



Bootstrap est le framework HTML, CSS et JavaScript le plus répandu pour le développement de sites Web réactifs axés sur le mobile.  
Responsive web design.

Bootstrap s'adapte du telephone, en passant par la tablette jusqu'à l'ordinateur, RIEN ne lui résiste !! Laissez-vous tenter !!

- Bootstrap est une infrastructure frontale gratuite pour un développement Web plus rapide et plus facile.
- Bootstrap inclut des modèles de conception basés sur HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les modaux, les carrousels d'images et bien d'autres, ainsi que des plugins JavaScript en option.
- Bootstrap vous donne également la possibilité de créer facilement des conceptions réactives.

- **Facile à utiliser:** toute personne possédant des connaissances de base en HTML et CSS peut commencer à utiliser Bootstrap
- **Fonctions adaptatives:** le CSS réactif de Bootstrap s'adapte aux téléphones, aux tablettes et aux ordinateurs de bureau.
- **Approche Mobile-first:** dans Bootstrap 4, les styles Mobile-First font partie du cadre principal.
- **Compatibilité de navigateur:** Bootstrap est compatible avec tous les navigateurs modernes (Chrome, Firefox, Internet Explorer, Edge, Safari et Opera)

Il existe deux façons de commencer à utiliser Bootstrap sur votre propre site Web Vous pouvez Télécharger Bootstrap à partir de [getbootstrap.com](https://getbootstrap.com) ou inclure le Bootstrap à partir d'un CDN : (à mettre juste avant `</body>`)

```
<!-- Latest compiled and minified CSS -->
```

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
```

```
<!-- jQuery library -->
```

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

```
<!-- Popper JS -->
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>
```

```
<!-- Latest compiled JavaScript -->
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
```

➤ Pour assurer un rendu correct et un zoom tactile, ajoutez la balise <meta> suivante à l'intérieur de l'élément <head>:

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

- La partie width = device-width définit la largeur du périphérique.
- L'échelle initiale = 1 définit le niveau de zoom initial lorsque la page est chargée pour la première fois par le navigateur.

Bootstrap nécessite également un élément « contenant » pour envelopper le contenu du site:

Il existe deux classes de conteneurs parmi lesquelles choisir:

- La classe **container** fournit un conteneur sensible à largeur fixe
- La classe **Conteneur-fluid** fournit un conteneur pleine largeur, couvrant toute la largeur de la fenêtre.

# Les Couleurs

# Les couleurs dans Bootstrap

Il existe différentes classes pour mettre en valeur les textes, les liens( en ajoutant une couleur plus sombres au survol) et les bg

<p class="text-muted">texte en gris très pâle

<p class="text-primary">Texte important en bleu ciel

<p class="text-success">Texte en vert

<p class="text-info">bleu ciel pâle

<p class="text-warning">Texte en orangé

<p class="text-danger">Texte en rouge

<p class="text-secondary">Texte en gris pâle(comme muted)

<p class="text-dark">Texte en gris foncé

<p class="text-body">couleur du body par défaut (souvent noir)

<p class="text-light">Texte gris moyen( sur bg blanc)

<p class="text-white">Texte blanc(sur bg foncé)



**Blue:**

#1195c6

**Indigo:**

#604c8d

**Purple:**

#800080

**Pink:**

#e83e8c

**Red:**

#eb3a26

**Orange:**

#f2672e

**Yellow:**

#ffd004

**Green:**

#32b140

**Teal:**

#1ca28a

**Cyan:**

#1702b8

**White:**

#FFFFFF

**Secondary:**

#6a626d

**Dark:**

#101920

# Le système de Grid

# Le système de Grid

## Le système de grille de Bootstrap autorise jusqu'à 12 colonnes sur la page

Si vous ne souhaitez pas utiliser les 12 colonnes individuellement, vous pouvez les regrouper pour créer des colonnes plus larges:

Span 1	Span 2	Span 3	Span 4	Span 5	Span 6	Span 7	Span 8	Span 9	Span 10	Span 11	Span 12
Span 4				Span 4				Span 4			
Span 4				Span 8							
Span 6							Span 6				
Span 12											

Qui dit Grid dit : `<div class= "row"> + <div class = "col">`

```
<div class="row">
```

```
  <div class="col-*-*"></div>
```

```
  <div class="col-*-*"></div>
```

```
</div>
```

Toujours égal à col-12

Faut  
fermer  
les div !!

Si l'on veut, Bootstrap gère seul les colonnes il suffit de créer le nombre de colonnes souhaité, et le framework divise celui-ci selon le nombre demandé on écrit donc juste `"col"` sans préciser quoique ce soit.

Vous pouvez également utiliser `col-sm`, `col-md`, `col-lg`, `col-xl` pour rendre les colonnes responsives.

## Les Grids pour tous les appareils

```
<div class="row">
```

```
  <div class="col-xs-7 col-sm-6 col-lg-8">.col-xs-7 .col-sm-6 .col-lg-8</div>
```

```
  <div class="col-xs-5 col-sm-6 col-lg-4">.col-xs-5 .col-sm-6 .col-lg-4</div>
```

```
</div>
```

```
<div class="row">
```

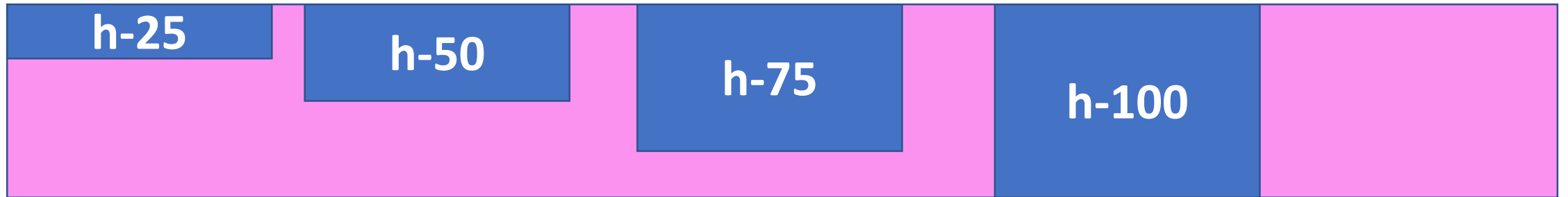
```
  <div class="col-xs-6 col-sm-8 col-lg-10">.col-xs-6 .col-sm-8 .col-lg-10</div>
```

```
  <div class="col-xs-6 col-sm-4 col-lg-2">.col-xs-6 .col-sm-4 .col-lg-2</div>
```

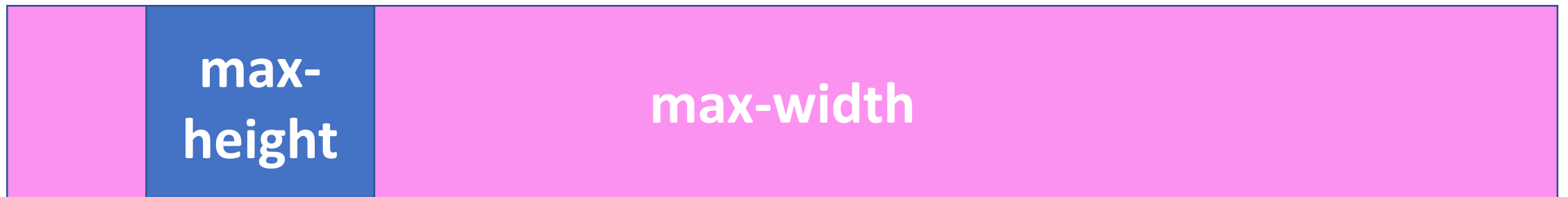
```
</div>
```

**Quelques bases de tailles, et  
de mise en valeur...**

Les hauteurs se définissent avec h-25 (par exemple soit  $\frac{1}{4}$  de la hauteur de la div)



On peut également utiliser max-height ou max-width:



Les largeurs se définissent avec **w-25** (par exemple soit  $\frac{1}{4}$  de la largeur de la div)

W-25

W-50

W-75

W-100



Le **Jumbotron**: permet d'avoir un container prenant toute la largeur de l'écran et une partie suffisamment haute pour présenter un site.

Placez le jumbotron dans le `<div class = "container">` si vous souhaitez que le jumbotron s'étende jusqu'au bord de l'écran:

```
<div class="container">  
  <div class="jumbotron">
```

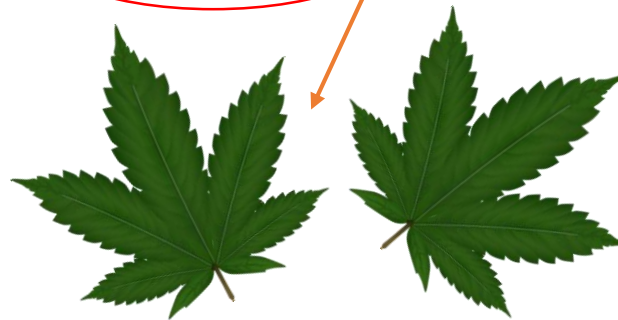
- Un en-tête de page est comme un diviseur de section.  
La classe **page-header** ajoute une ligne horizontale (gris très clair)  
sous l'en-tête  
(+ ajoute un espace supplémentaire autour de l'élément)

là !!

- La classe **well** ajoute une bordure arrondie autour d'un élément avec une couleur de fond gris et un peu de remplissage:

On peut choisir entre 3 tailles d'entourage de well: **small-well**,

**normal-well** ou **large-well** qui s'écrivent **well well-sm;** **well** ou **well well-lg.**



Repeat after me !

# S'adapter au supports

**XS** (pour les téléphones-écrans de moins de 768px de large)

**SM** (pour tablettes-écrans égaux ou supérieurs à 768px de large)

**MD** (pour les petits ordinateurs portables-écrans égaux ou supérieurs à 992px de large)

**LG** (pour les ordinateurs portables et les ordinateurs de bureau-écrans égaux ou supérieurs à 1200px de large)

# Taille des titres et sous-titres

h1 Bootstrap heading (36px)

h2 Bootstrap heading (30px)

h3 Bootstrap heading (24px)

h4 Bootstrap heading (18px)

h5 Bootstrap heading (14px)

h6 Bootstrap heading (12px)

## ➤ Quelques Trucs

- Dans Bootstrap, l'élément HTML `<small>` est utilisé `</small>` pour créer un texte secondaire plus léger dans n'importe quel titre
- L'élément `<mark>` **permet** `</mark>` de mettre en valeur du texte, celui-ci est mis en caractère gras
- L'élément `<abbr>` `</abbr>` est utilisé pour marquer une abréviation ou un acronyme:  
`<p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>`  
Cela donnera:

The WHO was founded in 1948

- l'élément `<BLOCKQUOTE></ BLOCKQUOTE>` est utilisé pour présenter le contenu d'une autre source (référence à un article d'un journal ou autre)

`<BLOCKQUOTE><footer>From WWF's website</footer></ BLOCKQUOTE>`

From WWF's website

Pour le mettre sur la droite il suffit d'appliquer `<BLOCKQUOTE-REVERSE>`

From WWF's website

- Le `dl` permet de faire une liste descriptive

`<dl>`

`<dt>Café</dt>`



Café

`<dd>- chaude</dd>`



- boisson chaude

`<dt>Lait</dt>`



Lait

`<dd>- Boisson froide</dd>`



- Boisson froide

`</dl>`

- Les éléments HTML suivants: `span`, `section` et `div` définissent une section dans un document.
- Utilisez `<kbd> Ctrl + p </kbd>` pour ouvrir la boîte de dialogue « Imprimer ».
- `<pre>` permet de garder les espaces que l'on souhaite ou les passages à la ligne même si la ligne n'est pas terminée `</pre>`
- Les classes de couleurs de texte sont: `text-mute`, `text-primary`, `text-success`, `text-info`, `text-warning`, `text-danger`, `text-secondary`, `text-white`, `text-body`, (couleur du body par défaut, souvent noir), et `text-light`.

- **Small**: Indique un texte plus petit (défini à 85% de la taille du parent)
- **Text-Left**: Indique le texte aligné à gauche
- **Text-Center**: Indique le texte aligné au centre
- **Text-Right**: Indique le texte aligné à droite
- **Text-justifier**: Indique le texte justifié
- **Text-nowrap**: Indique aucun texte d'habillage
- **Text-minuscule**: Indique le texte en minuscules



- **Text-majuscule**: Indique le texte en majuscules
- **Text-capitaliser**: Indique le texte en majuscules
- **List-non stylisés**: Supprime le style de liste par défaut et la marge de gauche sur les éléments de liste (fonctionne sur <ul> et <ol>). Cette classe s'applique uniquement aux éléments de liste enfants immédiats (pour supprimer le style de liste par défaut de toutes les listes imbriquées, appliquer cette classe à toutes les listes imbriquées ainsi)
- **List-Inline**: Place tous les éléments de liste sur une seule ligne

# Les Tables Bootstrap

# Les tables Bootstrap

Une table Bootstrap basique a seulement des séparateurs horizontaux.

Table  
table!..  
..on en  
reparle

En ajoutant la class **table table-striped** une ligne sur 2 sera colorée

Et avec **table table-bordered**, le tableau aura des bordures de tous les côtés.

**Table table-hover** permet une couleur grisée de la surface au survol

Feuille aussi....

Les classes contextuelles peuvent être utilisées pour colorier les lignes de table (<tr>) ou les cellules de tableau (<td>) avec les mêmes couleurs que pour les titres ex: <tr class = "danger">



- **Actif:** Applique la couleur de pointage à la ligne de tableau ou à la cellule de tableau

**le succès:** Indique une action réussie ou positive(success)



- **info:** Indique une modification ou une action informative neutre



- **AVERTISSEMENT:** Indique un avertissement qui peut nécessiter une attention(warning)



- **danger:** Indique une action dangereuse ou potentiellement négative



## Les tables

- La classe **table-Responsive** crée une table réactive. La table défilera ensuite horizontalement sur de petits appareils (sous 768px). Lors de la visualisation sur quelque chose de plus grand que 768px de large, il n'y a aucune différence
- Vous pouvez également utiliser le système de grille de Bootstrap en conjonction avec la classe **Thumbnail** pour créer une galerie d'images.
- Les images sont disponibles dans toutes les tailles. Les écrans aussi. Les images réactives s'ajustent automatiquement pour s'adapter à la taille de l'écran.
- Créez des images sensibles en ajoutant une classe. **img-responsive** à la balise `<img>`. L'image sera alors bien à l'échelle de l'élément parent.

## Création des titres de chaque colonne

```
<table class="table">
```

```
<tr>
```

```
<thead>
```

```
<th>Nom</th>
```

```
<th>Jour</th>
```

```
<th>Évènement</th>
```

```
</thead>
```

```
</tr>
```

Création des lignes de titres

Création 1ère colonne

Création 2ème colonne

Création 3ème colonne

Fermeture des titres

## Contenu de la table

<tbody>

<tr>

<td>Erwann</td>

<td>Lundi et Mardi!</td>

<td>Le cousin de Gaston!!!! </td>

</tr>

<tr>

<td>Fred</td>

<td> Mercredi et jeudi<td>

<td>@les blagues à papa !</td>

</tr>

<tr>

<td>Fabrice</td>

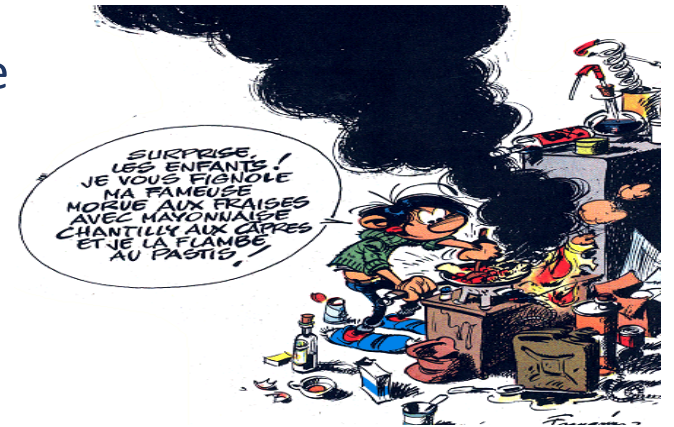
<td>Vendredi...fous rires !</td>

<td>@plus tard tu comprendras ...!</td>

</tr>

</tbody>

1<sup>ère</sup> colonne



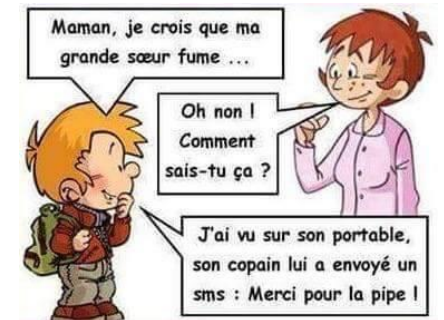
2<sup>ème</sup> colonne

Humour spécial  
pour #Nora...



3<sup>ème</sup> colonne

Les débats en tous  
genres du vendredi...



# Les Images



## Formes et images

































- La classe `img-Responsive` applique `Display: Block;` et `max-width: 100%;` et la hauteur: `auto;` à l'image
- Laissez également les vidéos ou les diaporamas évoluer correctement sur n'importe quel appareil.
- Les classes peuvent être appliquées directement aux éléments `<iframe>`, `<embed>`, `<video>` et `<object>`.

**Les Classes (fr)Class(En)**  
**(à répéter 2 fois car**  
**Bootstrap est dur de la**  
**feuille...)**

Il existe de très nombreuses classes, permettant une multitude de possibilités de créations et d'actions, elles sont toutes répertoriées dans la **doc Bootstrap 4** dans la rubrique: Bootstrap4 REF: All Classes

En voici quelques-unes:

- **m-** pour les classes **margin**
- **p** – pour les classes **padding**
- **t** – pour les classes **margin-top** ou **padding-top**
- **l** – pour les classes **margin-left** ou **padding-left**
- **r** – pour les classes **margin-right** ou **padding-right**
- **x** – pour les classes **both \*-left** et **\*-right**
- **y** – pour les classes **both \*-top** et **\*-bottom**
- **blank** – pour les classes **margin** ou **padding** sur tous les côtés de l'élément

- **Aligner un texte:** text-align-left ou right ou center ou text-justify
- **Changer la casse:** text-lowercase,(a,b,c) text-uppercase(A,B,C) ou text-capitalize(Ann-Katell 😊)
- **Les différents Font-weight:**
  - **Bold text**
  - **font-weight-bolder** (relative to the parent element).
  - Normal weight text
  - Light weight text
  - Lighter weight text (relatif à l'élément parent).
  - *font-italic*

Toujours dans les classes:

Vous pouvez également ajouter une opacité de 50%:

pour le texte noir: **text-black-50**

ou blanc avec les classes: **text-white-50**

- Dans les classes, on peut entourer du texte avec des couleurs d'alerte: `alert-info(bleu)`, `alert-warning(orangé)`, `alert-danger(rouge)` ou `alert-success(vert)`.

Ex: `<div class="alert alert-success">`

Vraiment très dur de la feuille !

Code JavaScript pour fermer une alert

```
$('.close').alert("close");
```

- Cela fonctionne de la même manière pour les links:

```
<div class="alert alert-success">
```

```
  <strong>Success!</strong> Vous devriez<a href="#" class="alert-link">lire ce message !</a>.
```

```
</div>
```



Les boutons :`btn btn(gris)`, `btn btn-default(blanc)`, `btn btn-primary(bleu foncé)`, `btn btn-success(vert)`, `btn btn-info(bleu ciel)`, `btn btn-warning(orangé)`, `btn btn-danger(rouge)`, et `btn btn-link` pour les liens.

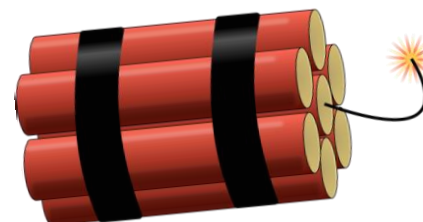
➤ Pour les couleurs de texte: ajouter `text-couleur choisie`



➤ Pour le background: `bg-couleur` (Attention Fred il aime pas quand les Bg sont dans le HTML !!!! Donc dans votre intérêt, oubliez ce que je viens d'écrire...)



➤ Pour un dégradé de background: `bg-gradient-couleur choisie`(idem que ci-dessus... je crois qu'il faut oublier....



# Les Boutons

➤ Les class boutons sont à utiliser de cette manière:

```
<button type="button" class="btn">Basic</button>  
<button type="button" class="btn btn-default">Default</button>  
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-info">Info</button>  
<button type="button" class="btn btn-warning">Warning</button>  
<button type="button" class="btn btn-danger">Danger</button>  
<button type="button" class="btn btn-link">Link</button>
```

**Toujours répéter 2 fois la class pour que Bootstrap comprenne !!!! Vous avez compris !!!  
Il est dur de la feuille !!!**



- Les classes de boutons peuvent être utilisées sur les éléments :<a>, <button> ou <input> :

```
<a href="#" class="btn btn-info" role="button">Link Button</a>
```

```
<button type="button" class="btn btn-info">Button</button>
```

```
<input type="button" class="btn btn-info" value="Input Button">
```

```
<input type="submit" class="btn btn-info" value="Submit Button">
```

# Les tailles des boutons:

comme toujours dans Bootstrap, selon ce que l'on souhaite comme taille btn-xs, ou sm ou md ou lg

```
<button type="button" class="btn btn-primary btn-lg">Large</button>  
<button type="button" class="btn btn-primary btn-md">Medium</button>  
<button type="button" class="btn btn-primary btn-sm">Small</button>  
<button type="button" class="btn btn-primary btn-xs">XSmall</button>
```

Si l'on souhaite que le bouton prenne la largeur entière de son parent on utilise **btn-block**:

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

- La classe active fait apparaître un bouton (et paf ! Un bouton!) et la classe disabled rend un bouton non cliquable( un bouton qu'on peut pas cliquer en même tps.... C'est nul !!!! C'est plus un bouton!)

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

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

- On peut aussi grouper plusieurs boutons avec **btn-group**:  
Et appliquer les tailles que l'on souhaite(xs, sm md ou lg)

```
<div class="btn-group btn-group-lg">
```

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

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

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

```
</div>
```

Fabrice

Fred

Erwann

- Il est également possible de les mettre à la verticale:

```
<div class="btn-group-vertical">  
  <button type="button" class="btn btn-primary">Mopi</button>  
  <button type="button" class="btn btn-primary">Ubuntu</button>  
  <button type="button" class="btn btn-primary">BB8</button>  
</div>
```



Code JavaScript :\$('.btn').button();

- ou qu'ils prennent toute la largeur avec **justified**:

```
<div class="btn-group btn-group-justified">  
  <a href="#" class="btn btn-primary">Quentin</a>  
  <a href="#" class="btn btn-primary">Voyages</a>  
  <a href="#" class="btn btn-primary">Photos</a>  
</div>
```

Quentin

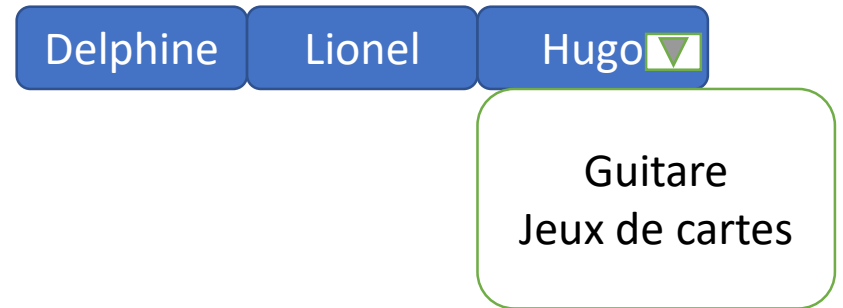
Voyages

Photos



➤ Créer des boutons regroupés avec un menu déroulant :

```
<div class="btn-group">  
  <button type="button" class="btn btn-primary">Delphine</button>  
  <button type="button" class="btn btn-primary">Lionel</button>  
  <div class="btn-group">  
    <button type="button" class="btn btn-primary dropdown-toggle" data-  
toggle="dropdown">  
      Hugo <span class="caret"></span></button>  
    <ul class="dropdown-menu" role="menu">  
      <li><a href="#">Guitare</a></li>  
      <li><a href="#">Jeux de cartes</a></li>  
    </ul>  
  </div>  
</div>
```



Avec simple collapsible je fait quoi ???

Tu caches le texte tant qu'il n'est pas cliqué ....

Et des fois c'est cool... imagine...

Tu as un bouton... il va,

quand tu cliqueras dessus, ouvrir des infos... tu veux les voir ???

T'es sûr(e) ?

Déconne pas j'ai encore plein de pages à te faire découvrir....

Donc t'as dit oui !!! Youpi !!! (à tes risques et périls...)

➤ Cliquez sur un bouton pour voir un texte caché:

```
<button type="button" class="btn btn-info" data-toggle="collapse" data-target="#demo">Simple collapsible</button>
```

```
<div id="demo" class="collapse">
```

Hono, c'est la sagesse même, le calme et la plénitude dans une promo, il apaise et est si calme que tout le monde lui envie sa zénitude...

```
</div>
```

Simple collapsible

au clic le texte: **Hono, c'est la sagesse même, le calme et la plénitude dans une promo...** apparaîtra.

Il est possible d'en mettre plusieurs(**pas Hono!**) les uns sous les autres avec **Accordion**.

**Code Javascript**

```
$('.collapse').collapse()
```

# Les glyphicons

- Bootstrap fournit 260 !!!! glyphicons de l'ensemble **Glyphicons Halflings**: (Bootstrap Glyphicon reference)

Les glyphicons peuvent être utilisés dans du texte, des boutons, des barres d'outils, la navigation, des formulaires, etc.

<p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>



<p>Envelope icon as a **link**:  
<a href="#"><span class="glyphicon glyphicon-envelope"></span></a>  
</p>



# Progress bar, Pagination

- Il est possible d'intégrer des « progress bar » de la couleur de notre choix ( info, danger, success, warning, danger) comme dab quoi...

```
<div class="progress">
```

```
  <div class="progress-bar progress-bar-info" role="progressbar" aria-  
valuenow="50"
```

```
  aria-valuemin="0" aria-valuemax="100" style="width:50%">
```

```
    50% Complete (info)
```

```
  </div>
```

```
</div>
```



- Pour quelle soit rayée on ajoute : progress-bar-striped
- Et pour l'animer on ajoute en plus active.



- Lorsque l'on a plusieurs pages sur notre site, nous pouvons montrer à l'utilisateur où il se trouve grâce à la **pagination active** (des fois qu'il soit paumé et qu'il n'ait pas de GPS...)

```
<ul class="pagination">  
  <li><a href="#">1</a></li>  
  <li class="active"><a href="#">2</a></li>  
  <li><a href="#">3</a></li>  
  <li><a href="#">4</a></li>  
  <li><a href="#">5</a></li>  
</ul>
```

1	2	3	4	5
---	---	---	---	---

(On peut aussi y ajouter les tailles sm, md, lg et xl)(c'est redondant je sais !)



- Pour créer des boutons précédents / suivants, ajoutez la classe pager à un élément <ul>:
- ```
<ul class="pager">  
  <li class="previous"><a href="#">Précédent</a></li>  
  <li class="next"><a href="#">Suivant</a></li>  
</ul>
```

Précédent

Suivant

➤ Pour créer un liste colorée:

Pfffffffff !!!!



```
<ul class="list-group">  
  <li class="list-group-item list-group-item-success">First item</li>  
  <li class="list-group-item list-group-item-info">Second item</li>  
  <li class="list-group-item list-group-item-warning">Third item</li>  
  <li class="list-group-item list-group-item-danger">Fourth item</li>  
</ul>
```

success

info

warning

Danger

# Listes avec dropdown

Ne pas oublier ton pote JavaScript pour animer...

Cf: doc W3schools

➤ **Menu avec liste et dropdown:**(pour les mettre sur une même ligne:display:inline-block)

```
<ul class="nav nav-tabs list-inline">
```

```
<li class="active"><a href="#">Home</a></li>
```

```
<li class="dropdown">
```

```
<a class="dropdown-toggle" data-toggle="dropdown" href="#">Menu 1
```

```
<span class="caret"></span></a>
```

```
<ul class="dropdown-menu">
```

```
<li><a href="#">Submenu 1-1</a></li>
```

```
<li><a href="#">Submenu 1-2</a></li>
```

```
<li><a href="#">Submenu 1-3</a></li>
```

```
</ul>
```

```
</li>
```

```
<li><a href="#">Menu 2</a></li>
```

```
<li><a href="#">Menu 3</a></li>
```

```
</ul>
```

**Navbar**

➤ Pour centrer les tabs du menus il faut ajouter **nav-justified**  
<ul class="nav nav-tabs nav-justified">

La nav-bar par défaut est grise si l'on veut la version noire il faut utiliser **inverse**

```
<nav class="navbar navbar-inverse">  
<div class="container-fluid">
```

```
  <div class="navbar-header">
```

```
    <a class="navbar-brand" href="#">Nom du site</a></div>
```

```
    <ul class="nav navbar-nav">
```

```
      <li class="active"><a href="#">Home</a></li>
```

```
      <li><a href="#">Page 1</a></li>
```

```
      <li><a href="#">Page 2</a></li>
```

```
      <li><a href="#">Page 3</a></li>
```

```
    </ul>
```

```
</div></nav>
```

➤ **Pour la mettre à droite:**

```
<ul class="nav navbar-nav navbar-right">
```

➤ **Ajouter une couleur:**

```
<button class="btn btn-danger navbar-btn">Button</button>
```

➤ **Ajouter un lien:**

```
<li><a href="#">Link</a></li>
```

➤ **Fixer la nav bar** au top de la page pour qu'elle reste toujours à sa place même si le reste de la page est en scroll:

```
<nav class="navbar navbar-inverse navbar-fixed-top">
```

Ou à l'inverse, la fixer au **bottom** à la place de "top"

Avec **collapse** les **links** seront cachés et s'ouvriront au clic de la souris

Sur : →



## Code Javascript à ajouter pour faire fonctionner le tout

```
<script>
$(document).ready(function(){
  $(".nav-tabs a").click(function(){
    $(this).tab('show');
  });
});
</script>
```



# Les formulaires

- Dans un formulaire en ligne, tous les éléments sont en ligne, alignés à gauche et les étiquettes sont à côté, Ajouter la classe **form-inline** à l'élément **<form>**

```
<form class="form-inline" action="/action_page.php">
  <div class="form-group">
    <label for="mail"> adresse mail:</label>
    <input type="email" class="form-control" id="mail"></div>
    <div class="form-group">
      <label for="pwd">Mot de passe:</label>
      <input type="password" class="form-control" id="pwd">
    </div><div class="checkbox"><label><input type="checkbox">
      Ne m'oubliez pas !</label>
    </div>
    <button type="submit" class="btn btn-default">Envoyer</button>
</form>
```

# Les checkbox

### ➤ Les checkbox:

```
<label class="checkbox-inline"><input type="checkbox" value="">Option  
1</label>
```

```
<label class="checkbox-inline"><input type="checkbox" value="">Option  
2</label>
```

☐ option 1   ☐ option 2

### ➤ Les radios buttons:

```
<div class="radio">
```

```
  <label><input type="radio" name="optradio" checked>Option 1</label>  
</div>
```

```
<div class="radio">
```

```
  <label><input type="radio" name="optradio">Option 2</label>  
</div>
```

☒ option 1   ☐ option 2

Le Carrousel  
En français avec 2 R,  
en Anglais 1 R...

- Les carrousels nécessitent l'utilisation d'un identifiant (`<div id = "myCarousel " >`) pour que les commandes du carousel fonctionnent correctement.
- La `class = "carousel"` spécifie que ce `<div>` contient un carousel.
- La **class slide** ajoute une transition CSS et un effet d'animation, ce qui fait glisser les éléments lors de l'affichage d'un nouvel élément. Omettez cette classe si vous ne voulez pas cet effet.
- **L'attribut data-ride = "carousel"** indique à Bootstrap de commencer à animer le carousel dès le chargement de la page.

- **La partie "Indicateurs"**: Les indicateurs sont les petits points au bas de chaque diapositive (qui indiquent le nombre de diapositives présentes dans le carrousel et la diapositive actuellement visualisée par l'utilisateur).
- Les indicateurs sont spécifiés dans une liste ordonnée avec des indicateurs de class de carrousel.
- **L'attribut data-target** pointe sur **l'id** du carrousel.
- **L'attribut data-slide-to** spécifie la diapositive à utiliser pour cliquer sur le point spécifique.

- La class **active** doit être ajoutée à l'une des diapositives. Sinon, le carrousel ne sera pas visible.

```
<div id="myCarousel" class="carousel slide" data-ride="carousel">  
  <!-- Indicators -->  
  <ol class="carousel-indicators">  
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>  
    <li data-target="#myCarousel" data-slide-to="1"></li>  
    <li data-target="#myCarousel" data-slide-to="2"></li>  
  </ol>
```



➤ La partie "Wrapper for slides":

Les diapositives sont spécifiées dans un `<div>` avec la class `carousel-inner`.

➤ Le contenu de chaque diapositive est défini dans un `<div>` avec `class item`. Cela peut être du texte ou des images.

```
<!-- Wrapper for slides -->
```

```
<div class="carousel-inner">
```

```
<div class="item active">
```

```

```

```
</div>
```

```
</div>
```



➤ La partie "**Commandes gauche et droite**":

Ce code ajoute des boutons "gauche" et "droit" qui permettent à l'utilisateur de naviguer manuellement entre les diapositives.

```
<!-- Left and right controls -->
```

```
<a class="left carousel-control" href="#myCarousel" data-slide="prev">
```

```
<span class="glyphicon glyphicon-chevron-left"></span>
```

```
<span class="sr-only">Previous</span>
```

```
</a>
```

```
<a class="right carousel-control" href="#myCarousel" data-slide="next">
```

```
<span class="glyphicon glyphicon-chevron-right"></span>
```

```
<span class="sr-only">Next</span>
```

```
</a>
```

```
</div>
```

➤ L'attribut **data slide** accepte les mots-clés "prev" ou "next", qui correspondent à la position par rapport à la position actuelle.

➤ Ajouter des légendes aux diapositives Ajoutez **<div class = "carrousel-caption">** dans chaque **<div class = "item">** pour créer une légende pour chaque diapositive:

La Mer

```
<!-- Wrapper for slides -->
<div class="carousel-inner">
  <div class="item active">
    
    <div class="carousel-caption">
      <h3>La Mer</h3>
      <p>Vive les Breizhous!!>
    </div>
  </div>
</div>
```



Vive les Breizhous !!

## Activer le Carrousel

```
$("#myCarousel").carousel();
```

## Activer les indicateurs du carrousel

```
$(".item").click(function(){  
    $("#myCarousel").carousel(1);  
});
```

## Activer les controls du carrousel

```
$(".carousel-control-prev").click(function(){  
    $("#myCarousel").carousel("prev");  
});
```

# Les Modals

# Les modals

- Pour déclencher la fenêtre modale:  
Il faut utiliser un **bouton ou un lien**. Ensuite, inclure les deux attributs:  
**data- \*: data-toggle = "modal"** ouvre la fenêtre modale  
**data-target = "# myModal"** pointe vers l'id du modal  
`<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#exampleModal"> demo</button>`
- La partie "modale": Le parent `<div>` du modal doit avoir un ID identique à la valeur de l'attribut.  
data-target utilisé pour déclencher le modal ("myModal").
- La **class modal** identifie le contenu ou `<div>` en tant que modal et le met en évidence.

- La **class fade** ajoute un effet de transition qui **active et désactive le modal**. Supprimez cette class si vous ne souhaitez pas cet effet.
- L'attribut rôle = "**dialog**" améliore l'accessibilité pour les personnes utilisant des lecteurs d'écran.
- La **class modal-dialog** définit **la largeur et la marge** appropriées du modal.
- La partie "**Contenu modal**": Le `<div>` avec `class = "modal-content"` **donne un style au modal** (bordure, couleur de fond, etc.).
- À l'intérieur de ce `<div>`, ajouter l'en-tête, le corps et le pied de page du modal.

- La classe d'en-tête modal est utilisée pour définir le style de l'en-tête modal.
- Le bouton à l'intérieur de l'en-tête a un attribut **data-rejet** = "modal" qui **ferme le modal** si l'on clique dessus.
- La classe **close** met en forme le **bouton de fermeture** et la classe **modal-title** donne un **style** à l'en-tête avec une hauteur de ligne appropriée.
- La classe **modal-body** est utilisée pour définir le style du corps du modal.



- La classe modal-footer est utilisée pour définir le style du pied de page du modal.

(A noter que cette zone est alignée à droite par défaut.)

On peut changer la taille d'un modal dans modal-dialog avec modal-sm, modal-lg....(ça faisait longtemps!!!)

Il existe différents modèles de modals (cf: doc Bootstrap)

A mettre dans un <script> dans le « head »

```
<script>
```

```
$('#myModal').on('shown.bs.modal', function () {  
  $('#myInput').trigger('focus') }) </script>
```

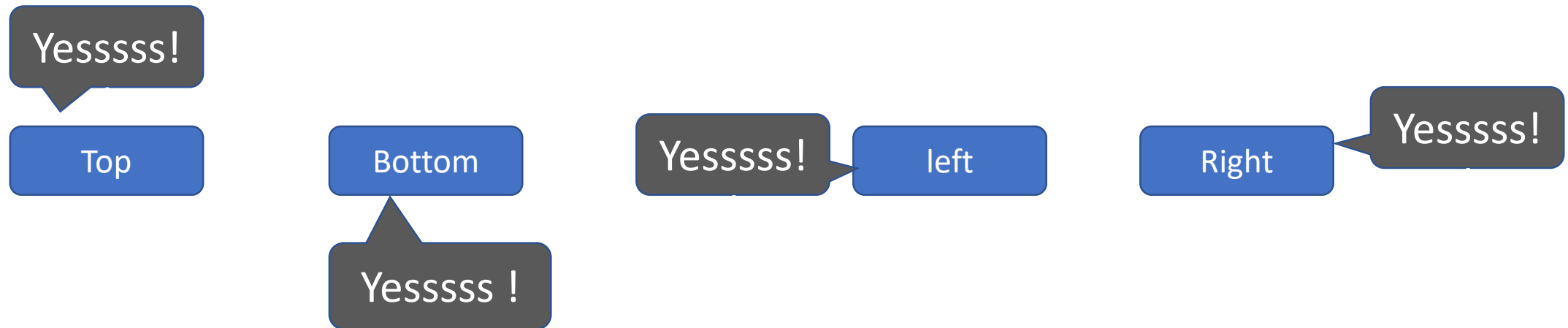
```
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel"
aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        blablabla, les vacances c'est fini.....au boulot !
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div> (je laisse ce code pour ceux qui éventuellement voudraient le dossier en pptx)
```

# Les Tooltips et Popovers

# Les Tooltips

(petits commentaires qui s'ouvrent au clic : top, bottom, right ou left)

```
<a href="#" data-toggle="tooltip" data-placement="top" title="Yesssss!">Hover</a>
```



Les **Popovers** sont comme les Tooltips, mais peuvent contenir plus de choses

- Pour créer un popover, ajoutez l'attribut **data-toggle = "popover"** à un élément.
- Utilisez l'attribut title pour le texte d'en-tête du popover et utilisez le contenu de données dans le corps du popover:  
Il faut initialiser un **code JavaScript** pour qu'il fonctionne:

```
<script>
```

```
 $('[data-toggle="tooltip"]').tooltip(); ➡ (s'adapte à tous les toggles du doc) ou
```

```
 $('#myTooltip').tooltip(); ➡ s'adapte à un élément spécifié
```

```
</script>
```

- Par défaut, le popover apparaîtra à droite de l'élément.  
Utilisez l'attribut **data-placement** pour positionner le popover en haut, en bas, à gauche ou à droite de l'élément:

```
<a href="#" title="Header" data-toggle="popover" data-placement="top" data-content="Content">Click</a>
```

- Par défaut le popover se referme en cliquant dessus, mais on peut ajouter la fonction **data-trigger-focus** qui permettra de cliquer n'importe où sur la page pour le refermer:

```
<a href="#" title="Dismissible popover" data-toggle="popover" data-trigger="focus" data-content="Cliquez où vous voulez pour fermer">Cliquez</a>
```

**Le Scrollspy**

## Le scrollspy:

- Requier un positionnement relatif: l'élément `data-spy = "scroll"` requiert la propriété de position CSS, avec une valeur "relative" pour fonctionner correctement.
- Ajoutez `data-spy = "scroll"` à l'élément à utiliser comme zone de défilement (il s'agit souvent de l'élément `<body>`).
- Ajoutez ensuite l'attribut `data-target` avec une valeur de l'id ou le nom de la classe de la barre de navigation (`.navbar`).

(Cela permet de s'assurer que la barre de navigation est connectée à la zone de défilement).



- Notez que les éléments défilants doivent correspondre à l'intérieur de la liste des éléments de la barre de navigation:

(`<div id = "section1">` correspond à `<a href="#section1">`).

- L'attribut facultatif `data-offset` spécifie le nombre de pixels à compenser lors du calcul de la position du défilement.

```
<body data-spy="scroll" data-target="#myScrollspy" data-offset="20">
```

```
<div class="container">
```

```
<div class="row">
```

```
<nav class="col-sm-3" id="myScrollspy">
```

```
<ul class="nav nav-pills nav-stacked">
```

```
<li><a href="#section1">Section 1</a></li>
```

```
...
```

```
</ul>
```

```
</nav>
```

Il est possible de **fixer une partie** du site(navbar) pour **qu'il ne scroll pas**:

```
<nav class="navbar navbar-inverse" data-spy="affix" data-offset-top="197">
```

- Déplacez les colonnes vers la droite en utilisant les classes `.col-md offset *`.

Ces classes augmentent la marge des colonnes par colonnes.

Code Javascript pour le fonctionnement

```
$('body').scrollspy({target: ".navbar"})
```

# Les Borders

# Les Borders

```
<span class="border"></span>
```

```
<span class="border-top"></span>
```

```
<span class="border-right"></span>
```

```
<span class="border-bottom"></span>
```

```
<span class="border-left"></span>
```

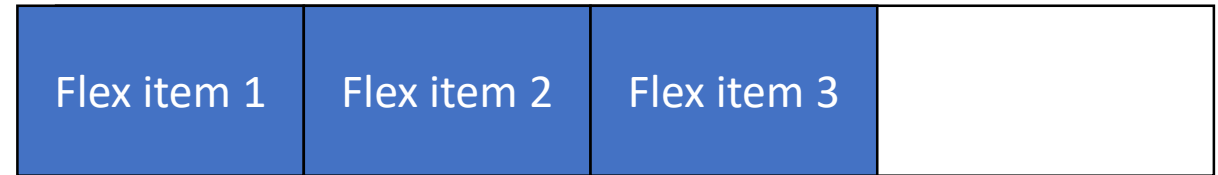
- **Enlever une bordure:** Ajouter -0 derrière border
- **Ajouter une couleur:** mettre primary, secondary, success, danger, warning, info, light, dark, ou white. (ça recommence !!!!)

# Le positionnement

# Direction

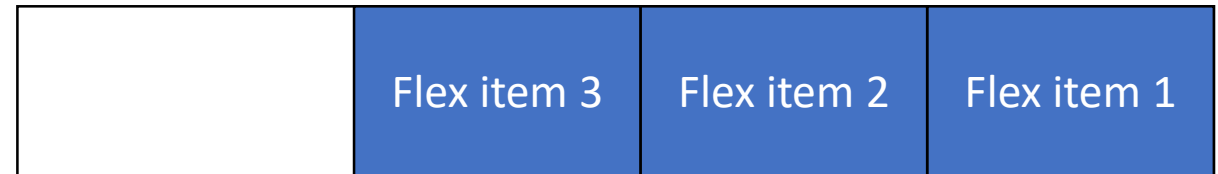
## Flex-row:

```
<div class="d-flex flex-row bd-highlight mb-3">  
  <div class="p-2 bd-highlight">Flex item 1</div>  
  <div class="p-2 bd-highlight">Flex item 2</div>  
  <div class="p-2 bd-highlight">Flex item 3</div>  
</div>
```



## Flex-row-reverse:

```
<div class="d-flex flex-row-reverse bd-highlight">  
  <div class="p-2 bd-highlight">Flex item 1</div>  
  <div class="p-2 bd-highlight">Flex item 2</div>  
  <div class="p-2 bd-highlight">Flex item 3</div>  
</div>
```



- On peut faire de même avec la disposition en colonnes avec **flex-column** ou **flex-column-reverse**

(on peut bien sûr adapter avec xs, sm md et lg à mettre après flex)

- Avec **justified-content** on choisi l'emplacement de nos colonnes ou rows  
start, end, center, between ou around

`<div class="d-flex justify-content-start">`

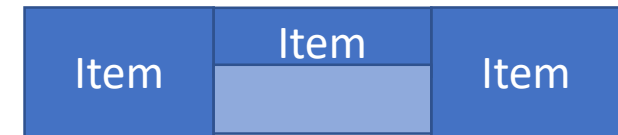
(on adapte également avec xs, sm, md et lg après content)

- **Align-items** : fonctionne de la même manière avec  
d-flex align-items-start ou align-items-end, align-items-center, align-items-baseline, align-items-stretch.

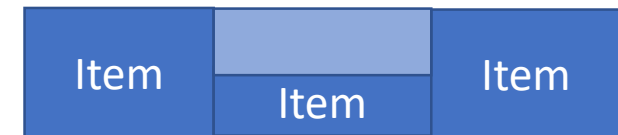


Utilisez les utilitaires **align-self** sur les éléments de la boîte souple pour modifier individuellement leur alignement sur l'axe transversal (l'axe des y pour commencer, axe des x si flex-direction: colonne)

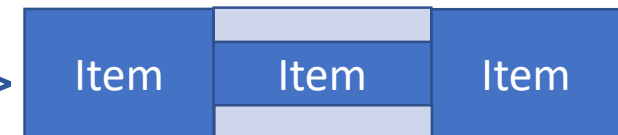
`<div class="align-self-start">Aligned flex item</div>`



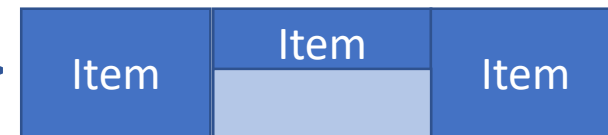
`<div class="align-self-end">Aligned flex item</div>`



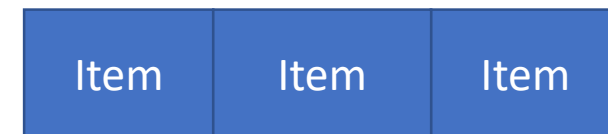
`<div class="align-self-center">Aligned flex item</div>`



`<div class="align-self-baseline">Aligned flex item</div>`



`<div class="align-self-stretch">Aligned flex item</div>`



➤ **Flex-fill** permet de s'adapter au contenu

`<div class="d-flex bd-highlight">`

`<div class="p-2 flex-fill bd-highlight">Flex item avec beaucoup de contenu</div>`

`<div class="p-2 flex-fill bd-highlight">Flex item</div>`

`</div>`



- Modifiez la manière dont les éléments flexibles s'emballent dans un conteneur flexible. Choisissez entre: pas d'habillage (valeur par défaut du navigateur) avec **flex-nowrap**, d'habillage avec **flex-wrap** ou inversé avec **flex-wrap-reverse**. (à adapter avec sm, md, lg et xl)
- **Ordonner les Items comme on le souhaite:** (de 0 à 12 et avec sm md, lg et xl)

```
<div class="d-flex flex-nowrap bd-highlight">  
  <div class="order-3 p-2 bd-highlight">First flex item</div>  
  <div class="order-2 p-2 bd-highlight">Second flex item</div>  
  <div class="order-1 p-2 bd-highlight">Third flex item</div>  
</div>
```

➤ Utilisez les utilitaires **align-content** sur les conteneurs de boîtes flexibles pour aligner les éléments flexibles sur l'axe transversal avec start, center, between, around ou stretch

➤ **Float:**

`<div class="float-left">`à gauche sur tous les écrans`</div>`

`<div class="float-right">`à droite sur tous les écrans`</div>` `<div`

`class="float-none">`Ne flotte pas, donc coule(oui, je sais Nora...je sors bientôt !)`</div>`

# Les Ombres

# Les ombres

```
<div class="shadow-none p-3 mb-5 bg-light rounded">No shadow</div>
```

(Margin bottom)

```
<div class="shadow-sm p-3 mb-5 bg-white rounded">Small shadow</div>
```

`<div class="shadow p-3 mb-5 bg-white rounded">Regular shadow</div>`

```
<div class="shadow-lg p-3 mb-5 bg-white rounded">Larger shadow</div>
```

Je n'ai pas tout dit... je sais c'était long... mais difficile de ne pas avoir envie de parler de tout tant il y a à dire sur le sujet....

Dans ma prochaine édition, je vous parlerai de Vuejs....et son pote Laravel Autant dire qu'il me faudra au moins 400 pages...

N'oubliez pas de réserver vos places !et merci de m'avoir supportée !