

Formulaires

Un champ de formulaire standard (`input type="text"`) est défini par le code suivant dans Bootstrap :

```
<form>
  <div class="form-group">
    <label for="courriel">Adresse email</label>
    <input type="email" class="form-control" name="courriel" id="courriel" placeholder="Saisir votre adresse" />
  </div>
</form>
```

Testez l'exemple.

- Les champs `input` et le `label` associé sont groupés dans un `<div>` qui prend en attribut la classe `form-group`
- Les champs `input` prennent une classe `form-control`.

Ensuite, il existe différentes variantes, dont des classes : [Documentation](#)

Messages (alertes)

Bootstrap propose un système de [messages](#) (boîtes d'alerte) pour communiquer avec l'utilisateur.

Boutons

Bootstrap permet d'habiller des boutons avec des classes de couleurs et de taille.

Exemple 1 : un bouton avec fond bleu et texte blanc :

```
<button type="button" class="btn btn-primary">Primary</button>
```

Primary

Exemple 2 : ajout de la classe `btn-lg` pour obtenir le même bouton qu'à l'exemple 1 mais en plus grand et vert:

```
<button type="button" class="btn btn-success btn-lg">Primary</button>
```

Primary

Ces classes de boutons peuvent être utilisées sur les liens (balises `<a>`).

[Documentation](#)

Cards

Bootstrap 4 apporte un nouveau composant, les [cards](#), qui permettent l'habillage de blocs de contenus, [exemples](#).

Une alternative au *cards* : [les médias](#).

Images

Bootstrap fournit des classes pour placer, habiller (bordures, arrondis) ou redimensionner [les images](#).

Pour afin de les rendre responsive, il suffit d'appliquer la classe `img-fluid` sur la balise `` tout en veillant à ne pas spécifier de dimensions fixes : suppression des attributs `width` et `height`), ce qui donne :

```

```

La classe `img-fluid` applique les règles CSS suivantes aux balises `` :

- `max-width: 100%;` : fixe la largeur de l'image à celle de son parent (pour résumer).
- `height: auto;` : adapte la hauteur de l'image en fonction de sa largeur, afin de ne pas obtenir de distorsion de l'image.

En dehors de cette classe et de Bootstrap, [d'autres solutions](#) existent pour gérer la problématique responsive des images : plusieurs dimensions d'une même image (attributs `srcset` et `sizes` , balise `<picture>`), propriétés css (`cover` etc.).

Enfin, Bootstrap fournit d'autres classes applicables aux images pour gérer les miniatures, les arrondis et leur placement.

En ce qui concerne les vidéos, voir cette page.

Typographie

Bootstrap fournit des classes pour gérer la typographie : polices, taille, alignement, gras, italique, titres (`<h1>` , `<h2>` ...), mise en exergue, habillage des listes (`` , ``) etc.

Polices

La police de caractère affichée par défaut varie en fonction du [système d'exploitation](#), par exemple :

- Sego UI* sous Windows
- Roboto* sous Android

La police par défaut peut bien entendu être changée, via une feuille de style CSS.

Taille

Bootstrap définit par défaut la taille de police à **16 pixels**, donc une unité de taille fixe. En effet, Bootstrap ne gère pas le redimensionnement des textes pour qu'ils deviennent « responsive » : il faudra ajouter des media queries spécifiques, nous y reviendrons plus tard.

Alignement

Il existe des classes pour l'alignement du texte, équivalents de `text-align` en CSS :

- `text-left` : aligne le texte à gauche de son conteneur (par défaut)
- `text-right` : aligne le texte à droite de son conteneur
- `text-center` : centre le texte dans son conteneur
- `text-justify` : justifie le texte

Exemple :

```
<div class="row">
  <div class="col-12">
    <p class="text-left">Texte aligné à gauche (normal)</p>
    <p class="text-right">Texte aligné à droite</p>
    <p class="text-center">Texte centré</p>
    <p class="text-justify">Texte justifié. Lorem ipsum dolor sit amet, consecteturadipiscing elit. Donec ege
  </div>
</div>
```

Ces classes peuvent être spécifiées pour un dispositif en particulier grâce aux préfixes de breakpoints : par exemple `text-sm-center` centrera le texte uniquement à la dimension correspondant à `sm` , c'est-à-dire à partir de 576 pixels.

- [Documentation](#)