></textarea>

Select :

<select class="form-control">

<option>Option 1</option>

<option>Option 2</option>

<option>Option 3</option>

</select>

<button>Envoyer</button>

</form>



**Formulaire en ligne**

Il arrive qu'on ait besoin d'avoir un formulaire disposé sur une ligne, par exemple dans une barre de navigation.

cc

Sjjj