

BOOTSTRAP

SOMMAIRE

Sommaire	1
Bootstrap	4
Bootstrap à toute allure	4
Késako	4
Téléchargement	4
Installation	4
Utilisation.....	4
Des classes css pour styler son site	5
Une grille la mise en page (qu'on place dans un contenair)	5
Des templates complets pour la mise en page	5
Tuto complet Bootstrap : à consulter sans modération !!!.....	5
Introduction	6
Prérequis	6
Les framework CSS	6
Utilisation	6
Avantages	6
Cross-browser : Normalize.css	6
Cross-device : design responsive	6
Rapidité.....	6
Inconvénients	6
Adapter Bootstrap : LESS.....	6
Ressources	7
Ressources officielles	7
Tutoriel.....	7
Outil de test, d'apprentissage et/ou de développement.....	7
Utilisation par CDN	8
CDN : késako ?	8
CDN du fichier css : bootstrapcdn.....	8
CDN des fichier js : google et bootstrapcdn.....	8
Exemple	8
Installation sur sa machine : tuto	9
Utilité : alternative au CDN	9
Download Bootstrap.....	9
Minimum.....	9
Avancé.....	9
Contenu du téléchargement.....	9
Minimum.....	9
Avec sources	9
Contenu du dossier dist.....	9
Sous dossier css.....	9
Sous dossier fonts	10
Sous dossier js.....	10
Download de plugin jQuery	10
Installation du répertoire bootstrap à la racine	10
Environnement de production.....	10
Environnement de développement.....	10
Regarder le code des classes bootstrap	10

Charger la bibliothèque bootstrap dans une page HTML.....	11
Chargement autonome.....	11
Head et javascript d'un fichier HTML bootstrap.....	12
<head>.....	12
Principes.....	12
Exemple : 00_chargerBS.html.....	12
<script>.....	12
La grille	14
Eléments.....	14
Présentation.....	14
Ligne.....	14
Colonnes.....	14
Container et container-fluid (responsive).....	14
Exemples.....	14
Comportement de base.....	15
Exemple : 01_1_grille.html.....	15
Exemple : 02_2_grille_float.html.....	16
Comportements responsive.....	17
Principes.....	17
Exemple : 02_grille_empilement.html.....	17
Mise en page avec la grille.....	19
Lignes et colonnes variables.....	19
Sauter des colonnes : offset.....	20
Imbrication d'éléments.....	21
Décaler des colonnes.....	22
Mise en page « classique »	23
Mise en page classique.....	23
Principes.....	23
Test.....	23
Code.....	23
Version avec plusieurs sections.....	25
Test.....	25
Mise en page avec des asides cachés.....	26
Objectifs.....	26
Tests.....	26
Solutions.....	27
Mise en page à photos	29
Problème de résolution.....	29
Principe.....	29
Exemple flou !.....	29
Solution.....	29
Autres exemples.....	30
Pagination de photos avec JavaScript.....	30
Listes en tout genres.....	30
Exemple de page avancée.....	30
Composants de base Bootstrap	31
Mise en valeur du texte.....	31
Liste.....	31
Table.....	31
Formulaire.....	31
Image.....	31
Bouton.....	31
Icônes.....	31
etc. !.....	31
Composants plus complexes Bootstrap	32

Collapse	32
Barre de navigation	32
Pagination.....	32
Liste et panel	32
Media.....	32
Navigation.....	32
Carrousel.....	32
etc. !.....	32
Templates Bootstrap	33

Edition : novembre 2016

BOOTSTRAP

<https://openclassrooms.com/courses/prenez-en-main-bootstrap>

<http://www.w3schools.com/bootstrap/default.asp>

<http://getbootstrap.com/getting-started/>

Bootstrap à toute allure

Késako

- Framework CSS, stylé et responsive.
- Créé par les développeurs de Twitter
- Basé sur HTML 5 et CSS3.
- Embarque des plugins [JQuery](#) de qualité
- Comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web
- Apporte du style pour les boutons, les formulaires, la navigation...
- Propose des templates de mises en page
- Permet de concevoir un site web rapidement et avec peu de lignes de code ajoutées.

Téléchargement

Télécharger Bootstrap : <http://getbootstrap.com/getting-started/#download>

Les plugins JQuery ne sont pas installés en standard dans bootstrap. Ils se récupèrent ici : <http://jquery.com>. Installer les plugins dans le dossier js.

Installation

Installer le dossier bootstrap avec ses 3 dossiers css, fonts et js à la racine du site.

Utilisation

Pour utiliser bootstrap dans un fichier HTML :

➤ *Dans le head, on linke un fichier .css*

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

➤ *Dans le body, on linke 2 fichiers .js (bootstrap utilise jquery)*

```
<script src="bootstrap/js/jquery.js"></script>
<script src="bootstrap/js/bootstrap.min.js"></script>
</body>
```

Des classes css pour styler son site

- *Simple reprise des balises HTML : <h1>*
http://www.w3schools.com/bootstrap/bootstrap_typography.asp
- *Boutons améliorés : <button type="button" class="btn btn-primary">*
http://www.w3schools.com/bootstrap/bootstrap_buttons.asp
- *Système de pagination : <ul class="pagination">*
http://www.w3schools.com/bootstrap/bootstrap_pagination.asp
- *Système de « collapse : <div id="demo" class="collapse">*
http://www.w3schools.com/bootstrap/bootstrap_collapse.asp
- *etc.*

Une grille la mise en page (qu'on place dans un contenair)

http://www.w3schools.com/bootstrap/bootstrap_grid_basic.asp

```
<div class="container">  
  <div class="row">  
    <div class="col-sm-4">  
      etc.
```

- *Tuto complet de la grille :*
http://www.w3schools.com/bootstrap/bootstrap_grid_system.asp

Des templates complets pour la mise en page

http://www.w3schools.com/bootstrap/bootstrap_templates.asp
<http://getbootstrap.com/getting-started/#examples-framework>

Tuto complet Bootstrap : à consulter sans modération !!!

<http://www.w3schools.com/bootstrap/default.asp>

Introduction

Prérequis

HTML-CSS

Les framework CSS

Il en existe beaucoup : Elasticss, Knacss, Blueprint, unsemantic, YUI, 52Framework, BlueTrip, ez-css, Pure, Gumbo, etc.

[Foundation](#) : très avancé

Bootstrap : très avancé et très populaire

Utilisation

- **Simple** : s'utilise en linkant un fichier contenant les classes définies par bootstrap.
- Ensuite on peut ajouter ces classes aux balises HTML pour obtenir des résultats stylés et responsives.
- Bootstrap propose une **bonne documentation** : <http://getbootstrap.com/css/>
- Il y a **beaucoup de ressources** sur le web : <http://www.w3schools.com/bootstrap/default.asp>
- Bootstrap garantit une **évolution permanente**.

Avantages

Cross-browser : Normalize.css

Abstraction des navigateurs qui réagiront tous pareil grâce au [Normalize.css](#) : petit fichier CSS qui établit des règles pour avoir un rendu identique quel que soit le navigateur utilisé.

Cross-device : design responsive

Abstraction des écrans (responsive design)

Rapidité

Permet de concevoir un site web stylé rapidement : structures déjà présentes, style homogène, grille de positionnement (12 colonnes chez Bootstrap), éléments stylés (bouton, barre de navigation, ...), etc.

Inconvénients

Normalisation du graphisme : tous les sites se ressemblent !

Mais Bootstrap est configurable avec LESS.

Adapter Bootstrap : LESS

Pour adapter Bootstrap, on utilise LESS qui étend le CSS avec un comportement dynamique, utilisant des variables, des classes abstraites, des opérations et des fonctions. LESS fonctionne aussi bien côté client (les navigateurs) que côté serveur, avec Node.js.

<https://openclassrooms.com/courses/simplifiez-vous-la-vie-avec-less>

<http://lesscss.org>

Ressources

Ressources officielles

Site : <http://getbootstrap.com>

Blog : <http://blog.getbootstrap.com/>

Github : <https://github.com/twbs/bootstrap>

Tutoriel

<http://www.w3schools.com/bootstrap/>

<http://getbootstrap.com/css/>

<http://www.webdesignerdepot.com/2014/10/the-ultimate-guide-to-bootstrap/>

<http://bootstrap.twit.free.fr/index.php>

Outil de test, d'apprentissage et/ou de développement

<http://www.bootply.com>

Utilisation par CDN

CDN : késako ?

CDN : Content Delivery Network : serveur qui fournit les services.

Le CDN va nous permettre de ne rien installer et de récupérer les fichiers directement sur un serveur !

Du coup, il n'y a aucune installation à faire !

Le site bootstrapcdn fournit les fichiers bootstrap .css et .js : <https://www.bootstrapcdn.com>

Google fournit les fichiers jquery .js : <https://developers.google.com/speed/libraries/>

CDN du fichier css : bootstrapcdn

Bootstrapcdn pour le bootstrap.css : <https://www.bootstrapcdn.com>

```
<link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/css/bootstr
ap.min.css" rel="stylesheet">
```

CDN des fichier js : google et bootstrapcdn

Bootstrapcdn pour le bootstrap.js : <https://www.bootstrapcdn.com>

Google pour jquery.js : <https://developers.google.com/speed/libraries/>

```
<script
src="https://code.jquery.com/jquery-1.11.2.min.js"> </script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.1/js/bootstrap
.min.js"></script>
```

Exemple

http://www.w3schools.com/bootstrap/tryit.asp?filename=trybs_temp_blog&stacked=h

Installation sur sa machine : tuto

Utilité : alternative au CDN

Pour pouvoir faire ses tests sans réseau et avoir une version contrôlée de BS et JS, on peut faire une installation sur sa machine.

Le chapitre présente les étapes de l'installation.

Download Bootstrap

<http://getbootstrap.com/getting-started/#download>

Minimum

- "Download Bootstrap" : permet de récupérer les fichiers nécessaires au fonctionnement de Bootstrap. C'est suffisant pour utiliser bootstrap.

Avancé

- "Download source" : permet de récupérer en plus tous les fichiers sources.
- "Download Sass" : c'est un portage de Bootstrap en Sass pour les utilisateurs de projets qui utilisent Sass (Rails, Compass...).

Contenu du téléchargement

Minimum

Si on télécharge seulement Bootstrap :

on obtient **un répertoire « dist »** avec **3 sous-répertoires** :

- dist/css
- dist/fonts
- dist/js.

Avec sources

Si on télécharge les sources : on a un dossier avec plusieurs fichiers et dossiers. Le seul dossier utile est le dossier dist.

Contenu du dossier dist

Sous dossier css

Il contient les fichiers suivants :

- **bootstrap.css** : il comporte les classes de base de Bootstrap
- **bootstrap-theme.css** : il contient des règles de styles particulières pour créer un thème spécifique pour Bootstrap
- **Les versions min** : ce sont les versions de production : le code est sur une ligne pour ne pas prendre toute la place. Les versions non min permettent de naviguer dans le code.

- **Les fichiers map** : bootstrap-theme.css.map et bootstrap.css.map permettent de retrouver l'emplacement original d'une ligne de code à partir du code « min ». Cette fonctionnalité est utilisable avec les dernières versions de Chrome et Firefox. Ces fichiers ne sont pas indispensables au fonctionnement.

Sous dossier fonts

Le répertoire contient des icônes à différents formats : svg, ttf, woff, eot

<http://www.opentuto.com/les-icônes-dans-bootstrap-3/>

<http://getbootstrap.com/components/>

Sous dossier js

- Il contient les fichiers **bootstrap.js** et **jquery.js** ne sont utiles que si on utilise des plugins jQuery
- bootstrap.js contient le code JavaScript des composants de Bootstrap.
- **Les versions min** sont les versions de production : le code est sur une ligne pour ne pas prendre toute la place.

Download de plugin jQuery

Les plugins JQuery ne sont pas installés en standard dans bootstrap.

Ils se récupèrent ici : <http://jquery.com>

Attention, il faut une version < 3 pour que ça marche avec bootstrap (2016) !!!

On trouve ça ici :

<https://blog.jquery.com/2016/05/20/jquery-1-12-4-and-2-2-4-released/>

Il faut les placer dans le dossier js.

Installation du répertoire bootstrap à la racine

A la racine du site, on crée un répertoire bootstrap et on met dedans les 3 sous-répertoires css, fonts et js

Environnement de production

On a besoin uniquement de : css/bootstrap.min.css, le dossier fonts, js/bootstrap.min.js, js/jquery.js

Environnement de développement

On ajoute css/bootstrap.css, le dossier fonts, js/bootstrap.js, js/jquery.js pour pouvoir consulter le fichiers source.

Regarder le code des classes bootstrap

Pour comprendre ce que fait une classe bootstrap, on peut toujours aller regarder le code dans le fichier bootstrap.css.

Charger la bibliothèque bootstrap dans une page HTML

Chargement autonome

➤ *Fichier css*

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

➤ *Fichiers js*

Les fichiers javascript se chargeront ainsi :

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

Head et javascript d'un fichier HTML bootstrap

<head>

Principes

Linker le fichier bootstrap.

Cas particuliers :

- IE
- Mobile
- IE < 9

Exemple : 00 chargerBS.html

Principes :

La page affiche un simple Hello World.

Elle linke le fichier bootstrap.min.css dans le <head>

Elle linke

```
<head>
  <title>Bootstrap template</title>
  <meta charset="utf-8">
  <!--link de bootstrap : -->
  <link href="../bootstrap/css/bootstrap.min.css"
rel="stylesheet">

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

  <!-- pour les mobiles : -->
  <meta name="viewport" content="width=device-width, initial-
scale=1">

  <!-- pour IE 8 -->
  <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></s
cript>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></scrip
t>
  <![endif]-->

</head>
```

<script>

2 fichiers à charger pour le JavaScript :

- le fichier bootstrap
- une api jquery de google

Dans une balise <script>, juste avant le </body>, on met les références aux fichiers .js :

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

La grille

Éléments

Présentation

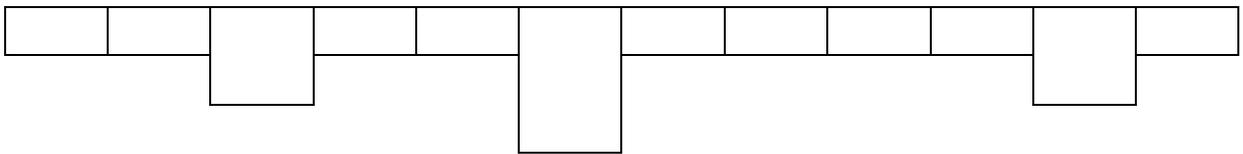
<https://openclassrooms.com/courses/prenez-en-main-bootstrap/une-grille>

<http://getbootstrap.com/css/#grid>

La grille est une division de la page en 12 colonnes qui ont toutes la même largeur.

La largeur des colonnes s'adapte à la largeur de la fenêtre.

Pour les ligne, la taille s'adapte à ce qu'on met dedans. C'est du HTML classique : la hauteur d'une cellule sera fonction de ce qu'on met dedans.



Ici dans les colonnes 3, 6 et 11, la hauteur du bloc s'adapte au texte.

Ligne

```
class=row
```

Colonnes

```
class=col-XX-*
```

XX : xs, sm, md ou lg. Ce sont les formats qui modifient le comportement.

* : de 1 à 12

exemples :

col-xs-1 pour un élément de 1 colonne

col-md-8 pour un élément de 8 colonnes

Container et container-fluid (responsive)

➤ *Principes*

1. La grille doit être contenu dans un container ou container-fluid.
2. contenair et container-fluid sont des blocs de regroupement de balises.
3. Le contenair laisse une marge à droite et à gauche.
4. Le contenair-fluid s'étale sur toute la fenêtre.

Exemples

<http://getbootstrap.com/css/#overview-container>

http://www.w3schools.com/Bootstrap/bootstrap_grid_system.asp

http://www.w3schools.com/bootstrap/bootstrap_get_started.asp

http://www.w3schools.com/bootstrap/bootstrap_grid_stacked_to_horizontal.asp

Exemple : 01_1_grille.html

➤ *Affichage*

Ici on affiche un bloc de 4 et un bloc de 8.

Le bloc de 4 contient beaucoup de texte.



➤ *Test*

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/01_1_grille.html

➤ *HTML*

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-4">
      <p>
        Bloc de 4<br>
        Bla bla bla
      </p>
    </div>
    <div class="col-xs-8">
      <p>
        Bloc de 8
      </p>
    </div>
  </div>
</div>
```

➤ *Structure HTML*

Un div « container-fluid » qui contient nos blocs bootstrap.

Un div pour la ligne : row

Dans la ligne : un div par colonne : une colonne de 4 et une colonne de 8.

➤ *CSS*

```
.row{ /* toutes les lignes */
  text-align: center;
  line-height: 40px; /* hauteur de la ligne */
}

[class*="col-"] { /* tous les attributs class = à %col-% */
  border: 1px solid black;
  border-radius: 6px; /* angles arrondis */
}

[class*="-4"] { /* tous les attributs class = à %-4% */
  background-color: aqua;
}
```

```
}
```

.row : on contrôle la ligne en général : texte centré, etc.

[class*="col-"] : on contrôle toutes les colonnes.

[class*="-4"] : on contrôle les colonnes de largeur 4

➤ **Rappels de syntaxe CSS :**

Poly « Bases du CSS », paragraphe « Syntaxe CSS : construction des sélecteurs » :

[] : cible un attribut de n'importe quel sélecteur (balise).

[class] cible l'attribut « class »

[class*="-4"] : cible les attributs class = %-4% (le nom de la class contient « -4 »).

Exemple : 02 2 grille float.html

➤ **Affichage**

On affiche un bloc de 4 et un bloc de 6. Il reste 2 colonnes qui ne sont pas dans un bloc.

On peut mettre du texte dans ces deux colonnes : il est sur la ligne (dans le div « row »)

Si on met beaucoup de texte, on voit que les blocs sont gérés comme des « float » : le texte qu'on entoure les blocs précédents.

Comportement « float » des blocs :

	bla bla bla bla
	bla
	bla
	bla

➤ **Test**

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/01_2_grille_float.html

Principes

Les formats des colonnes définissent un comportement en fonction de la taille de l'écran.

Les blocs vont s'empiler ou pas en fonction de la taille de l'écran et du format xs, sm, md ou lg.

Le choix du format définit le « empilé si taille < au format ».

- **col-xs-*** : x small (smartphone) : toujours aligné
- **col-sm-*** : small (tablette): aligné si ≥ 768 px, empilé sinon (empilé si smartphone).
- **col-md-*** : medium (ordinateur portable) : aligné si ≥ 992 px, empilé sinon (empilé si tablette).
- **col-lg-*** : large (écran fixe) : aligné si ≥ 1200 px, empilé sinon (empilé si portable).

Exemple : 02_grille_empilement.html

➤ Principes

On présente quatre lignes (row) avec deux colonnes dans chaque ligne : de 4 et de 8.

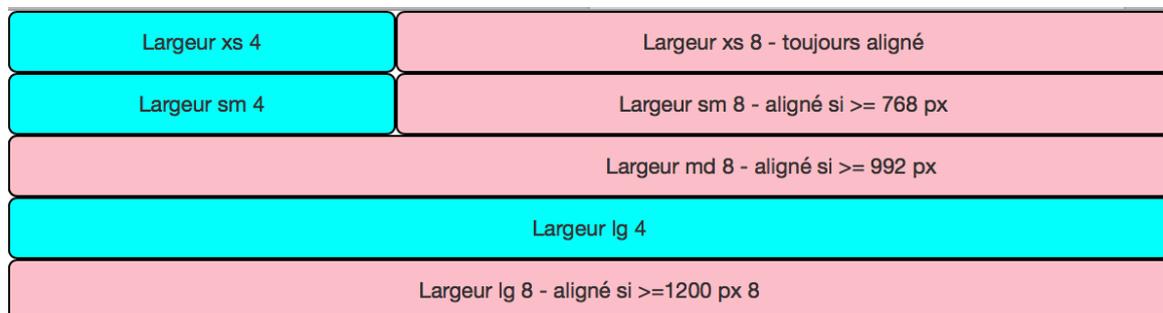
Ces colonnes seront xs, sm, md ou lg.

Leur comportement sera donc différent selon la taille de l'écran.

Les colonnes lg sont les premières à s'empiler quand l'écran diminue.

Les colonnes xs ne s'empileront jamais.

A noter que quand on empile, les deux blocs (de 4 et de 8) ont alors la même largeur.



tablette - sm : 792

➤ Test

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/02_grille_empilement.html

➤ HTML

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-4">Largeur xs 4</div>
    <div class="col-xs-8">Largeur xs 8 - toujours aligné</div>
  </div>

  <div class="row">
    <div class="col-sm-4">Largeur sm 4</div>
    <div class="col-sm-8">Largeur sm 8 - aligné si  $\geq 768$ 
px</div>
```

```

</div>

<div class="row">
  <div class="col-md-4">Largeur md 4</div>
  <div class="col-md-8">Largeur md 8 - aligné si >= 992
px</div>
</div>

<div class="row">
  <div class="col-lg-4">Largeur lg 4</div>
  <div class="col-lg-8">Largeur lg 8 - aligné si >=1200 px
8</div>
</div>

</div>

<script>
function myFunction() {
  document.getElementById("largeur").innerHTML = "Largeur de la
fenêtre: " + window.innerWidth;
}
</script>
</body>

```

➤ CSS

```

.row{ /* toutes les lignes */
  text-align: center;
  line-height: 40px; /* hauteur de la ligne */
}
[class*="-4"] { /* tous les attributs class = à %-4% */
  background-color: aqua;
}
[class*="-8"] { /* tous les attributs class = à %-8% */
  background-color: pink;
}
[class*="col-"] { /* tous les attributs class = à %col-% */
  border: 1px solid black;
  border-radius: 6px; /* angles arrondis */
}

```

➤ Remarque JavaScript

On définit une fonction javascript qui affiche la largeur de l'écran en pixel et le type d'écran (smartphone, tablette, ordi portable, ordi fixe).

Cette fonction met à jour la page HTML avec un « innerHTML » (gestion du DOM).

Elle est appelée par un « onload » et un « onresize » dans le <body> (gestion d'événement).

➤ Remarque

Il faut que les deux colonnes aient le même format (lg dans col-lg-4 et col-lg-8 par exemple) sinon on obtient un comportement étrange.

Mise en page avec la grille

<http://bootstrap.twit.free.fr/grille/page01.php>

Lignes et colonnes variables

➤ Exemple : 03_grille_complexe.html

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/03_grille_complexe.html

1 col	2 colonnes	3 colonnes	6 colonnes
12 colonnes			
4 colonnes		8 colonnes	

```
<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>
```

Sauter des colonnes : offset

➤ *Syntaxe*

`class="col-XX-offset-*`

➤ *Exemple : 04_grille_a_trous.html*

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/04_grille_a_trous.html



```
<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>
```

Imbrication d'éléments

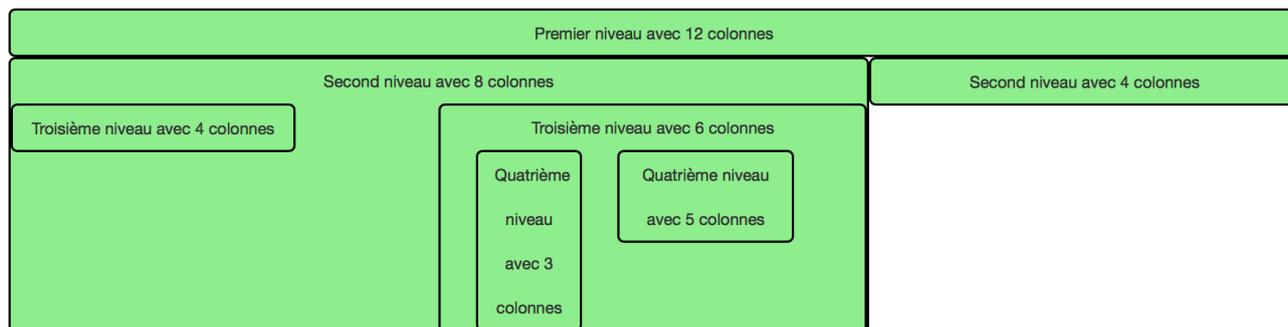
➤ Principes

Une première ligne indépendante de 12 colonnes

Une deuxième ligne avec 8 et 4 colonnes

Dans le bloc de 8 colonnes, un 3^{ème} ligne avec 4 colonnes et 6 colonnes avec un offset de 2

Dans ce bloc de 6 colonnes, une 4^{ème} ligne avec 3 colonnes et 1 offset de 1, 5 colonnes et un offset de 1



➤ Test

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/05_grille_imbrique.html

Décaler des colonnes

http://www.w3schools.com/bootstrap/bootstrap_grid_examples.asp

➤ *Syntaxe*

class="col-XX-push-*" : décaler vers la droite, à partir de sa position « normale »

class="col-XX-pull-*" : décaler vers la gauche, à partir de sa position « normale »

A noter que le comportement est complexe à analyser du fait d'un décalage à partir de la position « normale ».

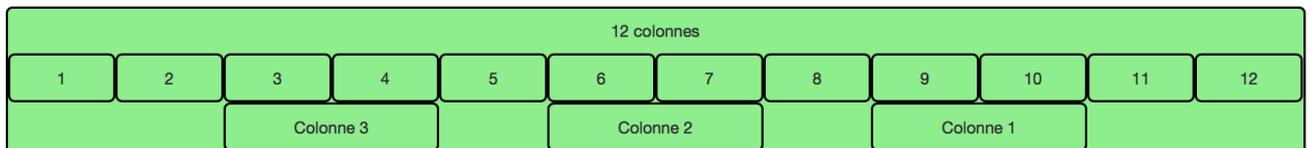
➤ *Exemple : 06_grille_decale.html*

Principes

Colonne 1 est poussée à droite de 8 à partir de 0 (du bord)

Colonne 2 est poussée à droite de 3 à partir de sa position « normale » qui serait juste après colonne 1 si celui-ci n'avait pas été poussé : donc, à partir de 2 : il démarre en 5.

Colonne 3 est poussée à gauche de 2 à partir de sa position « normale » qui serait juste après colonne 2 si colonne 1 et 2 n'avait pas été poussée : donc à partir de 4 : il démarre en 2



➤ *Test*

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/06_grille_decale.html

➤ *Code*

```
<div class="row">
  <div class="col-lg-2 col-lg-push-8">Colonne 1</div>
  <div class="col-lg-2 col-lg-push-3">Colonne 2</div>
  <div class="col-lg-2 col-lg-pull-2">Colonne 3</div>
</div>
```

Mise en page « classique »

Mise en page classique

Principes

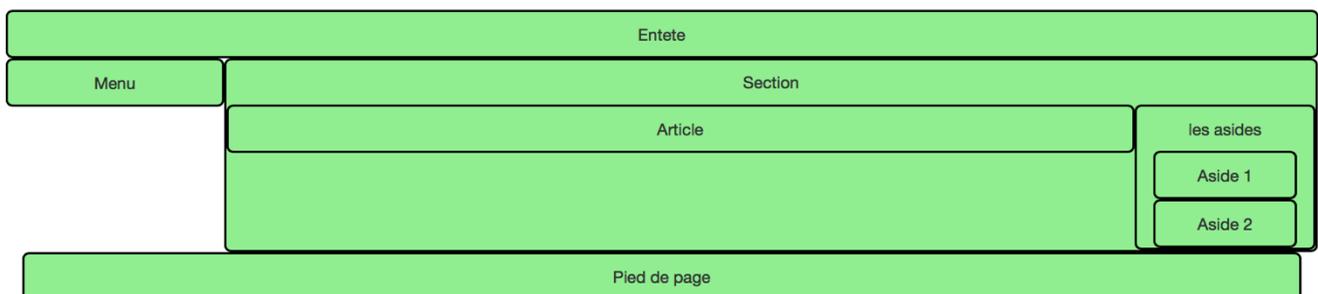
Le header est une ligne (row), sur 12 colonnes. Idem pour le footer.

On définit une ligne (row) qui contiendra le menu (nav) sur 2 colonnes, et la section sur 10 colonnes.

Dans la section, on définit une ligne qui contient l'article sur 10 colonnes, et div pour les asides de 2 colonnes.

Dans le div des asides, on place les aside, sur 12 colonnes chacun.

Pour le footer, on choisit de ne pas mettre de ligne, mais de le placer sur 12 colonnes.



Test

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/07_grille_mise_en_page_v1.html

Code

```
<header class="row">
  <div class="col-lg-12">
    Entete
  </div>
</header>

<div class="row">
  <nav class="col-sm-2">
    Menu
  </nav>
  <section class="col-sm-10">
    Section
    <div class="row">
      <article class="col-sm-10">
        Article
      </article>
      <div class="col-sm-2"><!--les asides-->
        les asides
        <aside class="col-sm-12">
          Aside 1
        </aside>
        <aside class="col-sm-12">
```

```
        Aside 2
      </aside>
    </div><!--les asides-->
  </div>
</section>
</div>

<footer class="col-lg-12">
  Pied de page
</footer></footer>
```

Version avec plusieurs sections

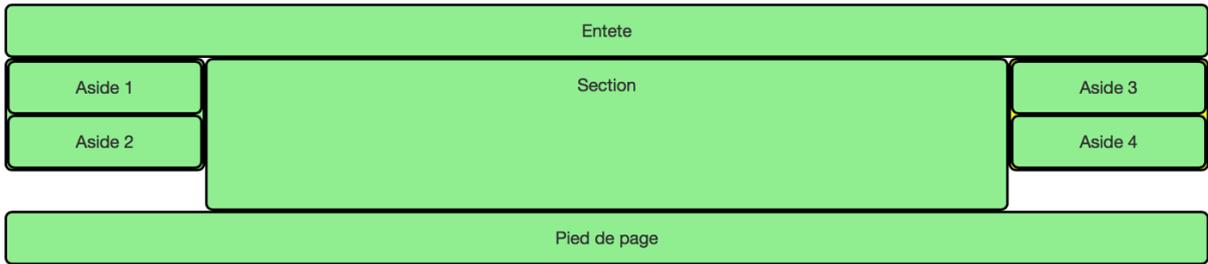
Comme le précédent mais avec 2 plusieurs sections.

Test

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/07_grille_mise_en_page_v2.html

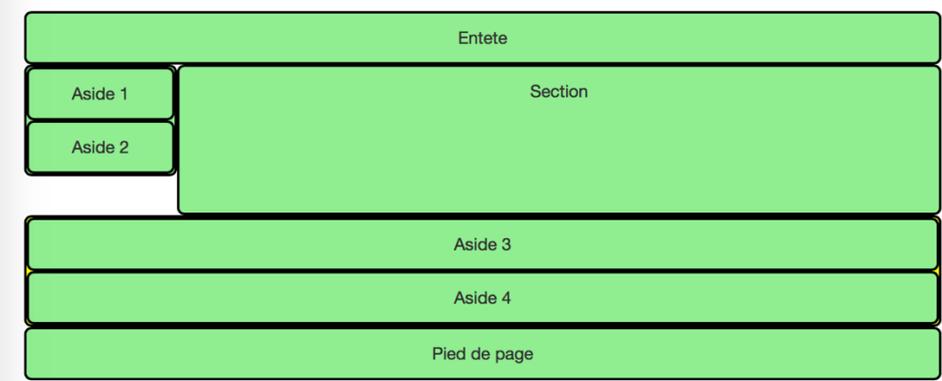
Objectifs

➤ *Fixe et portable : lg et md*



➤ *Tablette : sm*

Aside 3, 4 empilés

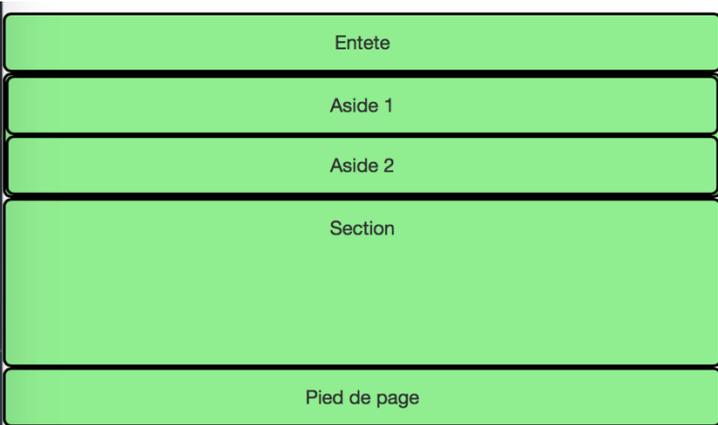


tablette - sm : 789

➤ *Smartphone : sm*

Tout empilé.

Aside 3,4 cachés



smartphone - xs : 523

Tests

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_01/07_grille_mise_en_page_v3.html

Solutions

➤ *Syntaxe*

```
<div class="clearfix visible-sm-block"></div>

<div class="hidden-xs col-md-2" style="background:yellow">
```

➤ *Définition du comportement selon la taille de la fenêtre*

Plutôt que d'avoir une simple définition de colonnes par bloc comme :

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

On va définir plusieurs comportements selon la taille de la fenêtre :

```
<div class="col-xs-4 col-sm-3 col-md-2">
```

Avec cette définition, on a un affichage sur 2 colonnes sur fixe et portable (col-md-2), sur 3 colonnes sur tablette (col-sm-3) et sur 4 colonnes sur smartphone (col-xs-4).

➤ *Solution*

On a 3 « row » : header, asides et section, footer

Les row header et footer s'étalent sur 12 colonnes.

La row avec l'aside gauche (1,2), la section et l'aside droite (3,4) est divisée en 3 blocs avec leurs nombres de colonnes.

Ce nombre de colonnes est fonction de la taille de l'écran.

```
<header class="row"><div class="col-md-12">

<div class="row">
  <!-- affichages des aside 1,2 - section - aside 3,4
  lg, md : sm2 - md8 - md2
  sm : sm2 - sm10 - 12(empilé)
  xs : 12(empilé) - 12(empilé) - hidden(caché)
  -->

  <div class="col-sm-2" > aside ...

  <section class="col-sm-10 col-md-8" > ...

  <div class="hidden_xs col-md-2" > aside ...

</div>

<footer class="row"><div class="col-md-12">
```

➤ *Ordinateur : lg et md*

Aside 1,2 sur 2 colonnes, Section sur 8 colonnes, Aside 3,4 sur 2 colonnes.

➤ *Tablette : sm*

Aside 3, 4 empilés

Comme il y a un « float » entre aside 1, 2 et section, l'aside 3,4 se place mal et recouvre les blocs aside 1,2 et section.

On doit réinitialiser les float avec un div vide en class clearfix et visible-sm-block

visible-sm-block : ne s'affiche qu'au format tablette

```
.clearfix:after { clear: both;}
```

```
<div class="clearfix visible-sm-block"></div>
```

➤ **Smartphone : sm**

Pour cacher les Aside 3,4 :

```
<div class="hidden-xs col-md-2">
```

Mise en page à photos

Problème de résolution

Principe

Avec des petites photos, il faut faire attention à ce que, quand on empile, elles ne deviennent pas grandes... et floues !

Exemple flou !

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_02/01_grille_a_photos_floues.html

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

On affiche une photo sur 2 colonnes.

On aura 6 photos sur une ligne (6*2=12 colonnes).

Mais dès que la fenêtre passe sous 1200 pixels de large, les photos s'empilent en pleine largeur. Elles risquent d'être floues !

Solution

➤ *Définition du comportement selon la taille de la fenêtre*

Plutôt que d'avoir une simple définition de colonnes par bloc comme :

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

On va définir plusieurs comportements selon la taille de la fenêtre :

```
<div class="col-xs-4 col-sm-3 col-md-2">
```

Avec cette définition, on a un affichage sur 2 colonnes sur fixe et portable (col-md-2), sur 3 colonnes sur tablette (col-sm-3) et sur 4 colonnes sur smartphone (col-xs-4).

➤ *La solution : contrôler le nombre de photos affichées pour chaque écran*

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_02/02_grille_a_photos.html

```
<div class="col-xs-4 col-sm-3 col-md-2">  
    
</div>
```

On affiche une photo sur 2 colonnes, donc 6 photos par ligne pour un ordinateur portable et un écran fixe.

Quand on passe à un smartphone, on affiche sur 4 colonnes donc 3 photos par ligne.

Quand on passe sur une tablette, on affiche sur 3 colonnes donc 4 photos par ligne.

Autres exemples

Pagination de photos avec JavaScript

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_02/03_pagination-javascript.html

Listes en tout genres

http://bliaudet.free.fr/Cours/HTML/51_Bootstrap/BS_tests_serie_02/04_des_listes.html

Exemple de page avancée

http://bootstrap.twit.free.fr/tutov3_3/base13.html

Composants de base Bootstrap

Mise en valeur du texte

Jumbotron : http://www.w3schools.com/bootstrap/bootstrap_jumbotron_header.asp

Badge et label : http://www.w3schools.com/bootstrap/bootstrap_badges_labels.asp

Liste

http://www.w3schools.com/bootstrap/bootstrap_list_groups.asp

Table

http://www.w3schools.com/bootstrap/bootstrap_tables.asp

Formulaire

http://www.w3schools.com/bootstrap/bootstrap_forms.asp

Site générateur de formulaire (voir la vue HTML) :

<http://bootsnipp.com/forms>

Image

http://www.w3schools.com/bootstrap/bootstrap_images.asp

Bouton

http://www.w3schools.com/bootstrap/bootstrap_buttons.asp

Site générateur de bouton :

<http://bootsnipp.com/buttons#>

autre site :

http://charliepark.org/bootstrap_buttons/

Icones

http://www.w3schools.com/bootstrap/bootstrap_glyphicons.asp

Site de glyphicons :

<https://glyphicons.com>

autre site d'icone

<http://fontawesome.io>

etc. !

Composants plus complexes Bootstrap

Collapse

http://www.w3schools.com/bootstrap/bootstrap_collapse.asp

Barre de navigation

http://www.w3schools.com/bootstrap/bootstrap_navbar.asp

Pagination

http://www.w3schools.com/bootstrap/bootstrap_pagination.asp

Exemple de pagination avec JQuery et Javascript :

➤ *Exemple : 11_pagination-javascript.html*

Liste et panel

http://www.w3schools.com/bootstrap/bootstrap_list_groups.asp

http://www.w3schools.com/bootstrap/bootstrap_panels.asp

➤ *Exemple : 12_des_listes.html*

Media

Pour mettre une photo ou une vidéo callée à droite ou à gauche avec du texte.

http://www.w3schools.com/bootstrap/bootstrap_media_objects.asp

Navigation

Chemin de navigation : breadcrumb

http://www.w3schools.com/bootstrap/bootstrap_ref_comp_navs.asp

Carroussel

http://www.w3schools.com/bootstrap/bootstrap_carousel.asp

etc. !

Templates Bootstrap

http://www.w3schools.com/bootstrap/bootstrap_templates.asp

<http://getbootstrap.com/getting-started/#examples-framework>