

ERGONOMIE ET NAVIGABILITÉ

Utilisation du Framework CSS Bootstrap 5

Définition d'un Framework

Composants de base utiles pour le développement

- Faciliter la programmation à l'aide de briques de base
→ plus rapide, cross-browser (compatible avec tous navigateurs)
- Faciliter la maintenance
→ tests exhaustifs, robustesse du code

Plusieurs types de Framework Web (PHP, JS, Python...) :

➤ **Côté serveur : back-end**

CodeIgniter, Laravel, Symfony – pour le PHP, Django - Python

➤ **Côté client : front-end**

Bootstrap , Foundation, Materialize, Tailwind, Pure, Bulma ou encore KNACSS – pour le CSS

Angular, ReactJS, Vue.JS - pour le JavaScript

Présentation de Bootstrap

Développé par Twitter en 2010, et placé sous licence open source en 2011, **Bootstrap** c'est :

- un framework CSS
 - ➔ mise en forme des pages Web (organisation, aspects, animation...)
- des composants HTML et JavaScript
- un système de grille facilitant la mise en forme
 - ➔ mise en page basée sur une grille de 12 colonnes

Code Web Responsive évolutif

(nouvelles versions régulières, communauté d'utilisateurs)

➔ Doc. : <https://runebook.dev/fr/docs/bootstrap/-index->

Installation de Bootstrap

<https://getbootstrap.com/>

➔ Téléchargement de ***bootstrap-5.3.2-dist.zip***

- ✓ **bootstrap-grid.css** : grille et flex
- ✓ **bootstrap.reboot.css** : réinitialisation des styles (indépendant des navigateurs)
- ✓ **bootstrap.css** : styles complets
- ✓ **bootstrap.js** : codes JavaScript des composants
- ✓ **bootstrap.bundle.js** : codes JS complets incluant popper.js

```
bootstrap/  
├── css/  
│   ├── bootstrap-grid.css  
│   ├── bootstrap.reboot.css  
│   ├── bootstrap.utilities.css  
│   └── bootstrap.css  
└── js/  
    ├── bootstrap.bundle.js  
    └── bootstrap.js
```

Fichier.min : code compressé des fichiers, à utiliser pour un site en ligne

➔ Lien vers un CDN (Content Delivery Network /réseau de distribution de contenu).

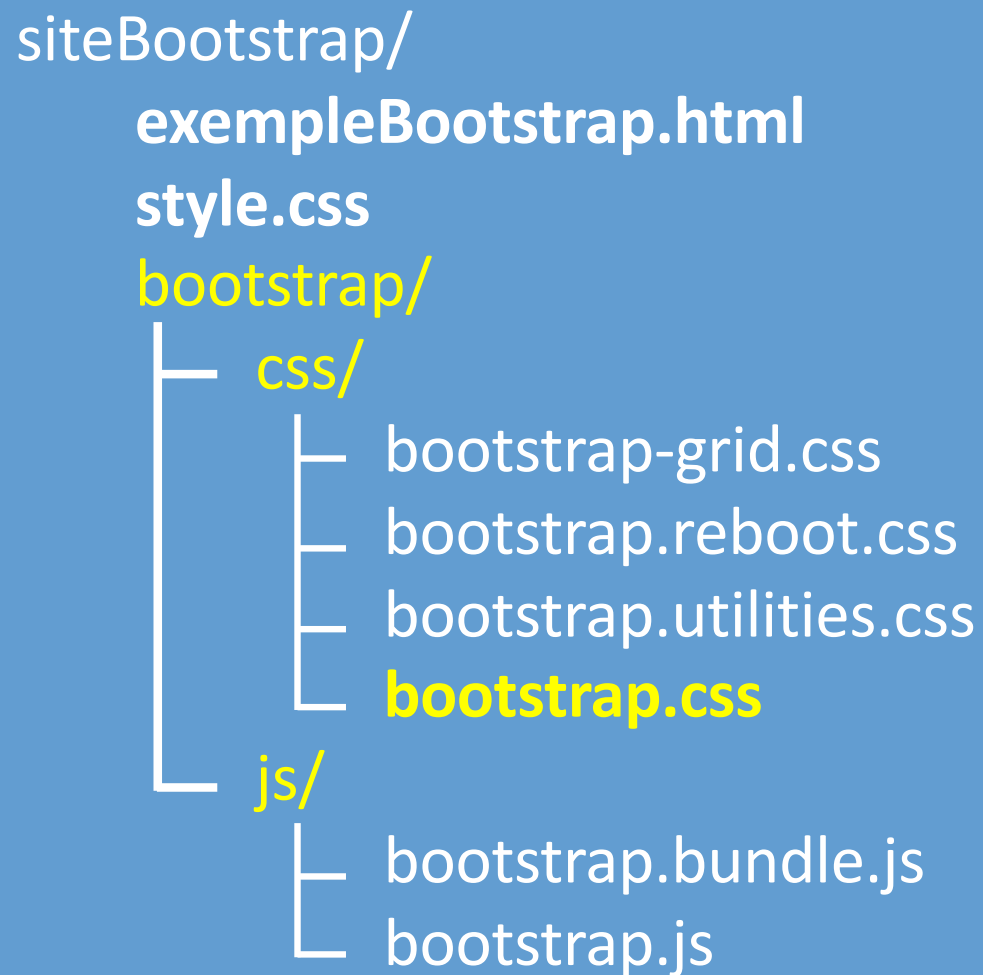
<https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css>

<https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js>

Installation de Bootstrap

Utilisation

- **Organisation**



```
graph LR; siteBootstrap[siteBootstrap/] --> exempleBootstrap[exempleBootstrap.html]; siteBootstrap --> styleCSS[style.css]; siteBootstrap --> bootstrap[bootstrap/]; bootstrap --> css[css/]; bootstrap --> js[js/]; css --> bootstrapGrid[bootstrap-grid.css]; css --> bootstrapReboot[bootstrap-reboot.css]; css --> bootstrapUtilities[bootstrap-utilities.css]; css --> bootstrapCSS[bootstrap.css]; js --> bootstrapBundle[bootstrap.bundle.js]; js --> bootstrapJS[bootstrap.js];
```

siteBootstrap/
exempleBootstrap.html
style.css
bootstrap/
├── css/
│ ├── bootstrap-grid.css
│ ├── bootstrap-reboot.css
│ ├── bootstrap-utilities.css
│ └── **bootstrap.css**
└── js/
 ├── bootstrap.bundle.js
 └── bootstrap.js

Grille – 1. Présentation

Orienté mobile

- **Aspect responsive** adapté en priorité pour les **mobiles**
- Prise en compte du support et **visualisation optimale** du site

<meta name="viewport"

content="width=device-width,

largeur

initial-scale=1,

zoom

shrink-to-fit=no" >

pas d'adaptation

pour des éléments hors écran

Grille – 1. Présentation

Définition du conteneur

- **3 classes permettent de préciser le conteneur :**
 - ✓ **.container** définit une largeur fixe entre chaque point de rupture

Bloc 4 colonnes

Bloc 8 colonnes

- ✓ **.container-fluid** définit une largeur maximale entièrement responsive

Bloc 4 colonnes

Bloc 8 colonnes

- ✓ **.container-sm ... container-xxl** définit un container de largeur maximale (fluid/responsive) jusqu'au point de rupture indiqué (sm...xxl).
Le container a ensuite une largeur fixe.
C'est un mixte entre les 2 classes précédentes.

Grille – 1. Présentation

Largeur et hauteur

- **Horizontalement :**
 - ✓ 12 colonnes maxi,
 - ✓ de même largeur,
 - ✓ proportionnelles à la largeur de la fenêtre
- **Verticalement :**
 - ✓ hauteur dépendant des contenus (plus haut contenu)

| | | | | | | | | | | | |
|--|--|----------------------------|--|--|--|--|--|--------------------------------|--|--|--|
| | | | | | | | | | | | |
| | | Élément de hauteur réduite | | | | | | Élément de hauteur plus grande | | | |
| | | | | | | | | | | | |

Positionnement d'un élément : sur une ou plusieurs colonnes

Grille – 2. Classes de base

Lignes et colonnes de la grille

- **Lignes**

- ✓ Classe "row" : définition d'une ligne

- **Colonnes**

- ✓ Classes "col-**x**-*" : définition du nombre de colonnes * selon le support **x**

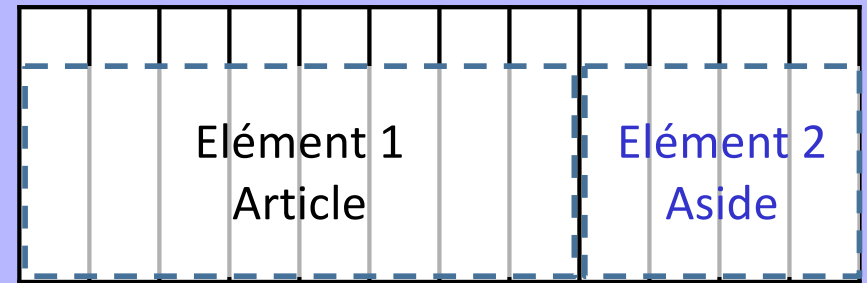
Pas de classe .col-**xs**
car mode par défaut

| Grille | Classe | Ecran | Taille écran |
|------------------------|------------|--------------------|--------------|
| extra small | .col-* | Smartphone | < 576px |
| sm : small | .col-sm-* | Grand smartphone | ≥ 576px |
| md : medium | .col-md-* | Tablette | ≥ 768px |
| lg : large | .col-lg-* | Desktop | ≥ 992px |
| xl : extra large | .col-xl-* | Grand desktop | ≥ 1200px |
| xxl: extra extra large | .col-xxl-* | Très grand desktop | ≥ 1400px |

Grille – Exemple : Desktop

1^{er} élément : 8 colonnes de large

2nd élément : 4 colonnes de large



```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
      shrink-to-fit=no" >
    <title>Exemple de structure</title>
    <link rel="stylesheet" href="bootstrap/css/bootstrap.css" >
  </head>
  <body>
    <!-- définition d'un conteneur fluid -->
    <!-- définition d'une ligne -->
    <!-- 8 colonnes pour la zone article-->
    <h2>Elément 1 - Article</h2>
    <p> Donec id elit non mi porta gravida at eget metus.</p>
    </ ? >
    <!-- 4 colonnes pour la zone aside-->
    <h2>Elément 2 - Aside</h2>
    <p> Donec id elit.</p>
    </ ? >
    </ ? >
  </body>
</html>
```

Grille – Exemple : Desktop

Habillage des éléments

The diagram shows a two-column layout. The left column contains a blue rounded rectangle with the text "Élément 1" and "Article" below it. The right column contains a blue rounded rectangle with the text "Élément 2" and "Aside" below it. The background is a grid of 10 columns and 4 rows, with the first column highlighted in light blue.

```
body {  
    padding-top: 10px;  
}  
[class*="col"] {  
    background-color: blue;  
    color: white;  
    border: 2px solid black;  
    border-radius: 5px;  
    line-height: 40px;  
    text-align: center;  
}
```

Sélecteur d'attribut [] :
cible tous les éléments qui
ont une classe contenant
la chaîne "col"

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
  <head>
    ...
    <link rel="stylesheet" href="bootstrap/css/bootstrap.css" >
    <link rel="stylesheet" href="style.css" >
  </head>
  <body>
    ...
  </body>
</html>
```

Habillage dans un fichier CSS spécifique : **style.css**

Lien vers ce
fichier CSS
spécifique :
<link...>

Grille – 3. Décalage d'élément

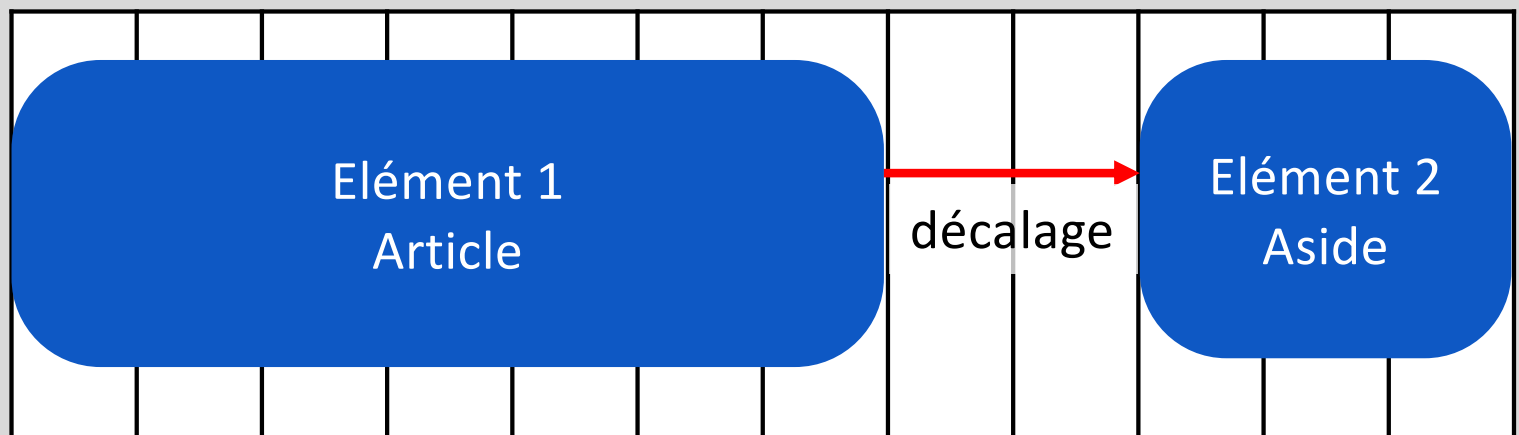
Colonnes vides

Classe pour créer un décalage

Classes : `.offset-*`, `.offset-sm-*`, `.offset-md-*`,
`.offset-lg-*`, `.offset-xl-*`, `.offset-xxl-*` selon le support
* : nombre de colonnes vides (ex : `.offset-md-4`)

➔ Insérer la classe offset dans l'élément décalé

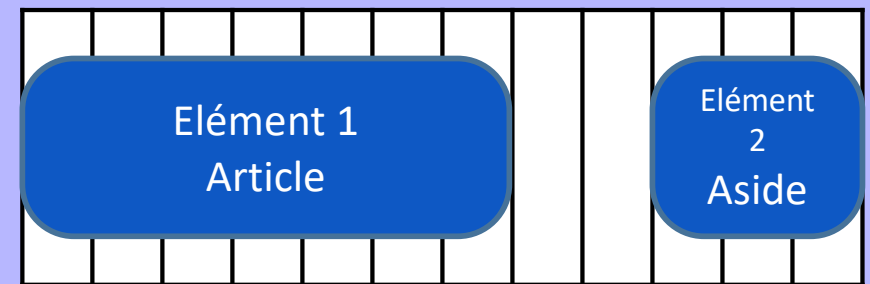
Exemple : Décaler Aside de 2 colonnes



Grille – Exemple : Desktop

1^{er} élément : 7 colonnes

2nd élément : 3 col après décalage 2 col



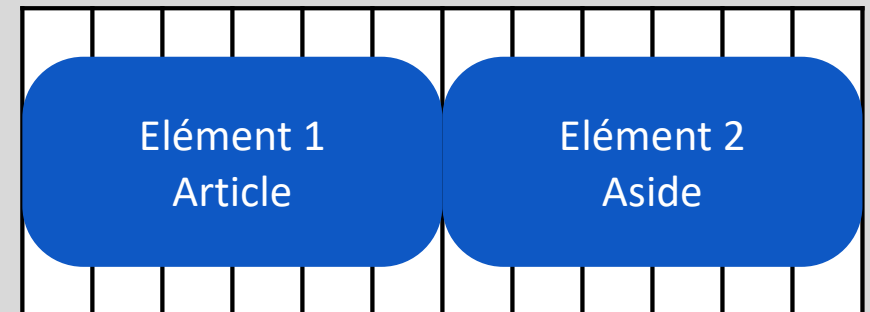
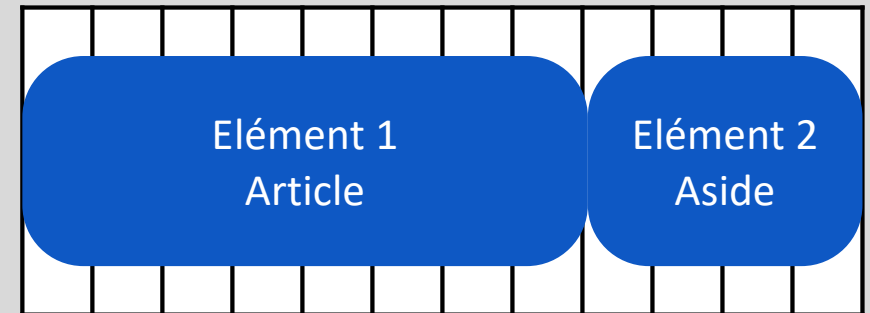
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ...
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <article ? >                                <!-- 7 col. pour la zone article -->
          <h2>Elément 1 - Article</h2>
          <p> Donec id elit non mi porta gravida at eget metus.</p>
        </article>
        <aside ? >                                    <!-- 3 col pour aside décalé de 2 col-->
          <h2>Elément 2 - Aside</h2>
          <p> Donec id elit.</p>
        </aside>
      </div>
    </div>
  </body>
</html>
```

Grille – 4. Adaptation au support

Modification de la largeur

A partir du 1^{er} exemple, adaptation pour plusieurs supports :

- Desktop
 - ✓ Élément 1 : 8 colonnes de large
 - ✓ Élément 2 : 4 colonnes de large
- Smartphone
 - ✓ Élément 1 : 6 colonnes de large
 - ✓ Élément 2 : 6 colonnes de large



Grille – 4. Adaptation au support

Modification de la largeur

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ...
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <article ? > <!-- 6 ou 8 colonnes pour la zone article-->
          <h2>Elément 1 - Article</h2>
          <p> Donec id elit non mi porta gravida at eget metus.</p>
        </article>
        <aside ? > <!-- 6 ou 4 colonnes pour la zone aside-->
          <h2>Elément 2 - Aside</h2>
          <p> Donec id elit.</p>
        </aside>
      </div>
    </div>
  </body>
</html>
```

Grille – 5. Largeur automatique

- Largeur automatique des éléments basée d'après le contenu

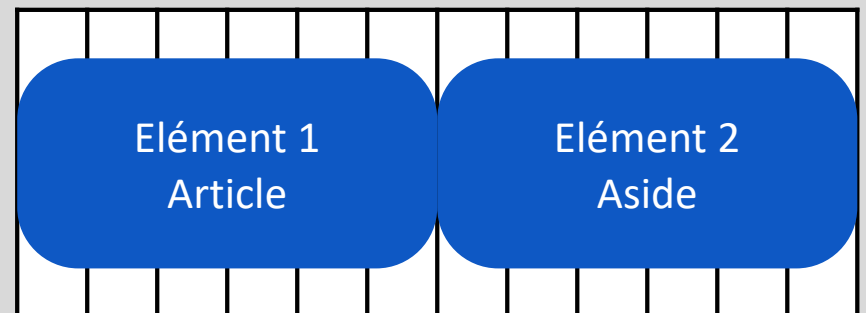
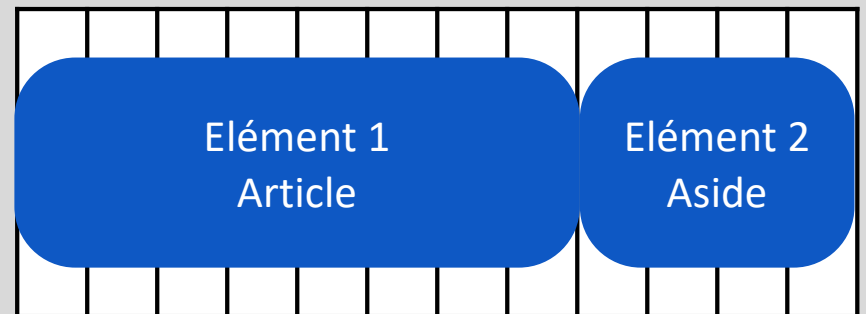
Classes : `.col-auto`,
`.col-sm-auto`,
`.col-md-auto`,
`.col-lg-auto`,
`.col-xl-auto`,
`.col-xxl-auto` selon le support

➔ la classe porteuse de la classe **auto** s'adapte à son contenu.
Si aucune largeur précise n'est donnée aux autres classes,
elles s'étirent sur toute la largeur

Grille – 5. Largeur automatique

Classes permettant de définir la largeur des éléments

- Desktop
 - ✓ Élément 1 : 8 colonnes de large
 - ✓ Élément 2 : 4 colonnes de large
- Tablette
 - ✓ Élément 1 : largeur complémentaire
 - ✓ Élément 2 : largeur auto
- Smartphone
 - ✓ Élément 1 : 6 colonnes de large
 - ✓ Élément 2 : 6 colonnes de large



Grille – 5. Largeur automatique

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ...
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <!-- 6, largeur complémentaire ou 8 pour l'article -->
        <article ? >
          <h2>Élément 1 - Article</h2>
          <p> Donec id elit non mi porta gravida at eget metus.</p>
        </article>
        <!-- 6, largeur auto ou 4 pour aside -->
        <aside ? >
          <h2>Élément 2 - Aside</h2>
          <p> Donec id elit.</p>
        </aside>
      </div>
    </div>
  </body>
</html>
```

Grille – 6. Réordonner des éléments

- Réordonnancement des éléments au sein d'une ligne

Classes : .order-1, .order-2, .order-3,order-11, .order-12
.order-sm-1, .order-sm-2,order-sm-11, .order-sm-12
.order-md-1, .order-md-2,order-md-11, .order-md-12
.order-lg-1, .order-lg-2,order-lg-11, .order-lg-12
.order-xl-1, .order-xl-2,order-xl-11, .order-xl-12
.order-xxl-1, .order-xxl-2,order-xxl-11, .order-xxl-12

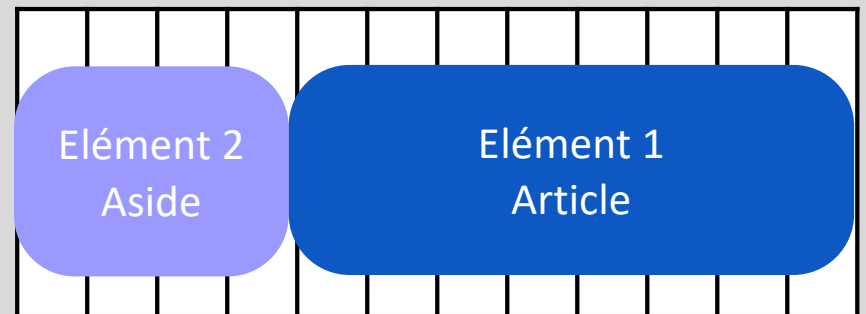
selon le support

➔ il existe également les classes .order-first (.order-sm-first...) et .order-last (.order-sm-last...) qui permettent de repositionner un élément en 1^{ère} ou en dernière position

Grille – 6. Réordonner des éléments

Classes permettant de réorganiser des éléments

- Desktop
 - ✓ Élément 1 : 8 colonnes de large dernière position
 - ✓ Élément 2 : 4 colonnes de large
- Tablette
 - ✓ Élément 1 : largeur complémentaire 1^{ère} position
 - ✓ Élément 2 : largeur auto
- Smartphone
 - ✓ Élément 1 : 6 colonnes de large dernière position
 - ✓ Élément 2 : 6 colonnes de large



Grille – 6. Réordonner des éléments

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ...
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <!-- 6, largeur comp. ou 8 pour article placé en 2ème, en 1er puis en 2ème -->
        <article ? >
          <h2>Elément 1 - Article</h2>
          <p> Donec id elit non mi porta gravida at eget metus.</p>
        </article>
        <!-- 6, auto ou 4 pour aside -->
        <aside ? >
          <h2>Elément 2 - Aside</h2>
          <p> Donec id elit.</p>
        </aside>
      </div>
    </div>
  </body>
</html>
```

Grille – 7. Afficher/Masquer des éléments

- **Masquage** des éléments

Classes : `.d-none`, `.d-sm-none`, ..., `.d-xl-none`, `.d-xxl-none`

- **Affichage** des éléments de type **block**

Classes : `.d-block`, `.d-sm-block`, ..., `.d-xl-block`, `.d-xxl-block`

- **Affichage** des éléments en ligne (**inline**)

Classes : `.d-inline`, `.d-sm-inline`, ..., `.d-xl-inline`, `.d-xxl-inline`

- **Affichage** des éléments en **inline-block**

Classes : `.d-inline-block`, `.d-sm-inline-block`, ... `.d-xxl-inline-block`

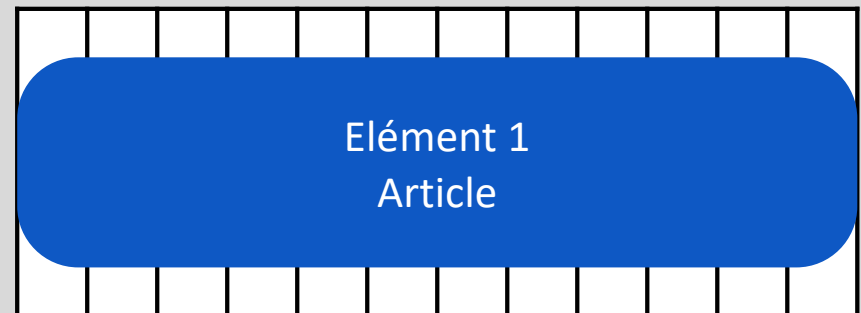
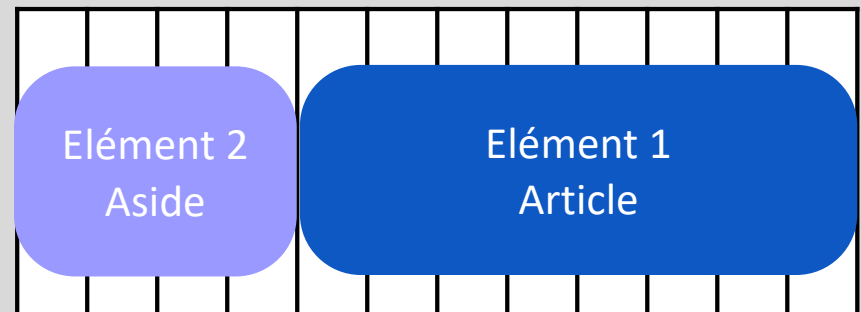
- **Affichage** des éléments en **flex**

Classes : `.d-flex`, `.d-sm-flex`, ..., `.d-xl-flex`, `.d-xxl-flex`

Grille – 7. Afficher/Masquer des éléments

Classes permettant de masquer des éléments

- Desktop
 - ✓ Élément 1 : 8 colonnes de large
 - ✓ Élément 2 : 4 colonnes de large
1^{ère} position
- Tablette
 - ✓ Élément 1 : largeur complémentaire
 - ✓ Élément 2 : largeur auto
- Smartphone
 - ✓ Élément 1 : Afficher sur 12 col
 - ✓ Élément 2 : Masquer



Grille – 7. Afficher/Masquer des éléments

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
  <head>
    ...
  </head>
  <body>
    <div class="container-fluid">
      <div class="row">
        <!-- 12, largeur comp. ou 8 pour l'article placé seul, en 1er puis en 2ème -->
        <article                                     ?                                     >
          <h2>Elément 1 - Article</h2>
          <p> Donec id elit non mi porta gravida at eget metus.</p>
        </article>
        <!-- Masqué, auto ou 4 pour l'aside non affiché, en 2ème puis en 1er -->
        <aside                                       ?                                       >
          <h2>Elément 2 - Aside</h2>
          <p> Donec id elit.</p>
        </aside>
      </div>
    </div>
  </body>
</html>
```


Grille – 8. Aligner les éléments

Alignement global

- Alignement horizontal

* : " ", sm, md, lg, xl, xxl

Classes : .justify-content-*
-start, .justify-content-*
-center, .justify-content-*
-end, .justify-content-*
-around, .justify-content-*
-between, .justify-content-*
-evenly

→ Insérer la classe dans la ligne contenant les éléments à aligner

| | | | | | |
|---------|---------|---------|-------|---------|-------|
| Article | Aside | | | | |
| | | Article | Aside | | |
| | | | | Article | Aside |
| | Article | | | Aside | |
| Article | | | | | Aside |
| | Article | | | Aside | |

.justify-content-*
-start

.justify-content-*
-center

.justify-content-*
-end

.justify-content-*
-around

.justify-content-*
-between

.justify-content-*
-evenly

Grille – 8. Aligner les éléments

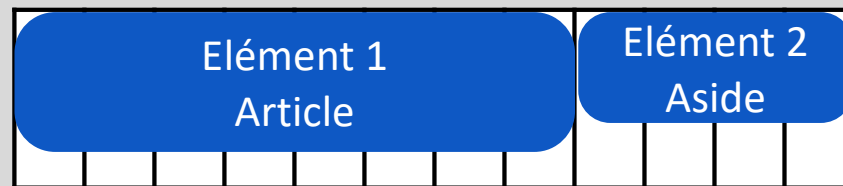
Alignement global

- Alignement vertical

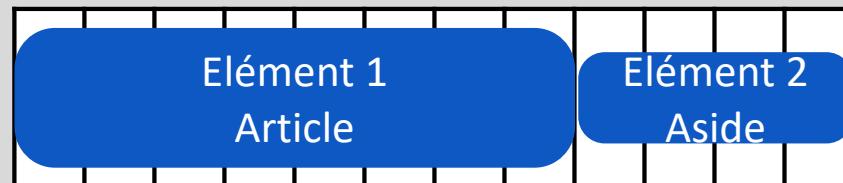
* : " ", sm, md, lg, xl, xxl

Classes : `.align-items-*start`, `.align-items-*center`,
`.align-items-*end`, `.align-items-*stretch`

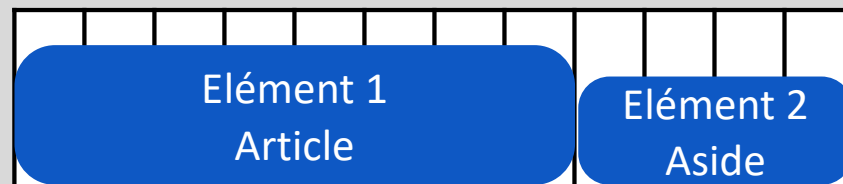
→ Insérer la classe dans la ligne contenant les éléments à aligner



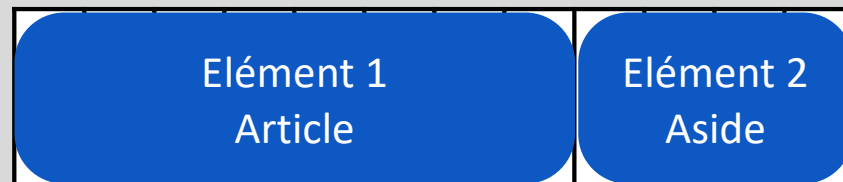
`.align-items-*start`



`.align-items-*center`



`.align-items-*end`



`.align-items-*stretch`

Grille – 8. Aligner les éléments

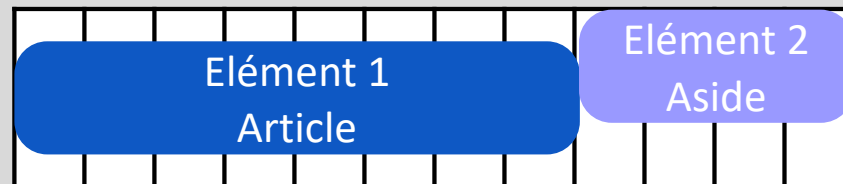
Alignement vertical

- Alignement particulier

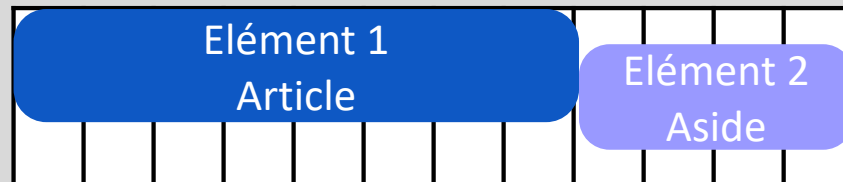
* : " ", sm, md, lg, xl, xxl

Classes : `.align-self-*start`, `.align-self-*center`,
`.align-self-*end`, `.align-self-*stretch`

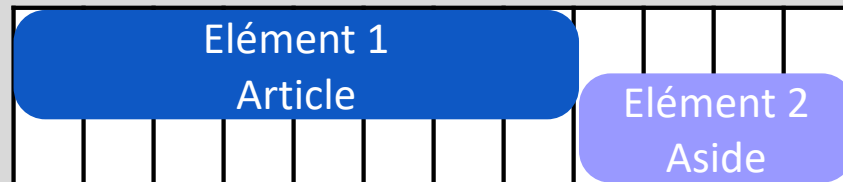
→ Insérer la classe dans l'élément à aligner



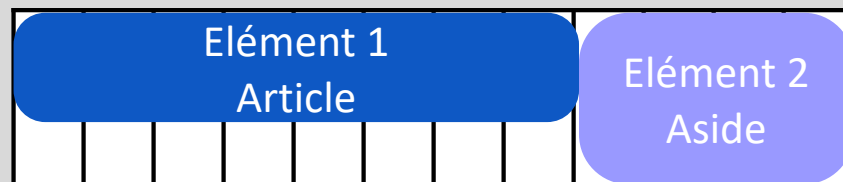
`.align-self-*start`



`.align-self-*center`



`.align-self-*end`



`.align-self-*stretch`

Grille – 9. Placer les éléments

Marges

`.pt-4` : padding-top: 1.5rem;

`.mx-auto` : margin-left et margin-right à auto;

`.p-md-3` : padding: 1rem; pour le format md

- **Gestion des espaces internes et externes**

Principes : {propriété}{direction}-{format}-{taille}

→ **Propriété**

"m" pour margin,

"p" pour padding

→ **Direction**

Rien pour un espace dans toutes les directions

"l" left,

"r" right,

"x" left et right

"t" top,

"b" bottom,

"y" top et bottom

→ **Format**

Rien ou sm, md, lg, xl et xxl

→ **Taille** : 0, 1 (.25rem), 2 (.5rem), 3 (1rem), 4 (1.5rem), 5 (3rem)

auto (valeur automatique pour les marges horizontales)

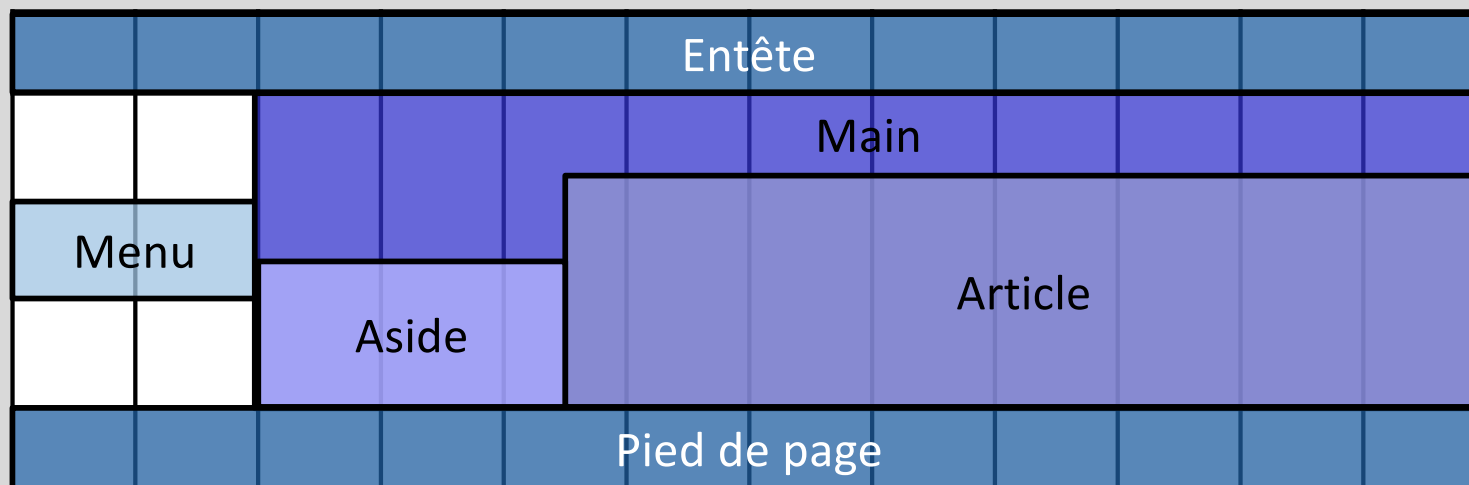
Grille – 10. Exercice

Format Tablette

Réaliser la structure suivante (Zoning) au format Tablette :

- Entête : 12 colonnes
- Menu : 2 colonnes
- Main : 10 colonnes
 - ✓ Aside : 3 colonnes
 - ✓ Article : 9 colonnes
- Pied de page : 12 colonnes

Remarque : chaque bloc constitue un contenant pouvant être divisé en 12 colonnes
Exemple : Main divisé en Article et Aside



Rq : Une hauteur sera précisée pour Article pour mieux visualiser l'effet avec Aside

Grille – 10. Exercice

Format Smartphone

Adapter la structure au format Smartphone :

- Entête : 12 colonnes
- Menu : 3 colonnes
- Main : 9 colonnes
 - ✓ Article : 12 colonnes
 - ✓ Aside : masqué
- Pied de page : 12 colonnes

| | | | | | | | | | |
|------|--|--|--------------|--|--|--|--|--|--|
| | | | Entête | | | | | | |
| | | | Main | | | | | | |
| Menu | | | Article | | | | | | |
| | | | | | | | | | |
| | | | Pied de page | | | | | | |

Grille – 10. Exercice

Format Desktop

Adapter la structure au format Desktop :

- Entête : 12 colonnes
- Menu : 2 colonnes
- Main : 10 colonnes
 - ✓ Article : 8 colonnes
 - ✓ Aside : 2 colonnes
- Pied de page : 12 colonnes

