



Le Framework Bootstrap

V5.2

Issam REBAI – Équipe MOTEL – Département informatique – Campus de Brest
✉ issam.rebai@imt-atlantique.fr

Version 0.3 [R.2023.01.02]



- Présentation
- Intégration
- Viewport
- Layout
- Formulaire
- Les éléments complexes
- Icônes

AU SOMMAIRE

INT475 - Développement Web
Front End pour les débutants

[2]



Présentation

- Bootstrap est un framework HTML5, CSS3 et JavaScript pour réaliser des sites web responsives dédiés en priorité aux mobiles
 - Avantages:
 - Très populaire : large communauté active, ressources communautaires, aide et conseil sur le web
 - Développement rapide : facilite la mise en place de site avec un design modulable (responsive)
 - Personnalisable : plusieurs thèmes disponibles
 - Open Source : possibilité d'enrichir et d'adapter le framework
 - Compatibilité avec les navigateurs : supporté par les navigateurs récents
 - Inconvénients
 - Similarité : sans personnalisation ressemblance à plusieurs sites web
 - Apprentissage : coût conséquent (temps et effort à fournir) pour maîtriser le framework
- Il définit des classes et des composants regroupant des règles prédéfinies et des styles applicables sur des éléments HTML

Présentation

- Fiche signalétique
 - Nom: Bootstrap
 - Naissance : 2010 sous une ancienne identité : Twitter Blueprint
 - Créateurs : Mark Otto et Jacob Thornton
 - Développeurs : Twitter, depuis 08/2011 projet Open Source
 - Écrit en: HTML, CSS, JS et Sass (depuis la V4)
- Sites web
 - Officiel: <https://getbootstrap.com/>
 - GIT: <https://github.com/twbs>
- Versions
 - 1^{er} version : 19-08-2011
 - Dernière stable : 5.2.3 (02-01-2023)
 - Dernière en développement : 5.3.0-alpha1 (02-01-2023)

Présentation ► v5 versus v4

Bootstrap v5
fonctionne
correctement sur
les navigateurs

- Chrome ≥ 60
- Firefox ≥ 60
- Firefox ESR
- iOS Safari ≥ 12
- Safari ≥ 12
- Microsoft Edge ≥ 16

Bootstrap v5
n'est pas
supporté par

- Internet Explorer 10 et 11 → utiliser Bootstrap v4.

Présentation ► la v5 versus v4

Framework plus léger, simple et rapide

Fin d'utilisation de jQuery → les requêtes doivent être réalisées en JS

Fin du support d'Internet Explorer (10 et 11)

Moteur RFS (Responsive Font Sizes) activé par défaut

- Redimensionnement des polices selon la taille de la fenêtre

Nouvelle librairie d'icônes

Intégration de l'écriture RTL (Right-To-Left)

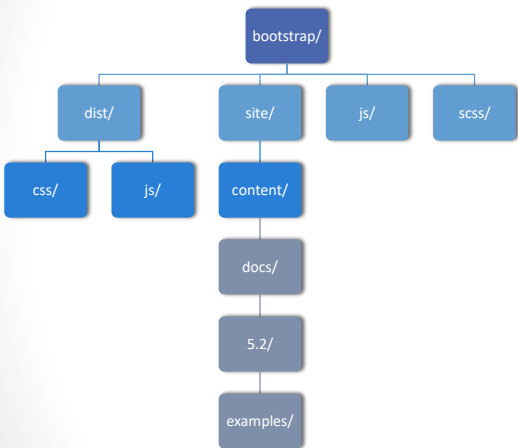
Voir le détail sur : <https://getbootstrap.com/docs/5.0/migration/>

Intégration de Bootstrap

source (<https://getbootstrap.com/docs/5.2/getting-started/contents/>)

- Téléchargement des sources (CSS+JS) et compilation des fichiers
 - Lien : <https://github.com/twbs/bootstrap/archive/v5.2.3.zip>
 - Outils nécessaires : Pipeline de compilation + Compilateur Sass + autoprefixer CSS
- Téléchargement des fichiers compilés et minifiés
 - Lien : <https://github.com/twbs/bootstrap/releases/download/v5.2.3/bootstrap-5.2.3-dist.zip>
 - Décompresser dans le dossier du projet et lier les fichiers CSS et JS aux pages web
- Référencement du framework à partir du CDN recommandé par Bootstrap
 - Lien du tuto : <https://getbootstrap.com/docs/5.2/getting-started/introduction/>
 - CDN : Content Delivery Network (réseau de distribution de contenu)
 - Un système de serveurs distribués géographiquement et fournissant le contenu web à un débit élevé.
- Les gestionnaires de paquetages : npm, yarn, RubyGems, Composer, Nuget
 - `npm install bootstrap@5.2.3`
 - `gem install bootstrap -v 5.2.3`

Intégration de Bootstrap ▶ arborescence – les dossiers et fichiers JS



JS files	Popper
bootstrap.bundle.js bootstrap.bundle.min.js	✓
bootstrap.js bootstrap.min.js	✗

- Dossiers du paquetage code source
 - js, scss : code source
 - dist : livrable
 - site/content/docs : documentation
 - site/content/docs/5.2/examples/ : exemples d'usage
- Popper est une librairie JS nécessaire pour certains plugins. Elle permet d'afficher des info-bulles élaborées.

Intégration de Bootstrap ► les fichiers CSS et leur contenu

CSS files	Layout	Content	Components	Utilities
bootstrap.css bootstrap.rtl.css bootstrap.min.css bootstrap.rtl.min.css	✓	✓	✓	✓
bootstrap-grid.css bootstrap-grid.rtl.css bootstrap-grid.min.css bootstrap-grid.rtl.min.css	grid system	✗	✗	flex utilities
bootstrap-utilities.css bootstrap-utilities.rtl.css bootstrap-utilities.min.css bootstrap-utilities.rtl.min.css	✗	✗	✗	✓
bootstrap-reboot.css bootstrap-reboot.rtl.css bootstrap-reboot.min.css bootstrap-reboot.rtl.min.css	✗	Reboot	✗	✗

Intégration de Bootstrap ► les fichiers CSS et leur contenu

- Les fichiers CSS contiennent
 - Des sélecteurs
 - D'éléments : styles de base appliqués par défaut aux éléments HTML (contenu de [Reboot](#))
 - De classes : styles prédéfinis associables à des éléments HTML via l'attribut class
 - Des règles [@media](#) applicables selon la taille de l'écran pour définir un système de point d'arrêt ([breakpoint](#)) ➔
 - Mobile first & responsive design
 - Les propriétés CSS en dehors des règles s'appliquent pour tous les écrans
 - Les propriétés CSS dans les règles s'appliquent selon la taille de l'écran
 - Des ordres d'affichage des éléments et composants Bootstrap pour gérer leur superposition quand ils sont sur une même page (propriété [z-index](#))
- Principe :
 - Ne pas modifier les fichiers CSS Bootstrap
 - Trouver les noms des classes Bootstrap ayant le style souhaité et attribuer ces noms à la valeur de l'attribut [class](#) des éléments auxquels on souhaite appliquer le style

Viewport

La zone d'affichage visible sur l'appareil

Pour un navigateur : fenêtre du navigateur sans les éléments de l'interface du navigateur

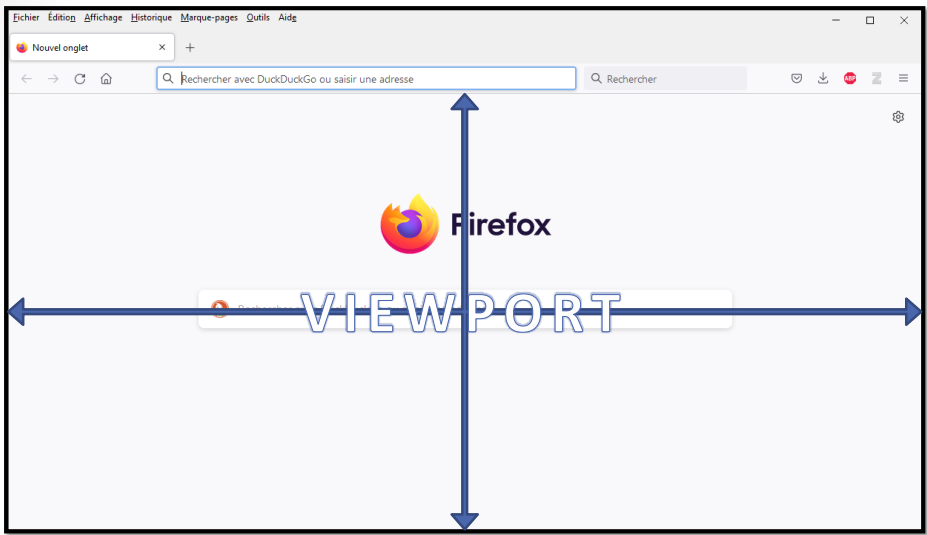
En plein écran : viewport = taille de l'écran

Pour afficher une page plus grande que le viewport, l'utilisateur défile le contenu de la page via les ascenseurs

Pour un mobile : la taille de la fenêtre d'affichage des pages web est souvent différente de la taille physique des écrans → zone virtuelle (ex: 960px) plus large que la taille réelle de l'écran

Objectif: éviter de dézoomer lors de l'affichage d'un site non optimisé pour le mobile

Le navigateur applique une réduction du résultat pour permettre à l'utilisateur de visualiser le contenu entier → peu lisible
→ se déplacer ou zoomer au doigt



Viewport

Cas d'un navigateur web sur PC

iPhone 5


largeur :

largeur "réelle" : 640px
 screen.width (JS) : 320px
 device-width : 320px
 viewport (Safari) : 980px

hauteur :

hauteur "réelle" : 1136px
 screen.height (JS) : 568px
 device-height : 568px
 viewport (Safari) : 1090px

Viewport

Cas d'un mobile

Viewport

Pour contrôler le viewport, il est recommandé d'ajouter dans le document HTML, la balise méta :

```
<meta name="viewport" content="width=device-width,
initial-scale=1, shrink-to-fit=no, user-scalable=no">
```

- **width=device-width** permet d'utiliser la taille réelle de l'écran du mobile au lieu de la taille de la zone virtuelle
- **initial-scale=1** Initialise le zoom à 1
- **shrink-to-fit=no** Interdit à la page de se mettre à l'échelle pour s'adapter à la fenêtre
- **user-scalable=no** Interdit à l'utilisateur de zoomer ou dé-zoomer

Viewport ▶ les attributs de la balise <meta> viewport

width	• largeur de fenêtre viewport (par exemple width="device-width")
height	• hauteur de fenêtre viewport (par exemple height="device-height")
initial-scale	• niveau de zoom initial (par exemple initial-scale="1.0")
minimum-scale	• niveau de zoom minimal (par exemple minimum-scale="0.5")
maximum-scale	• niveau de zoom maximal (par exemple maximum-scale="3.0"). Attention, la valeur "1.0" interdit le zoom et peut rendre vos pages inaccessibles
user-scalable	• possibilité à l'utilisateur de zoomer (par exemple user-scalable="yes"). Attention, la valeur "no" interdit le zoom et peut rendre les pages inaccessibles si le CSS n'est pas responsive

Les conteneurs (container)
Système de grilles
Le système de marge et d'espacement (Margin & Padding)

LAYOUT

Les conteneurs

- Sont nécessaires à Bootstrap pour y positionner tout le contenu HTML de la page
- Sont les éléments de mise en page les plus basiques
- Sont utilisés pour contenir, remplir et centrer leur contenu
- Sont requis pour pouvoir utiliser le [système de grille](#)
- Sont emboîtables [mais les cas d'usages sont rares](#)
- 3 classes de conteneurs :
 - `.container` : 100% si <576px, sinon largeur max réactif à chaque point d'arrêt (breakpoint)
 - `.container-fluid` : toujours 100% de la largeur
 - `.container-{breakpoint}` : 100% jusqu'au point d'arrêt et ensuite largeur max
- Utilisation : définir un élément HTML structurant avec `class="container"` ou `container-fluid` ou `container-{breakpoint}` qui servira de conteneur générique dans lequel tout le contenu HTML du body sera placé.
 - `.container-fluid` → pour les menus, barres de navigation, header et footer
 - `.container` → le contenu de la page entre le header et le footer



Les conteneurs

[Exemple de démonstration](#)

	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%



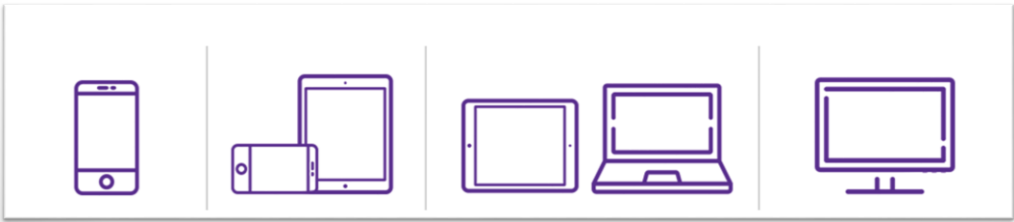
Système de grille

- Est basé sur flexbox (Flexible Box Layout Module) un modèle de disposition unidimensionnel → gère une seule dimension à la fois (ligne ou colonne)
 - En opposition au modèle de grille CSS (CSS grid) bidimensionnel contrôlant à la fois les lignes et colonnes.
 - L'axe principal est horizontal → les lignes (`.row`) servent de conteneurs aux colonnes.
 - L'axe secondaire est vertical → les colonnes (`.col`)
- Est un découpage virtuel de l'espace en plusieurs lignes et 12 colonnes qui peuvent être agrégés en espaces plus larges symétriques ou asymétriques

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Système de grille ▶ les largeurs d'écrans et classes associées

Classe	Nom	Préfixe	Catégorie d'écran	Largeur viewport	Largeur max du conteneur max-width
xs	Extra small	.col-	Très petits téléphones	< 576 px	Rien (auto)
sm	Small	.col-sm-	Petits téléphones & Tablettes	≥ 576 px	540 px
md	Medium	.col-md-	Tablettes	≥ 768 px	720 px
lg	Large	.col-lg-	Petits écrans d'ordinateurs	≥ 992 px	960 px
xl	Extra large	.col-xl-	Écrans larges d'ordinateurs	≥ 1200 px	1140 px
xxl	Extra extra Large	.col-xxl-	Grand écran et télévision	≥ 1400 px	1320 px



Système de grille ► les classes

`.row` : créer une ligne pour mettre des colonnes

`.col` : créer une colonne de largeur équitable avec les autres colonnes de la ligne

`.col-{breakpoint}` : créer une colonne de largeur équitable avec les autres colonnes de la ligne pour la classe d'écran breakpoint

- Si la largeur du viewport est supérieure à la largeur de la classe breakpoint alors les colonnes restent sur la même ligne
sinon chacune occupe toute la largeur du viewport → une colonne par ligne

`.col-1, .col-2, ..., .col-12` : créer une colonne de taille explicite occupant n fois la taille d'une colonne de base / $n \in [1,12]$

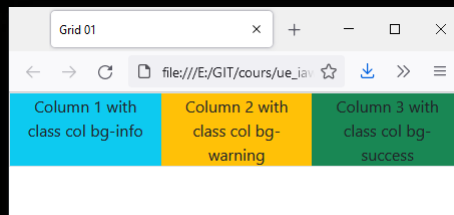
`.col-{breakpoint}-n` : créer une colonne de taille explicite occupant n fois la taille d'une colonne de base pour la classe d'écran breakpoint / breakpoint $\in \{sm, md, lg, xl, xxl\}$ et $n \in [1,12]$

- Chaque colonne occupe un nombre de colonnes de base différent selon la taille de l'écran

`.col-{breakpoint}-auto` : créer une colonne dont la taille est calculée automatiquement en fonction de son contenu pour la classe d'écran breakpoint

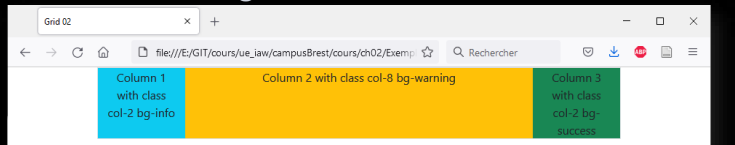
Système de grille ► exemples

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Grid 01</title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- Bootstrap CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet" >
  </head>
  <body>
    <div class="container bg-primary text-center border">
      <div class="row">
        <div class="col bg-info">
          Column 1 with class col bg-info
        </div>
        <div class="col bg-warning">
          Column 2 with class col bg-warning
        </div>
        <div class="col bg-success">
          Column 3 with class col bg-success
        </div>
      </div>
    </div>
    <script src="js/bootstrap.js"></script>
  </body>
</html>
```



Système de grille ► exemples

```
<div class="container bg-primary text-center border ">
  <div class="row">
    <div class="col-2 bg-info">
      Column 1 with class col-2 bg-info
    </div>
    <div class="col-8 bg-warning">
      Column 2 with class col-8 bg-warning
    </div>
    <div class="col-2 bg-success">
      Column 3 with class col-2 bg-success
    </div>
  </div>
</div>
```



Système de grille ► les règles

- Les lignes (.row) doivent être placées dans des conteneurs
- Les lignes ne doivent être utilisées que pour contenir des colonnes
- Les éléments de classes .col doivent absolument être des enfants directs des éléments de classes .row
- Quand plusieurs éléments d'une même ligne dépassent la largeur du viewport, les derniers éléments s'affichent sur la ligne suivante
- Le padding entre les colonnes est de 15px par défaut
 - La classe .no-gutters le supprime
 - La classe .g-* change l'espacement horizontal et vertical * ∈ [0,5]
 - La classe .gx-* change l'espacement horizontal
 - La classe .gy-* change l'espacement vertical
- Quand le nombre de colonnes de base qui doivent être utilisées pour créer chacune des colonnes personnalisées, n'est pas explicitement précisé alors l'espace dans la ligne sera distribué équitablement entre chacune des colonnes créées.

Système de grille ► les règles

- Si dans une ligne, il y a des colonnes de taille explicite et des colonnes sans taille explicite, les colonnes de taille explicite occuperont l'espace en premier puis l'espace restant sera distribué entre les colonnes définies avec la classe `.col`.
- Si dans une ligne, la somme de taille des colonnes explicite > 12, alors les colonnes ne rentrant pas dans la ligne iront à la ligne suivante
- Si le contenu d'une colonne ne tient pas dans l'espace qui lui a été originellement attribué, l'espace dans la ligne sera réarrangé pour que cette colonne ait une place suffisante à l'affichage de son contenu → peut provoquer le passage automatique à la ligne des autres colonnes personnalisées
- Il est possible d'utiliser plusieurs breakpoint pour une même colonne
- Pour forcer le retour à la ligne des colonnes, on peut utiliser une div avec la classe `.w-100`
 - `<div class="w-100"/>`

Système de grille ► alignement vertical

- Par défaut :
 - une colonne occupe toute la hauteur d'une ligne.
 - La hauteur d'une ligne est définie par la hauteur de la colonne ayant le contenu prenant plus de place en hauteur
- Si on précise la hauteur d'une ligne ou si on demande à ses colonnes d'adapter leur hauteur à leur contenu alors on peut imposer un alignement vertical aux colonnes
 - En alignant toutes les colonnes d'une même ligne, on ajoute à la ligne (div de classe `row`) la classe :
 - `.align-items-start` → colonnes alignées en début (en haut) de la ligne
 - `.align-items-center` → colonnes alignées au centre de la ligne
 - `.align-items-end` → colonnes alignées en fin (en bas) de la ligne
 - En alignant chaque colonne (div de classe `col`) individuellement
 - `.align-self-start` → colonne alignée en début (en haut) de la ligne ;
 - `.align-self-center` → colonne alignée au centre de la ligne ;
 - `.align-self-end` → colonne alignée en fin (en bas) de la ligne.

Système de grille ► alignement horizontal

- Il s'applique quand les colonnes d'une ligne n'occupent pas tout l'espace de la ligne :
 - La somme des classes `.col-{nombre}` < 12
- Il s'effectue en ajoutant à la ligne (div de classe `row`) la classe :
 - `.justify-content-start` → colonnes alignées en début (à gauche) de la ligne:
Comportement par défaut
 - `.justify-content-center` → colonnes alignées au centre de la ligne
 - `.justify-content-end` → colonnes alignées en fin (à droite) de la ligne
 - `.justify-content-around` → colonnes équitablement réparties sur la ligne avec écart entre le bord et la 1^{er} colonne + écart entre le bord et la dernière colonne = écart entre 2 colonnes en milieu
 - `.justify-content-between` → colonnes équitablement réparties sur la ligne avec la première colonne collée au début de la ligne et la dernière colonne collée à la fin de la ligne
 - `.justify-content-evenly` → colonnes équitablement réparties sur la ligne avec l'écart entre le bord et la première colonne et l'écart entre le bord et la dernière colonne est identique à l'écart entre des colonnes du milieu

Système de grille ► ordonnancement

- Il est possible de changer l'ordre d'affichage visuel des colonnes par rapport à leur ordre de déclaration dans le code avec la classe `.order-*`
- Par défaut les colonnes ont un ordre égal à 0
- L'ordonnancement de l'affichage des colonnes se fait par ordre croissant des valeurs de la classe `order`
- Les classes `order` sont responsives et supportent la notion de breakpoint
`.order-{breakpoint}-*`
- Les classes disponibles sont
 - `.order-1` à `.order-12`
 - `.order-first` ⇔ `.order-1`
 - `.order-last` ⇔ `.order-13`

Le système de marge et d'espacement (Margin & Padding)

- Bootstrap définit plusieurs classes pour spécifier la marge et l'écart de remplissage
- Format de nommage des classes
 - La classe xs : {property}{sides}-{size}
 - Les autres (sm, md, lg, xl, xxl) : {property}{sides}-{breakpoint}-{size}
- Sachant que :
\$spacer est une variable déclarant la valeur par défaut de l'espacement

property =	sides =	size =
m → margin p → padding	t → margin-top padding-top b → margin-bottom padding-bottom s → (start) margin-left padding-left en LTR, margin-right padding-right en RTL e → (end) margin-right padding-right en LTR, margin-left padding-left en RTL x → margin-left & margin-right padding-left & padding-right y → margin-top & margin-bottom padding-top & padding-bottom blank → les 4 cotés d'un element	0 → supprime la marge ou le padding ⇔ 0 1 → par défaut \$spacer * 0.25 2 → par défaut \$spacer * 0.5 3 → par défaut \$spacer 4 → par défaut \$spacer * 1.5 5 → par défaut \$spacer * 3 auto → margin ou padding automatique

Exercices de l'activité B1 sur l'espace Moodle

PAUSE ACTIVITÉ PRATIQUE