

Le système de grille

Avec Bootstrap, il n'est plus nécessaire d'écrire des requêtes CSS **media queries** : Bootstrap l'a déjà fait pour vous ! Il d'utiliser des classes CSS correspondant aux points de rupture selon le tableau suivant :

	Extra small ≤576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				

Le principe de grille consiste à découper votre page en cellules, comme dans un tableau, donc à créer des lignes et des colonnes.

Bootstrap découpe la largeur d'écran en **12 blocs de largeurs égales** (égales mais variables selon la largeur du dispositif sur lequel on affiche). Les classes `col-` vont spécifier combien de blocs on veut utiliser.

Par exemple, pour faire 2 colonnes, la première avec une largeur de 8 blocs et la seconde avec une largeur de 4 blocs, on écrira :

```
<div class="container">
  <div class="row">
    <div class="col-8">Colonne avec une largeur de 8 blocs</div>
    <div class="col-4">Colonne avec une largeur de 4 blocs</div>
  </div>
</div>
```

Conteneurs

Pour créer une grille, il faut d'abord l'ouvrir avec une balise `<div>` avec la classe CSS `container`, laquelle a une largeur par défaut de 980px sur PC (donc sur un écran plus grand que 980px on obtient un effet de centrage).

```
<div class="container">
  <div class="row">
    <div class="col-8">Colonne avec une largeur de 8 blocs</div>
    <div class="col-4">Colonne avec une largeur de 4 blocs</div>
  </div>
</div>
```

Il existe une seconde classe de conteneur, `container-fluid`, permettant d'utiliser toute la largeur de l'écran (100%).

Dans une page HTML, il faut au moins un conteneur (dans ce cas il sera placé immédiatement sous `<body>` et se terminera à la fin de la page, avant la fermeture de `</body>`.

Il est possible d'utiliser plusieurs conteneurs dans une page, y compris mixer les classes `container` et `container-fluid`.

Dans le conteneur, on placera tout le reste de la grille : lignes (rangées) et colonnes.

Les lignes (= rangées)

On crée ensuite une rangée (ligne) en appliquant la classe `.row` (= rangée) à une seconde balise `<div>` :

```
<div class="container">
  <div class="row">
    [ ... ]
  </div>
</div>
```

Les colonnes

On découpe ensuite une ligne en colonnes en appliquant la classe `col` à une ou plusieurs balises `<div>` :

```
<div class="container">
  <div class="row">
    <div class="col-12">Une (seule) colonne occupant toute la largeur</div>
  </div>
</div>
```

Une ligne est découpée en **12 blocs de largeur égale**, on va préciser combien de ces blocs une colonne utilise, en ajoutant ce nombre après le mot `col` (plus un tiret), par exemple :

```
<div class="container">
  <div class="row">
    <div class="col-8">Une première colonne qui utilise 8 blocs</div>
    <div class="col-4">Une seconde colonne qui utilise les 4 blocs restants</div>
  </div>
</div>
```

Dans cet exemple, il n'y a que 2 colonnes, une de 8 blocs et une de 4 blocs, le total étant bien de 12.

Testez l'exemple.

Il existe de nombreuses possibilités de découpe, le total de blocs utilisés devant toujours être égal à 12, par exemple :

- 3 colonnes de 4 blocs
- 6 colonnes de 2 blocs
- 2 colonnes de 6 blocs (largeur égale/divison de la zone en 2)
- 1 colone de 7 blocs et une seconde de 5 blocs etc.

A noter :

- Chaque colonne peut être redécoupée en créant une ligne elle-même redécoupée en colonnes
- Les colonnes peuvent dimensionnées automatiquement, en indiquant uniquement `col` comme classe ou en ajoutant le suffixe `auto` (dans ce cas la colonne est ajustée à la largeur de son contenu)
- Les colonnes peuvent être "sautées" via la classe `offset`

Ces 2 points permettent d'effectuer des mises en page assez complexes.

[documentation sur la grille](#)

Affichage responsive

Toute la "magie" de Bootstrap est ici ! Nous allons mettre en oeuvre les points de rupture (breakpoints).

Reprenons l'exemple précédent (colonnes) :

```
<div class="container">
  <div class="row">
    <div class="col-8">Une première colonne qui utilise 8 blocs</div>
    <div class="col-4">Une seconde colonne qui utilise les 4 blocs restants</div>
  </div>
</div>
```

Testez l'exemple.

Ce code s'affiche de la même façon sur tout les dispositifs. Mais on pourrait souhaiter obtenir un affichage différent selon la largeur d'écran.

Par exemple :

- Règle 1* : Une page présentant deux colonnes : une partie centrale d'une largeur de 8 blocs et une seconde colonne de largeur 4 sur les plus grands écrans.
- Règle 2* : pour les écrans supérieurs à 992px, la colonne de droite doit être un peu plus large; on a donc une largeur de 8 pour la partie centrale et une largeur de 4 pour l'autre colonne.
- Règle 3* : sur tablettes en mode portrait, la colonne principale aura une largeur de 7 blocs et la seconde en occupera les 5 restant.
- Règle 4* : nos 2 colonnes auront une largeur égale pour les dispositifs à partir de 576px.
- Règle 5* : sur les mobiles, l'affichage se fait sur une seule colonne.

Pour cela, on va ajouter un préfixe de point de rupture entre `col` et le nombre de blocs, par exemple `col-md-7` va indiquer que pour la dimension `md`, c'est-à-dire de 768 à 991 pixels, la colonne occupera 7 blocs.

On va répéter ce principe pour tous les points de rupture :

```
<div class="container">
  <div class="row">
    <div class="col-12 col-sm-6 col-md-7 col-lg-8 col-xl-9">Colonne 1</div>
    <div class="col-12 col-sm-6 col-md-5 col-lg-4 col-xl-3">Colonne 2</div>
  </div>
</div>
```

Testez l'exemple.

Rappel : Bootstrap 4 étant *Mobile first*, on part d'abord de la dimension la plus petite, celle pour mobile (< 576px), pour aller vers la plus grande (xl).

Ici on a toujours 2 colonnes, sauf sur mobile puisqu'on a indiqué que les colonnes y occupaient les 12 blocs disponibles; la seconde colonne se placera alors sous la 1^{ère}.

Dernier point, on a indiqué ici des souhaits pour tous les points de rupture, mais ce n'est pas obligatoire

Masquer une zone

Sur mobile, il peut être utile de ne pas afficher une zone, pour améliorer le temps de chargement ou parce que le contenu n'est pas pertinent (publicité par exemple).

Bootstrap propose un ensemble de classes pour afficher/masquer des éléments HTML selon les breakpoints :

Exemples

```
<div class="d-none d-sm-block">
  Ce bloc sera masqué pour tout (d-none), et l'affichage sera rétabli à partir de sm, c'est-à-dire pour les éc
</div>

<div class="d-lg-none d-xl-block">
  Ce bloc sera masqué à partir de la plage de breakpoints lg (de 992 à 1199px), puis rétabli à partir des écri
</div>
```

Explications :

- le préfixe `d` signifie *display*, c'est-à-dire afficher.
- `none` signifie *aucun*, c'est-à-dire masquer, équivalent de `display: none` en CSS.
- `block` signifie afficher, équivalent de `display: block` en CSS.
- les préfixes de breakpoints `xs`, `sm` etc. indiquent à quelle(s) plage(s) de breakpoints afficher/masquer
- S'il n'y a pas de préfixe de classes (par exemple `d-none`), la règle est s'applique à tout les breakpoints, sauf mention contraire annulant cette règle.

Pour bien comprendre, consultez les exemples de [la documentation](#).