

*@ Utopios Consulting*



# Bootstrap

**Un framework pour le style**

*@ Utopios Consulting*

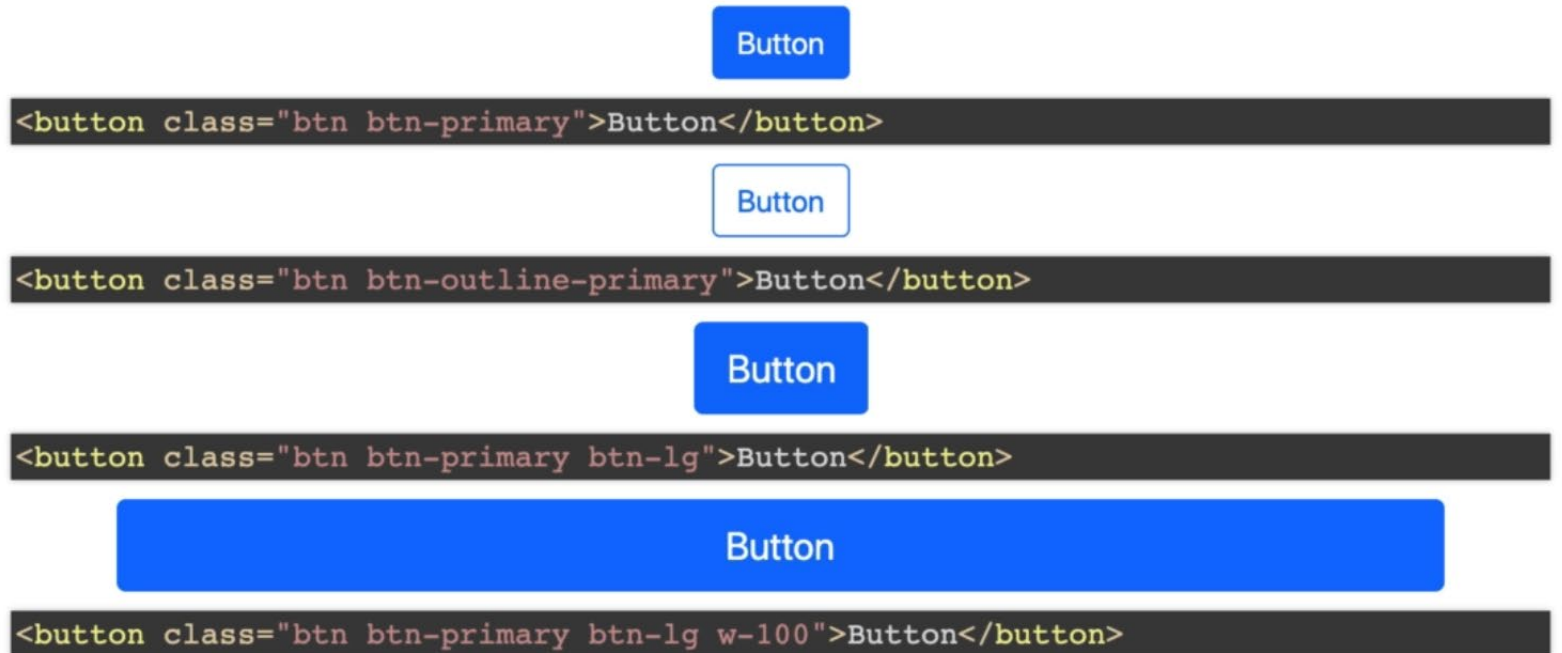
# **Introduction à Bootstrap**

# Qu'est-ce que Bootstrap ?

Bootstrap est un framework **gratuit** et **open-source** prévu pour le front-end. Il fut inventé dans le but d'aider la production de sites webs via la logique du « mobile first » (on commence par designer son site en pensant à son layout mobile) et est hautement customisable.

Ses téléchargements s'élèvent à plus de 2,9 millions pour NPM et le projet a été forked plus de 71 000 fois sur Github dans le but de récupérer plus de 60 composants créés avec le Sass.

L'objectif de Bootstrap est d'aider les développeurs à créer des composants HTML et CSS rapidement sans avoir à manipuler beaucoup de CSS. Pour ce faire, le framework embarque des classes pré-faites que l'on peut utiliser en fonction de nos besoins.



# Installer Bootstrap

Bootstrap peut être installé sur notre ordinateur dans le but de le posséder localement et de ne pas dépendre d'un serveur externe pour son fonctionnement. Pour ce faire, il nous suffit de nous rendre sur le site de Bootstrap :

<https://getbootstrap.com/>

Une fois fait, il nous suffit de cliquer sur le bouton de téléchargement, qui nous emmène à la page ci-contre :

Cette page propose plusieurs versions de bootstrap, dont une minifiée (dont sa taille a été réduite par la compression du texte).

Les deux autres solutions d'installation de Bootstrap sont de se servir du Node Package Manager (NPM) ou de passer par le CDN.

Dans la majorité des cas de développement, l'utilisation du CDN est la plus simple et permet de tester notre front-end sans avoir à se soucier du changement de versions. Pour ce faire, il nous faut copier aux bons endroits les deux balises (link et script) pour ajouter Bootstrap à nos pages webs.

## Download

[View on GitHub](#)

Download Bootstrap to get the compiled CSS and JavaScript, source code, or include it with your favorite package managers like npm, RubyGems, and more.



If your product accepts JavaScript, then it can import CSVs with Flatfile. Start free today.

ads via Carbon

## Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.2.0-beta1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

[Download](#)

# Installer Bootstrap

Bootstrap peut être installé sur notre ordinateur dans le but de le posséder localement et de ne pas dépendre d'un serveur externe pour son fonctionnement. Pour ce faire, il nous suffit de nous rendre sur le site de Bootstrap :

<https://getbootstrap.com/>

Une fois fait, il nous suffit de cliquer sur le bouton de téléchargement, qui nous emmène à la page ci-contre :

Cette page propose plusieurs versions de bootstrap, dont une minifiée (dont sa taille a été réduite par la compression du texte).

Les deux autres solutions d'installation de Bootstrap sont de se servir du Node Package Manager (NPM) ou de passer par le CDN.

Dans la majorité des cas de développement, l'utilisation du CDN est la plus simple et permet de tester notre front-end sans avoir à se soucier du changement de versions. Pour ce faire, il nous faut copier aux bons endroits les deux balises (link et script) pour ajouter Bootstrap à nos pages webs.

## Download

[View on GitHub](#)

Download Bootstrap to get the compiled CSS and JavaScript, source code, or include it with your favorite package managers like npm, RubyGems, and more.



If your product accepts JavaScript, then it can import CSVs with Flatfile. Start free today.

ads via Carbon

## Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.2.0-beta1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

[Download](#)

*@ Utopios Consulting*

# **Le Layout**

# Les Containers

Utopios Consulting

Les **Containers** sont les éléments permettant à Bootstrap de contenir une série d'éléments HTML de façon à ce qu'ils disposent tous de **marges latérales adaptées**. Il en existe de plusieurs types, et chaque type de container va posséder des **breakpoints** différents. Ces breakpoints vont altérer le comportement du conteneur une fois atteints.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

Pour réaliser un conteneur, il suffit de créer une **div** possédant la classe de conteneur que l'on souhaite, sous la forme **container-<breakpoint>**

```
<div class="container-md">L
```



# Les Grids

@ Utopios Consulting

Bootstrap se sert énormément d'un système de « grids » pour proposer un layout responsive (qui s'adaptera aux différentes résolutions d'écrans). Ces grids se basent sur les containers pour fonctionner, et demandent ainsi de faire partie d'une div possédant une classe de type container pour fonctionner.

Une fois cette première condition satisfaite, on peut créer des lignes via l'utilisation de la classe **row** qui contiendra des colonnes responsives via l'utilisation de la classe **col**. Il est possible de fixer le nombre de colonnes disponibles (le système de grids Bootstrap repose par défaut sur des lignes allant jusqu'à **12 sections**) dans la ligne via le classe **row-cols-<valeur>** (une version responsive : **row-cols-<breakpoint>-<valeur>** est également disponible)

Ces colonnes peuvent également posséder des breakpoints et des tailles. Lorsque aucune dimensions latérales n'est spécifié, les colonnes se répartiront équitablement à l'horizontal. Par contre, si l'on spécifie manuellement des tailles (**col-<valeur>**), alors on peut créer des colonnes plus grandes que les autres. **col-auto** va permettre de fixer la largeur de la colonne à son contenu.

Ces valeurs de grandeurs peuvent également être conditionnées par les breakpoints évoqués précédemment, sous la forme de **col-<breakpoint>-<valeur>**





# L'Alignement des Columns

Il est possible de gérer le comportement des colonnes via l'utilisation de classes. Par exemple, on peut gérer l'alignement vertical ou horizontal des colonnes, leur répartition sur les axes horizontaux des rows, changer leurs ordres ou faire un espace fantôme sur leurs côtés.

Pour gérer l'alignement vertical, il suffit d'utiliser la classe **align-items-<type>** dans le but de centrer vers le haut, le milieu ou le bas les **col** contenues dans le **row**. Chaque colonne peut être alignée indépendamment via la classe **align-self-<type>**

Pour l'alignement horizontal, il est possible d'utiliser la classe **justify-content-<type>** pour aligner les **col** au sein d'un **row**. Il est également possible de créer de l'espace sur les côtés de la colonne via l'utilisation d'offsets. Pour ce faire, il faut utiliser **offset-<breakpoint>-<valeur>**. Une autre façon de faire est de spécifier des marges via l'utilisation des classes **m<orientation>-<breakpoint>-<valeur>** (l'orientation des marges peut être principalement **s** pour start, **e** pour end, **t** pour top et **b** pour bottom).

.col-md-4

.col-md-4 .ms-auto

.col-md-3 .ms-md-auto

.col-md-3 .ms-md-auto

.col-auto .me-auto

.col-auto

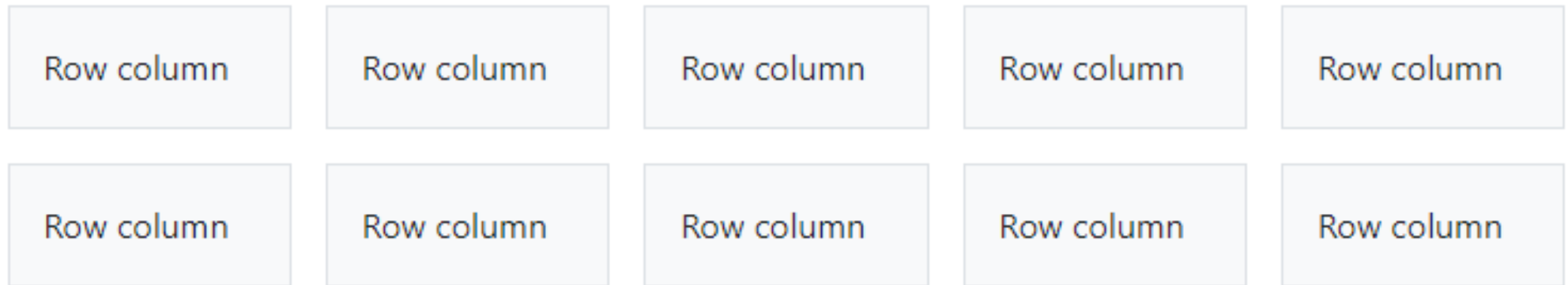
# Les Gutters

@ Utopios Consulting

Les **Gutters** servent dans Bootstrap à créer une bordure entre les colonnes et / ou les lignes de sorte à rendre notre layout plus jolis et présenter d'une meilleure façon nos différents éléments HTML.

Pour créer une gouttière horizontale, il suffit d'utiliser la classe **gx-<breakpoint>-<valeur>**. De même, pour l'axe Y, il suffit de se servir de **gy-<breakpoint>-<valeur>** pour obtenir une gouttière verticale. Une combinaison des deux peut se faire via l'utilisation de **g-<breakpoint>-<valeur>**

Par défaut, le système de grid Bootstrap n'utilise aucune gouttière. Il est donc nécessaire d'en ajouter une au besoin. Il est également important de noter qu'une utilisation de gouttière trop large peut demander de modifier la valeur de padding de nos colonnes via l'utilisation de la classe **p<direction>-<breakpoint>-<valeur>**



*@ Utopios Consulting*

# **Le Content**

# La Typographie *Utopios Consulting*

Bootstrap propose également une mise en forme de la typographie. Par exemple, les headings (h1, h2, etc...) sont mis en forme par défaut par Bootstrap. Il est cependant possible de donner une mise en forme de heading à du texte via l'utilisation de la classe **h<valeur>**. D'autres types de headings existent en la classe **display-<valeur>**. Une variation supplémentaire du texte est disponible via la classe **text-muted**. Toujours dans la modification du texte : Pour tronquer un texte trop long, la classe **text-truncated** est également disponible.

Lorsque l'on réalise des listes, il est possible d'en modifier le style avec par exemple la classe **list-unstyled** qui permet de retirer les puces et de présenter les listes comme des blocks de texte verticaux.

Si l'on veut réaliser des listes présentées horizontalement, il est possible de le faire via l'utilisation de la classe **list-inline** dans le parent et **list-inline-item** dans les enfants

L'alignement du texte est facilement modifiable via l'utilisation de la classe **text-<alignement>**

A well-known quote, contained in a blockquote element.

— Someone famous in *Source Title*

A well-known quote, contained in a blockquote element.

— Someone famous in *Source Title*

A well-known quote, contained in a blockquote element.

— Someone famous in *Source Title*

# Les Images

@ Utopios Consulting

Bootstrap permet également la manipulation du caractère responsive d'une image. Pour rendre une image responsive, il n'y a rien de plus simple ! Il suffit de lui ajouter la classe **img-fluid**.

A côté de ça, on peut facilement altérer une image en lui donnant un caractère de « vignette » par l'ajout de la classe **img-thumbnail**. Ces images peuvent obtenir également un arrondi par l'ajout de la classe **rounded**. Leur alignement se fait quant à lui via l'utilisation de **float-<direction>** ou de marges par utilisation des marges Bootstrap vues précédemment (par exemple **mx-auto**)



Image Heading Here



Image Heading Here



Image Heading Here

# Les Tableaux

@ Utopios Consulting

En nous servant de Bootstrap, il est possible de réaliser d'élégants tableaux très facilement. Il nous suffit pour cela d'ajouter la classe **table** à notre balise **table**. D'autres designs de tableaux sont disponibles (table-primary, table-danger, etc...), par exemple l'utilisation d'un tableaux proposant des couleurs alternées aux lignes pairs et impairs se fait par l'ajout de la classe **table-striped**. Le même schéma peut être appliqué aux colonnes par ajout de **table-striped-columns**.

Si l'on veut rendre notre tableau un peu plus interactif, il est possible d'utiliser la classe **table-hover** qui modifiera le style des lignes survolées par l'utilisateur. Ce processus peut être fixé via l'utilisation de la classe **table-active** sur des lignes ou des cellules.

Au niveau des bordures, il est possible d'en ajouter via l'utilisation de **table-bordered**. Ceci peut être suivi de **border-<type de couleur>** pour donner une couleur particulière au bordures. Si l'on ne veut absolument pas de bordures, alors la classe **table-borderless** est pour nous !

Pour l'alignement vertical, encore une fois la classe **align-<type>** est disponible. Cette classe est héritée par les enfants du tableaux ou des lignes, il n'est donc pas forcément nécessaire de l'apposer sur chacune des cellules.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

*@ Utopios Consulting*

# **Les Forms**



# Les Bases des Forms

La création de formulaire est grandement simplifiée par l'utilisation de la librairie Bootstrap, qui nous permet d'obtenir rapidement des zones dans notre site web servant à la saisie d'informations utilisateurs comme cela serait le cas dans un formulaire de contact, une zone de login ou le poste d'une annonce sur un site dédié à ce genre de services.

Les classes les plus simples à utiliser sont sans doute **form-label** et **form-control**, qui nous servent à créer facilement une succession d'inputs utilisateurs. Via l'utilisation des attributs d'accessibilité de type aria, il est aussi possible d'y adjoindre une description via la propriété **aria-describedby**. De plus, l'ajout de l'attribut **disabled** va rendre notre input impossible à interagir tout en l'empêchant d'envoyer les données. Cet attribut peut être ajouté au niveau des inputs mais aussi des fieldsets.

Une solution dans la création d'une version inline de ce formulaire pourrait être l'utilisation des **grids** de Bootstrap, par exemple l'utilisation d'un **row** et de multiples **col-auto**

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

Submit

# Les Form Control

Geopios Consulting

Les **Form control** sont pour la plupart des éléments de type input, que l'on style via l'utilisation des classes offertes par la librairie Bootstrap.

Pour commencer, nous avons donc le **form-label** combiné à la classe **form-control** pour les inputs de type texte (**input type="text"** ou **textarea** par exemple) ou listes de données. Une version plus petite ou plus grande du contrôle existe via les classes **form-control-lg** et **form-control-sm**. Pour des inputs de type ColorPicker, on peut utiliser la classe **form-control-color**.



.form-control-lg

Default input

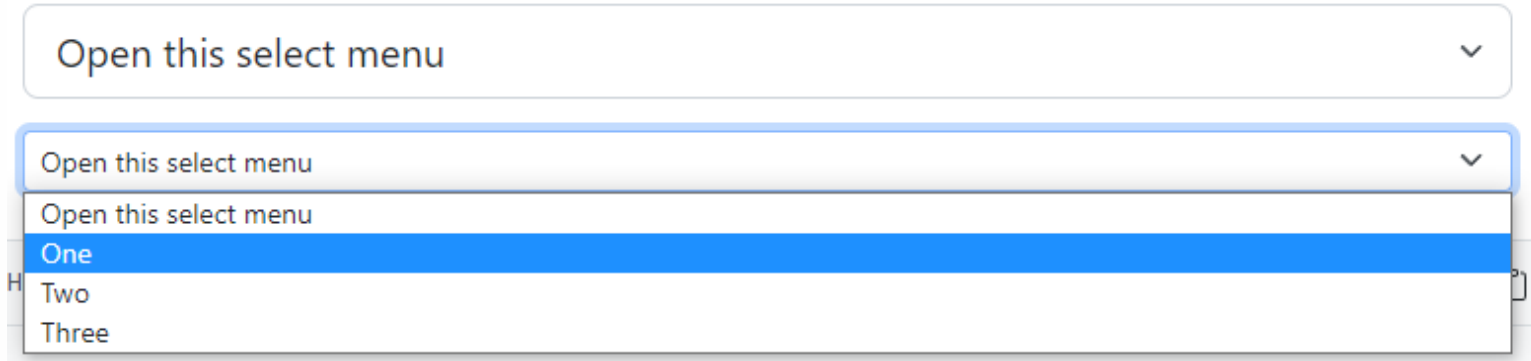
.form-control-sm

Pour créer des inputs de type readonly, il faut penser à ajouter l'attribut **readonly** sur les élément HTML de type **input**. Une variante ne possédant pas de fond existe via la classe **form-control-plaintext**.

# Les Selects

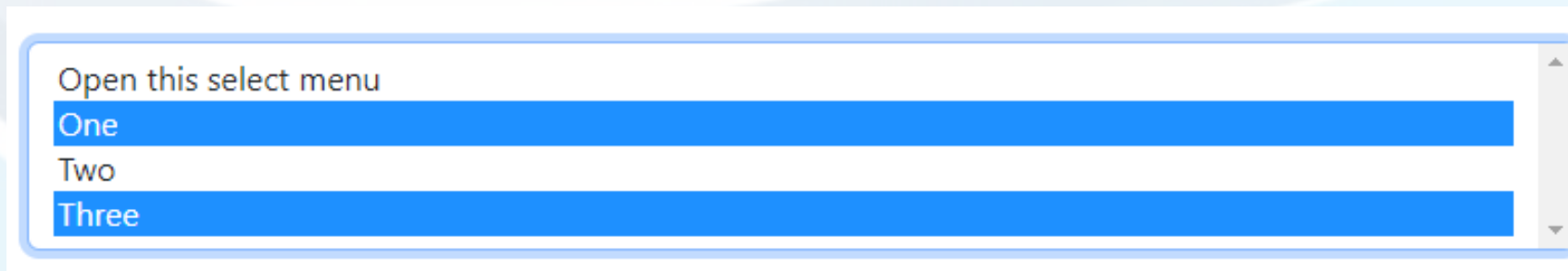
@ Utopios Consulting

Un élément de type select peut être modifié via Bootstrap par les classes **form-select**, **form-select-lg** et **form-select-sm**. Ces trois classes permettent une variation de taille dans les sélections.



Three Bootstrap form-select elements are shown. The top one is the default size. The middle one is the large size (form-select-lg) and is open, displaying a dropdown menu with three options: 'Open this select menu', 'One', 'Two', and 'Three'. The bottom one is the small size (form-select-sm).

Si l'on souhaite avoir plusieurs objets sélectionnables, il nous suffit d'ajouter l'attribut HTML **multiple** à notre balise **<select>**. Si l'on souhaite changer quel élément est sélectionné par défaut, on peut passer par l'attribut HTML **selected**. L'attribut **size=<valeur>** peut servir si l'on veut n'afficher que 3 options à la fois de façon verticale.



A Bootstrap form-select element with the multiple attribute is shown. It displays three options: 'Open this select menu', 'One', 'Two', and 'Three'. The options are displayed vertically, and the 'One' option is selected.

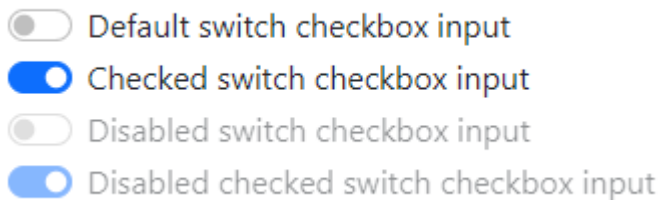
# Les Checks et les Radios

Bootstrap permet l'amélioration du design des check (checkboxes par exemple), les radios, et la création de boutons de switchs ou toggleable.

Pour travailler sur des checkboxes, il suffit de leur donner la classe **form-check-input** combinée avec un label portant la classe **form-check-label**. Si l'on veut que notre checkbox soit cochée par défaut, il suffit de lui donner l'attribut **checked**.

Pour le cas des radios buttons, les classes à utiliser seront les mêmes que pour les checkboxes, mais il ne faudra pas oublier de donner la même valeur à notre propriété **name** dans le but de les relier entre eux, en leur donnant une **id** et une **value** différente.

Si l'on veut réaliser un switch, il nous faut utiliser à la fois les classes **form-check** et **form-switch** sur un élément de type **checkbox**.



Une version inline de notre input est disponible via l'utilisation de la classe **form-check-inline** à ajouter à **form-check**. De même, l'ajout de la classe **btn-check** permet d'obtenir un bouton au fonctionnement de checkbox.

# Les Ranges

@ Utopios Consulting

D'autres entrées utilisateur plutôt utiles sont les Ranges, qui nous permettent de sélectionner une valeur sur une jauge HTML. Pour réaliser ce type d'input, il faut se servir de la classe **form-range** sur un input de type range. Pour spécifier la valeur maximale et minimale de la jauge, il faut utiliser les attributs **min=<valeur>** et **max=<valeur>**. L'ajout de **step=<valeur>** peut être utile si l'on veut agrandir la précision de notre jauge.

Exemple range

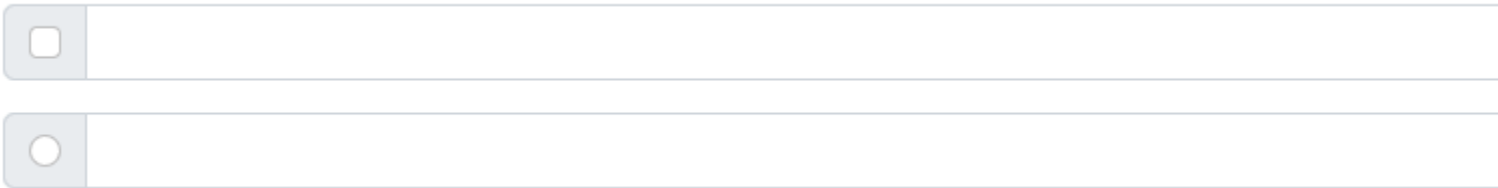


# Les Input Groups

Opios Consulting

Si l'on veut obtenir des regroupement d'éléments HTML pour former des inputs plus détaillés, il faut se servir des **input groups**. Pour le texte, il n'y a rien de plus simple. Il suffit d'ajouter au sein d'une **<div>** portant la classe **input-group** (ou ses variantes **input-group-sm** et **input-group-lg** pour des input groups plus petits ou plus grands) un span portant la classe **input-group-text** ainsi que d'un élément de type input possédant une **aria-describedby=<valeur>** menant à l'Id du span de texte.

Il est également possible de placer un autre élément HTML dans notre **<span>**, par exemple une checkbox, pour créer un input-group rassemblant plusieurs inputs au même endroit comme dans l'exemple ci-dessous :



The image shows two examples of input groups. The first example consists of a checkbox (a small square with a dot inside) followed by a text input field. The second example consists of a radio button (a small circle with a dot inside) followed by a text input field. Both examples are enclosed in a light blue border.

Un autre élément intéressant de la famille des input-group est sans doute le **dropdown**. Ce dernier permet d'avoir accès à une série d'options en appuyant sur un texte ou un bouton. Pour réaliser un tel élément, il faut avoir au sein d'une **<div>** portant la classe **input-group** un bouton possédant la classe **dropdown-toggle** et une autre **<div>** contenant les éléments du menu portant la classe **dropdown-menu**. A cela doivent s'ajouter des propriétés aria telles que **aria-haspopup**, **aria-expanded** et **aria-labelledby**. Cette dernière propriété aria doit cibler le même nom que la propriété **data-bs-toggle** de notre élément portant la classe **dropdown-toggle**

# Forms Avancés

Utopios Consulting

Pour styliser un peu plus nos inputs, il est possible de placer le label dans la case nous permettant d'entrer nos données. Le label se placera automatiquement après redimensionnement au sommet de l'input lorsque l'on cherchera à interagir avec lui. Pour obtenir un tel résultat, il faut se servir de la classe **form-floating** qui contiendra à la fois notre **form-control** et notre **form-label**.

On peut ajouter des classes pour styliser la validations des formulaires via **is-valid** et **is-invalid** qui seraient idéalement ajoutée en Javascript à des formulaires forçant l'ajout de données par leur attribut **required**. Pour retirer la validation du navigateur sur un input, il est aussi possible de lui ajouter l'attribut **novalidate**. Si l'on veut avoir une validation stylisée sur tout le formulaire, il nous faudra utiliser la classe **needs-validation** qui devra être ajoutée au niveau de la balise **<form>**

First name	Last name	Username
<input type="text" value="Mark"/>	<input type="text" value="Otto"/>	<input type="text" value="@"/>
Looks good!	Looks good!	Please choose a username.
City	State	Zip
<input type="text"/>	<input type="text" value="Choose..."/>	<input type="text"/>
Please provide a valid city.	Please select a valid state.	Please provide a valid zip.
<input type="checkbox"/> Agree to terms and conditions You must agree before submitting.		
<input type="button" value="Submit form"/>		