IS3/IS2A4 - Internet

CM2 - Bootstrap

Thibault Liétard
Polytech' Lille

2025

Avant tout : CSS

Définition

Qu'est-ce que CSS ? [Fondation Mozilla]
Les CSS (Cascading Style Sheets en anglais, ou « feuilles de style en cascade ») sont le code utilisé pour mettre en forme une page web.

Concrètement ?
Le CSS sert à mettre du texte en rouge, centrer un paragraphe, réduire la taille d'une image... on a l'habitude de dire que CSS gère la forme, alors que HTML gère le fond (en Javascript le dynamisme, mais nous avons déjà vu ça).

Où écrire le CSS ?

fichier .css
On écrit toujours le CSS dans un fichier à part, situé en général dans le même dossier que le HTML ou dans un sous-dossier. Ce fichier doit être d'extension .css

Référence
<link href="styles/style.css" rel="stylesheet" type="text/css">

Format d'une feuille de style CSS

bloc de commande
Le CSS s'écrit par bloc de commande, c'est à dire par un ensemble de valeurs qu'on applique à des propriétés pour un ensemble d'objets donnés.

```
01 - selecteur {
02 -     propriete : value;
03 - }
```

Exemples

```css
p {
  color: red;
  width: 500px;
  border: 1px solid black;
}

p,li,h1 {
  color: red;
}

.toto-class {
  font-size: 60px;
}

#toto-id {
  text-align: center;
}
```

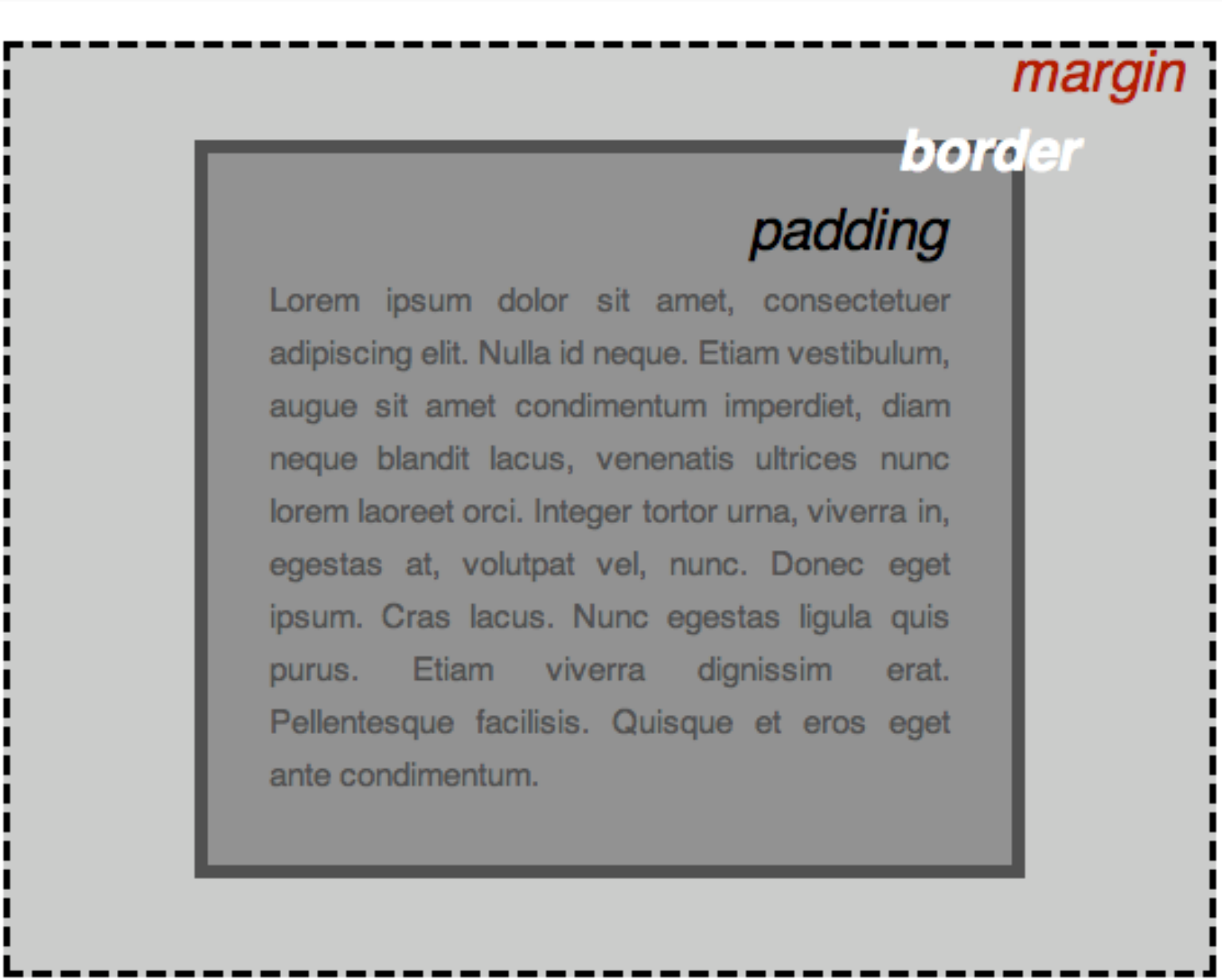
Boites

Principe
Avec CSS, tout tourne autour de boîtes : définir leurs tailles, leurs couleurs, leurs positions, etc. Les éléments HTML d'une page peuvent, pour la plupart, être vus comme des boîtes placées les unes sur/dans les autres.

Propriétés des boîtes

Espacement

- **padding** : l'espace autour, proche du contenu (par exemple, l'espace autour du texte d'un paragraphe) (en français, remplissage)
- **border** : la ligne qui est juste autour du padding (en français cela correspond à la bordure)
- **margin** : l'espace extérieur, autour de l'élément (en français cela correspond à la marge)


margin
border
padding
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla id neque. Etiam vestibulum, augue sit amet condimentum imperdiet, diam neque blandit lacus, venenatis ultrices nunc lorem laoreet orci. Integer tortor urna, viverra in, egestas at, volutpat vel, nunc. Donec eget ipsum. Cras lacus. Nunc egestas ligula quis purus. Etiam viverra dignissim erat. Pellentesque facilisis. Quisque et eros eget ante condimentum.

Exemple

Le code CSS

```
20  body {
21      width: 600px;
22      margin: 0 auto;
23      background-color: #FF9500;
24      padding: 0 20px 20px 20px;
25      border: 5px solid black;
26  }
```

Le rendu

Exemple 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Cas des images

Problème
Les images sont des éléments de type *inline*, c'est à dire qui sont considérés comme faisant partie des lignes de texte. On ne peut donc pas leur appliquer de marge, remplissage, bordure, etc... Sauf en les transformant en éléments de type bloc (comme body, div,...)

Solution

```
 9  img {
10    display: block;
11    margin: 0 auto;
12  }
```

Introduction à Bootstrap

Historique

Wikipédia
Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript

Histoire

- Créé chez twitter en 2010
- Passe en licence Open Source en 2011
- v3 en 2013 : *mobile first*
- v4 : finalisée en 2018. Dernière version stable
- Bootstrap5 en développement (bêta depuis décembre 2020)

Principe

Avant tout...
Bootstrap fournit une feuille de style CSS qui contient des définitions de base pour tous les composants HTML, ce qui permet de disposer d'une apparence uniforme pour les textes, tableaux et les éléments de formulaires.

Ensuite...
La modularité de Bootstrap consiste en une série de feuilles de styles LESS (un langage compilé qui donne du CSS). Une feuille de style principale (Bootstrap.less) englobe les feuilles de style des composants. Les développeurs peuvent alors sélectionner les composants dont ils ont besoin en modifiant cette feuille principale.

Enfin...
Bootstrap repose sur un système de grille pour placer les éléments dans une page.

Utiliser Bootstrap

inclure Bootstrap
Il faut commencer par lier le fichier HTML aux feuilles de style de Bootstrap :

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com
/bootstrap/4.4.1/css/bootstrap.min.css" integrity="
sha384-Vkoo8x4CGsO3+Hhxv8T/
Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="
anonymous">
```

Les scripts JS

Mais ça n'est pas tout. Bootstrap utilise les API JQuery et Popper, en plus de sa propre API js.

```
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imG
FAV0wwj1yYfoRSJoZ+n" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist
/usm/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEW
ldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous
"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1
/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMB
JnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="
anonymous"></script>
```

Éléments de base

La grille

Principe
Bootstrap utilise un principe de grille : chaque élément est disposé sur une ou plusieurs lignes et une ou plusieurs colonnes de cette grille.

Bootstrap propose un ensemble de classes qui permet de gérer facilement la position des éléments sur la grille. En fait, l'API fournie le code CSS associée à ces classes pour que le positionnement se fasse automatiquement.

Container

Usage
C'est le bloc d'élément le plus basique de Bootstrap. En général, tout autre bloc est contenu à minima dans un *container*.

Utilisation
On distinguera 2 types de *container* :

- **container** : il est de largeur *max-width*. En effet, Bootstrap instaure par défaut de confortables marges horizontal pour aérer le contenu. Le document ne fait donc pas toute la largeur de la page.
- **container-fluid** : idem, mais 100% de la largeur de la page.

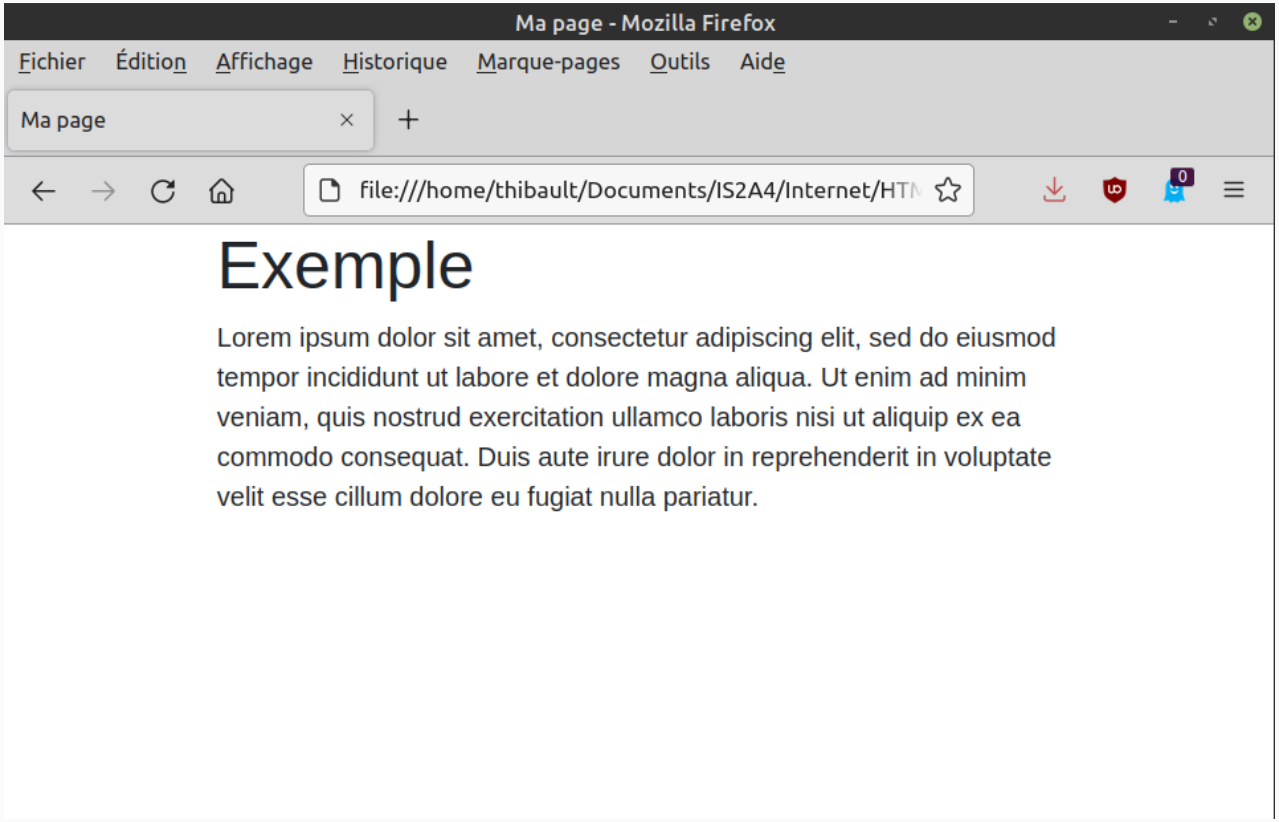
Exemple

HTML

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Ma page</title>
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com
        /bootstrap/4.4.1/css/bootstrap.min.css" integrity="
        sha384-Vkoo8x4CGsO3+Hhxv8T/
        Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="
        anonymous">
    </head>
    <body>
        <div class="container">
            <main>
                <h1>Exemple</h1>
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing
                elit, sed do eiusmod tempor incididunt ut labore et
                dolore magna aliqua. Ut enim ad minim veniam, quis
                nostrud exercitation ullamco laboris nisi ut aliquip
                ex ea commodo consequat. Duis aute irure dolor in
                reprehenderit in voluptate velit esse cillum dolore eu
                fugiat nulla pariatur.</p>
            </main>
        </div>
```

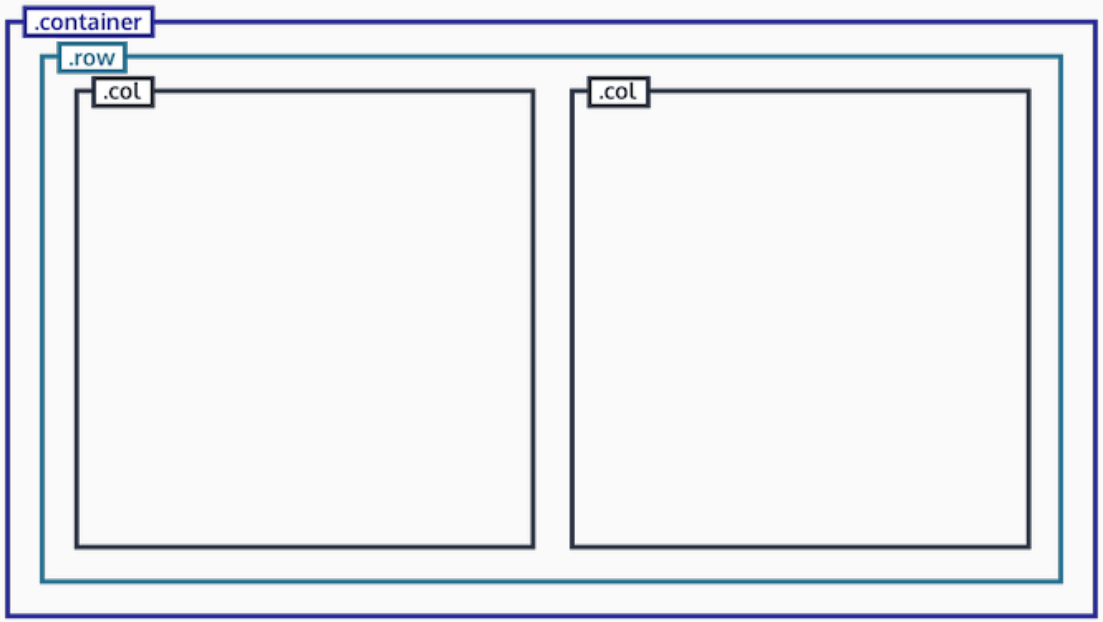
Exemple

Rendu



Colonnes

Principe
Au sein d'un *container*, pour placer les éléments sur la grille, on utilisera les classe **row** et **col**, qui permettent respectivement de définir des lignes et des colonnes



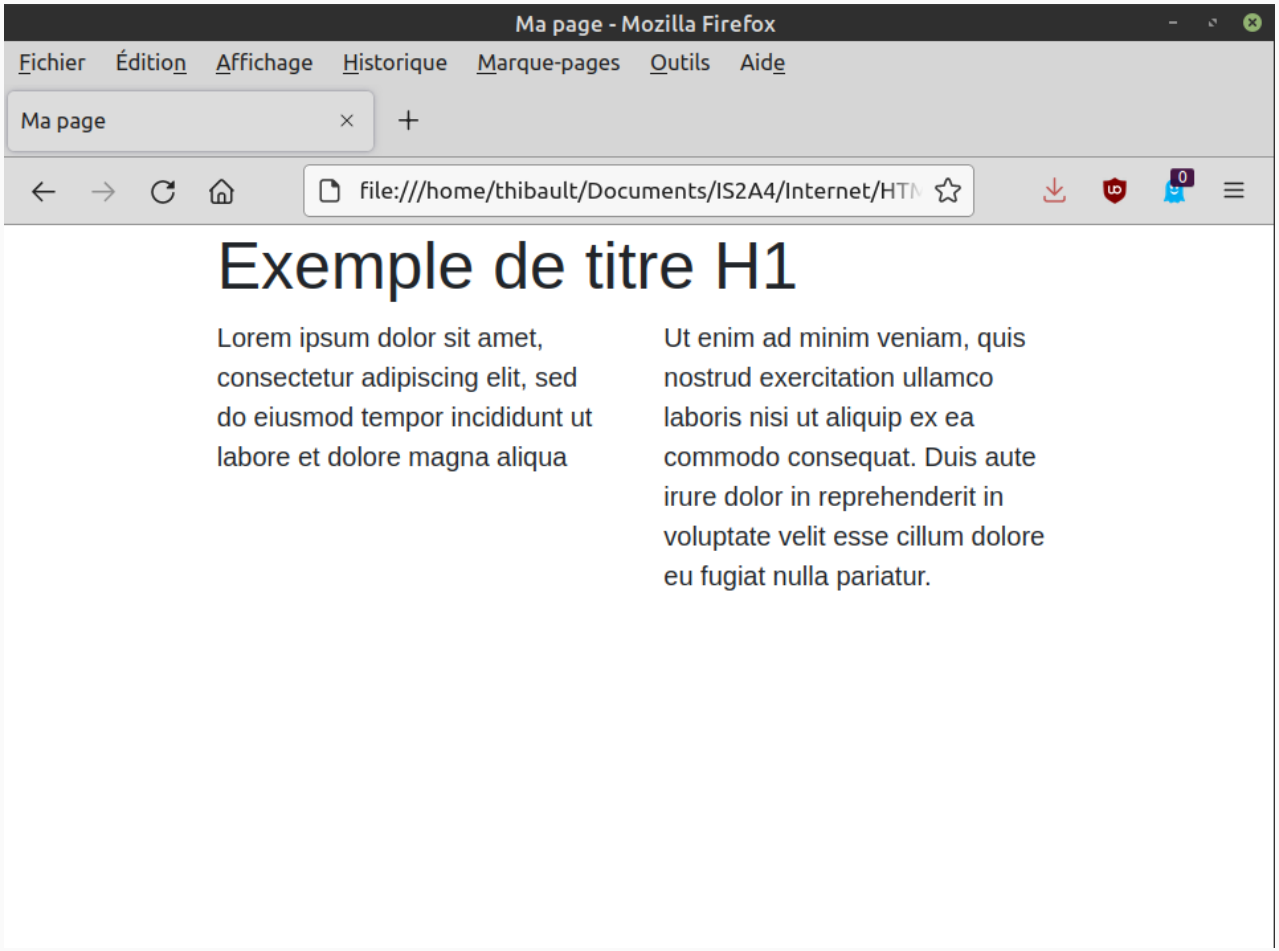
Exemple

HTML

```
<div class="container">
    <main>
        <div class="row">
            <div class="col">
                <h1>Exemple de titre H1</h1>
            </div>
        </div>
        <div class="row">
            <div class="col">
                <p>Lorem ipsum dolor sit amet, consectetur
                adipiscing elit, sed do eiusmod tempor
                incididunt ut labore et dolore magna aliqua</p>
            </div>
            <div class="col">
                <p>Ut enim ad minim veniam, quis nostrud
                exercitation ullamco laboris nisi ut aliquip
                ex ea commodo consequat. Duis aute irure dolor
                in reprehenderit in voluptate velit esse
                cillum dolore eu fugiat nulla pariatur.</p>
            </div>
        </div>
    </main>
```

Exemple

Rendu



Limite des colonnes

col-n
Il est possible d'utiliser une classe particulière de colonne : **col-n**, où **n** est un chiffre entre 1 et 12. En effet, la grille Bootstrap compte par défaut 12 colonnes. On peut ainsi choisir sur combien de colonne de la grille s'étend une *div* de classe *col*.

Fonctionnement par défaut
Par défaut, les colonnes se répartiront uniformément sur la grille. Si le compte ne tombe pas juste (par exemple 5 colonnes) une colonne passera à la ligne suivante.

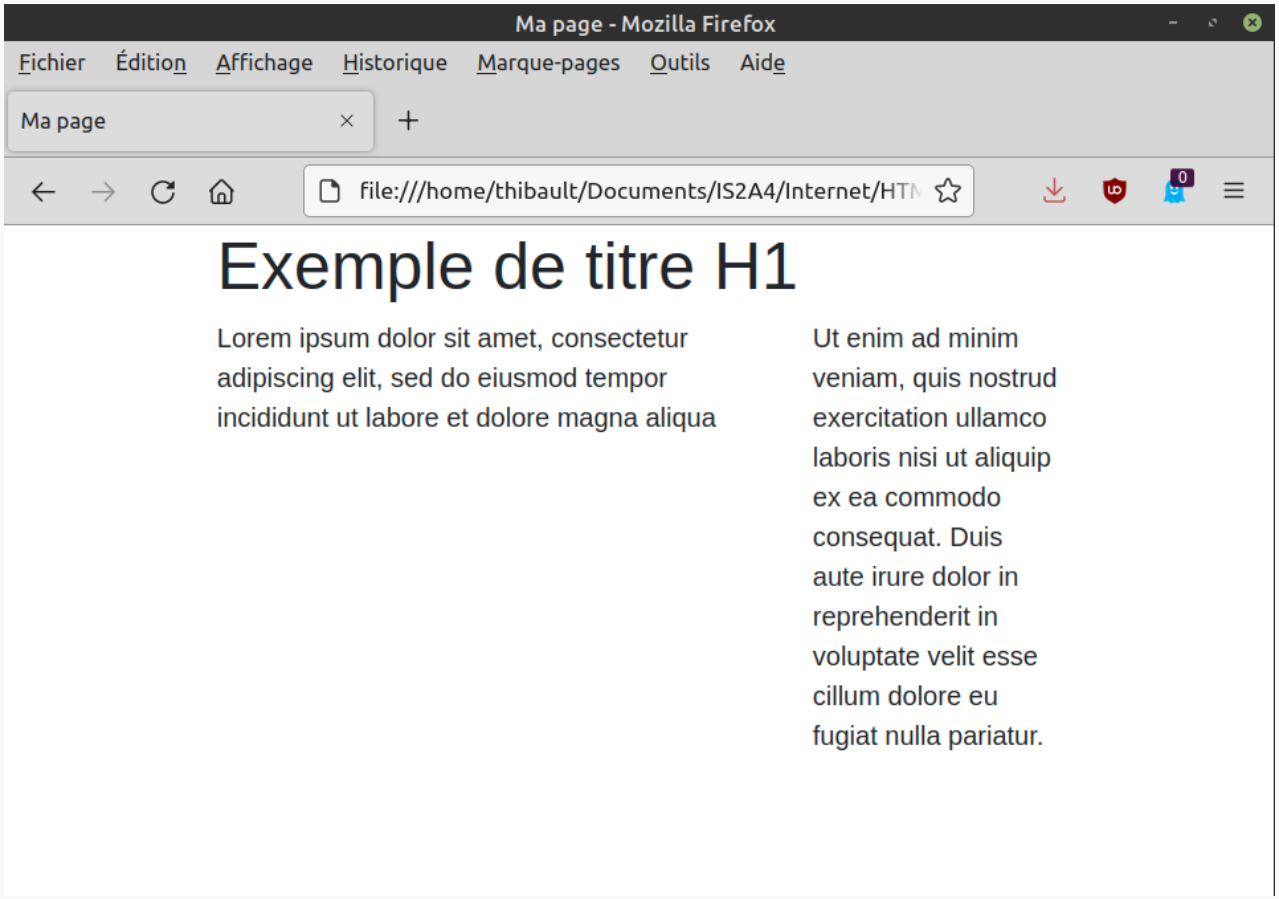
Exemple

HTML

```
<div class="container">
    <main>
        <div class="row">
            <div class="col">
                <h1>Exemple de titre H1</h1>
            </div>
        </div>
        <div class="row">
            <div class="col-8">
                <p>Lorem ipsum dolor sit amet, consectetur
                adipiscing elit, sed do eiusmod tempor
                incididunt ut labore et dolore magna aliqua</p>
            </div>
            <div class="col-4">
                <p>Ut enim ad minim veniam, quis nostrud
                exercitation ullamco laboris nisi ut aliquip
                ex ea commodo consequat. Duis aute irure dolor
                in reprehenderit in voluptate velit esse
                cillum dolore eu fugiat nulla pariatur.</p>
            </div>
        </div>
    </main>
</div>
```

Exemple

Rendu

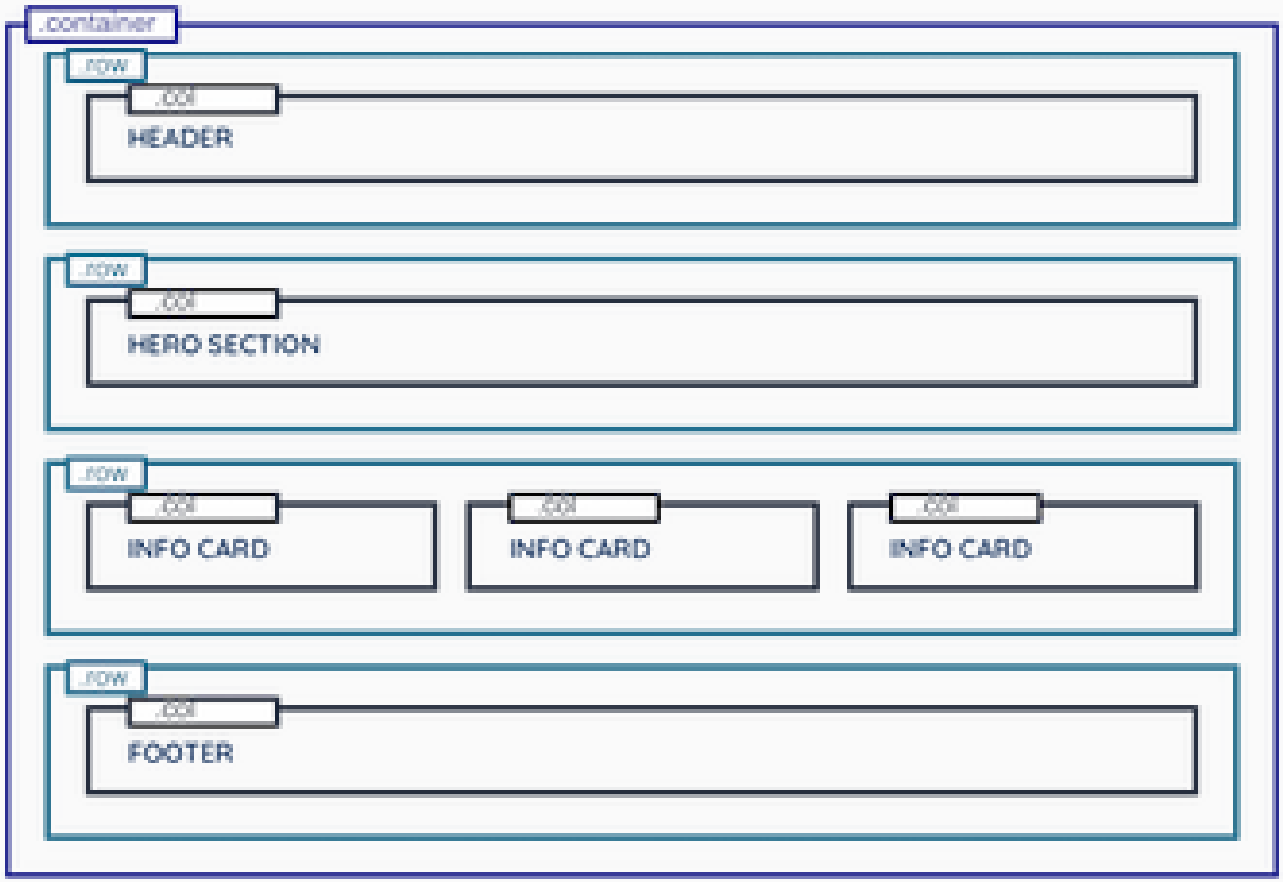


Section hero

Principe
Une section de classe *hero* avec Bootstrap sera une section qui sera particulièrement mise en avant par rapport au reste. Souvent placée en tête de page, elle comporte les informations principales que les visiteur verra directement en arrivant sur la page.

Design
Par défaut, Bootstrap ne définit aucun design par défaut pour une section *hero*. l'API fournit juste un environnement qui permet de définir un CSS différencié.

Composition d'une page



Design Responsive

Quoi qu'est-ce ?

Historique
Aux débuts de la conception Web, les pages étaient construites pour cibler une taille d'écran particulière. Si l'utilisateur avait un écran plus grand ou plus petit que celui du concepteur, les résultats attendus pouvaient aller de barres de défilement indésirables, à des longueurs de lignes trop longues, et à une mauvaise utilisation de l'espace. À mesure que des tailles d'écran plus variées devenaient disponibles, le concept de *responsive web design* (RWD) est apparu, un ensemble de pratiques qui permet aux pages Web de modifier leur disposition et leur apparence pour s'adapter à différentes largeurs d'écran, résolutions, etc.

Gestion par Bootstrap

Principe
Il n'est pas nécessaire de créer un CSS différent pour chaque taille d'écran. Bootstrap propose nativement des classes permettant de définir des designs qui s'adaptent à plusieurs taille d'écran.

Fonctionnement
En CSS, il est tout à fait possible pour un élément d'avoir plusieurs classe. C'est ce que propose de faire Bootstrap, en adaptant notamment le nombre de colonne de chaque élément en fonction de la taille de l'écran.

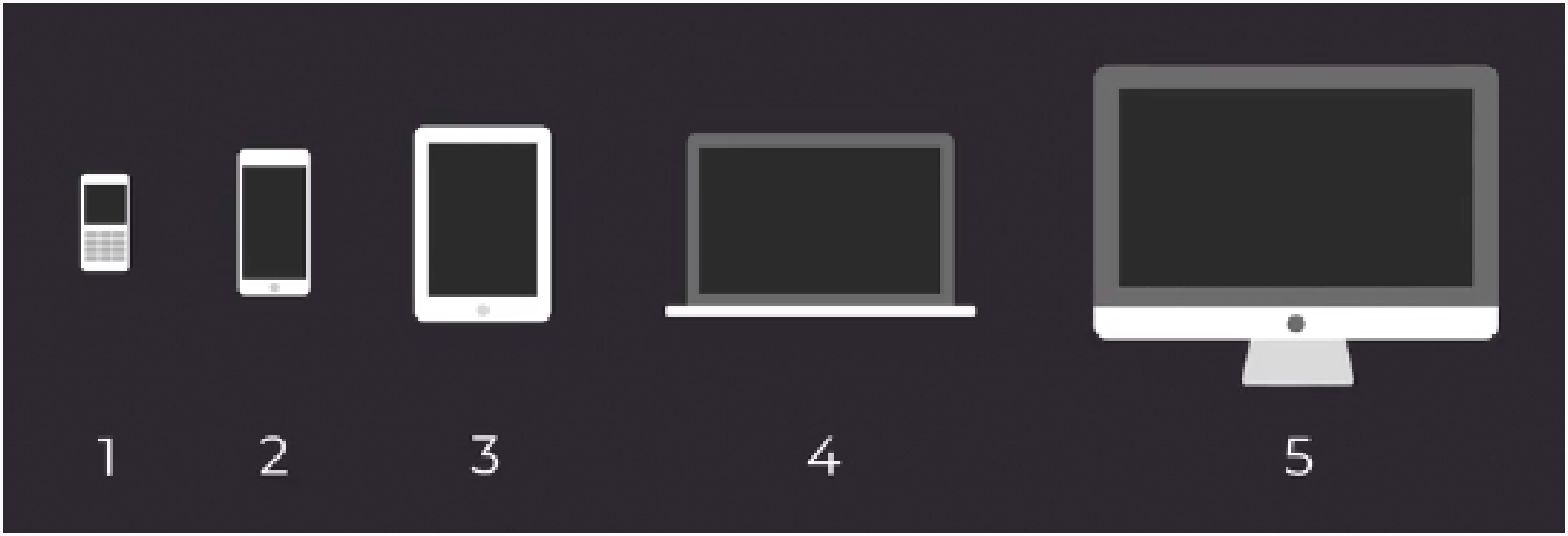
Différentes tailles

Principe
Il existe des modificateurs de classe qui permettent de définir pour quelle taille d'écran une classe est applicable.

Liste

Taille	Point d'arrêt	Modificateur
Très petit	576px	aucun
Petit	$\geq$ 576 px	-sm
Moyenne	$\geq$ 768 px	-md
Grande	$\geq$ 992 px	-lg
Très grande	$\geq$ 1200 px	-xl

Taille d’écran



Mobile first

Explication
Bootstrap est une API orientée *mobile first*, c'est à dire à destination prioritairement des mobiles, donc des petits écrans. C'est la raison pour laquelle il n'existe pas de modificateur pour la taille la plus petite, car c'est la taille standard.

Modificateur de colonne

Syntaxe
On peut dériver des classe *col-n* vue précédemment des classes modifiées pour un design responsive. Par exemple *col-md-8* pour dire que la colonne aura une largeur de 8 (sur 12) sur l'écran est de taille moyenne.

Compréhension

Cet élément prendra toute la largeur de la page sur les écran les plus petits, seulement 2/3 de la page sur les écrans entre 768 et 992px, et la moitié de la largeur au delà de 992px.

Ça dépasse

```
<div class="col-12 col-md-8 col-lg-6">
...
</div>
<div class="col-12 col-md-8 col-lg-6">
...
</div>
```

Problème
Sur les écrans les plus petits, cela est supposé afficher sur la première ligne 2 éléments de largeur 12 colonnes. Impossible ?

Résolution
En cas de débordement, Bootstrap ajoute automatiquement une nouvelle ligne.

Exemple

HTML

```
<div class="container">
    <main>
        <div class="row">
            <div class="col">
                <h1>Exemple de titre H1</h1>
            </div>
        </div>
        <div class="row">
            <div class="col-12 col-md-6 col-lg-4">
                <p>Lorem ipsum dolor sit amet, consectetur
                adipiscing elit, sed do eiusmod tempor
                incididunt ut labore et dolore magna aliqua</p>
            </div>
            <div class="col-12 col-md-6 col-lg-8">
                <p>Ut enim ad minim veniam, quis nostrud
                exercitation ullamco laboris nisi ut aliquip
                ex ea commodo consequat. Duis aute irure dolor
                in reprehenderit in voluptate velit esse
                cillum dolore eu fugiat nulla pariatur.</p>
            </div>
        </div>
    </main>
</div>
```

Design One-Page

Définition
En général, les sites Web sont conçus de la même manière qu'un média imprimé, puisque l'utilisateur qui recherche des informations et du contenu navigue de page en page

Un site *one page* remet en question cette notion de saut d'une page à l'autre et choisit une autre structure, puisque le site se limite à une seule page. On utilise alors en général différentes sections les unes au dessus des autres.

Scrollspy

Principe
Définir un outil Javascript permettant le suivi du parcours de la page. Très utile pour du design *one page*

`https://getbootstrap.com/docs/5.1/components/scrollspy/`

Éléments de design

Barre de navigation

La classe Bootstrap
Bootstrap propose une classe **navbar** qui permet d'implémenter un composant extrêmement efficace de barre de navigation. Il est fournit avec tous les modificateurs nécessaires pour en changer le style et le rendre responsive.

Lien avec HTML5
La classe *navbar* est une classe de l'élément *nav* définit par HTML5.

Classes liées

- nav-item
- nav-link
- active
- navbar-brand

Éléments de navigation

navitem
Les différents éléments de la barre de navigation sont définis avec la classes **nav-item**. Cette classe est définit pour des items de liste (élément <*li*>) d'une liste non ordonnées (élément <*ul*>) à laquelle on attribut la classe **navbar-nav**.

navbar-brand
Sert en général à définir, en en-tête d'une barre de navigation, la classe d'un élément lien <*a*> qui reprend le logo du site et pointe vers l'accueil.

nav-link
Pour tous les liens contenus dans des items de navigation

active
Permet éventuellement de mettre en avant l'élément de navigation correspondant à la page courante (CSS à définir).

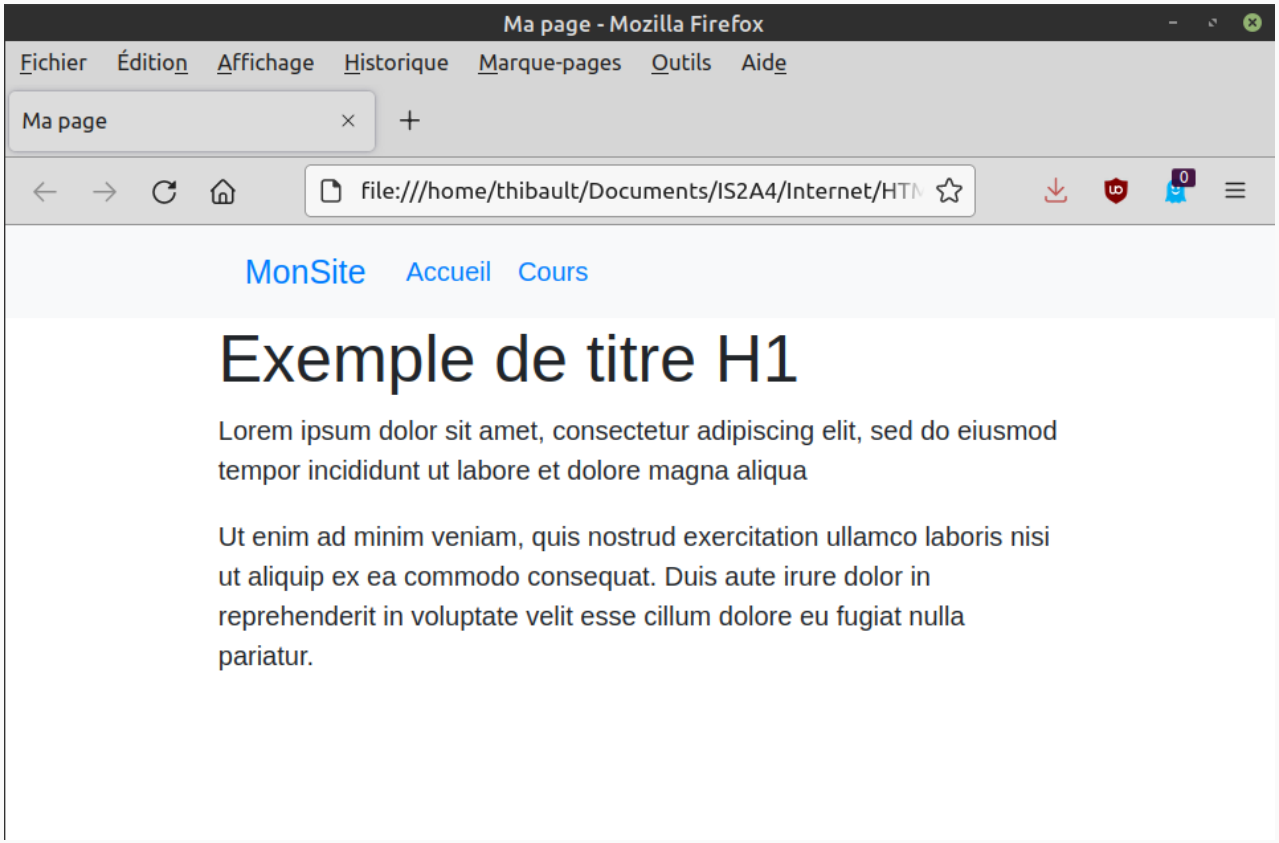
Exemple

HTML

```
<body>
    <div class="bg-light">
        <div class="container">
            <header>
                <nav class="navbar navbar-expand">
                    <a class="navbar-brand" href="#">MonSite</a>
                    <ul class="navbar-nav">
                        <li class="nav-item active">
                            <a class="nav-link" href="index.html">
                            Accueil</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="#">Cours</a>
                        </li>
                    </ul>
                </nav>
            </header>
        </div>
    </div>
```

Exemple

Rendu



Rendre responsive

Comment ?
Le principe est simple : on veut qu'en dessous d'une certaine taille d'écran, la barre de navigation disparaisse pour laisser place à un bouton permettant d'afficher un menu déroulant avec les différents liens.

Mobile first
Souvenez vous du concept *mobile first* de Bootstrap : on considère les tailles de la plus petite à la plus grande. Il serait plus juste de dire qu'on veut *qu'à partir* d'une certaine taille d'écran, on veut que le bouton laisse place à la barre de navigation entière.

Expand

navbar-expand
Il s'agit d'une extension de la classe *navbar* qui lui permet de s'étendre au delà d'une certaine taille. Attention, il faut obligatoirement appliquer un modificateur de classe pour la taille, sinon cela s'appliquera pour tous les écrans.

Exemple
<nav class="navbar navbar-expand-lg">
Cela permet à la barre de navigation de s'étendre au delà de 992px.

Le bouton

Position
On situe le code de ce bouton entre le lien *navbar-brand* et la liste *navbar-nav*

Code

```
<button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="
#navbarContent">
    <span class="navbar-toggler-icon"></span>
</button>
```

Le bouton est de classe **navbar-toggler** (littéralement, "bascule de barre"). On note par ailleur que le *span* à l'intérieur ne sert qu'à introduire l'icône du bouton.

Collapse

Quoi qu'est-ce ?
Nous voyons apparaître la classe **collapse** sur le code précédent. Elle sert à définir le comportement du bouton, c'est à dire l'action qui sera appliquée (*data-toggle*) aux données visées par le bouton (*data-target*).

Une div supplémentaire
Pour que les données visées soit les bonnes, il devient nécessaire de rajouter une *div* dont l'id sera *navbarContent* et dont les classes seront *collapse* et *navbar-collaspe*.

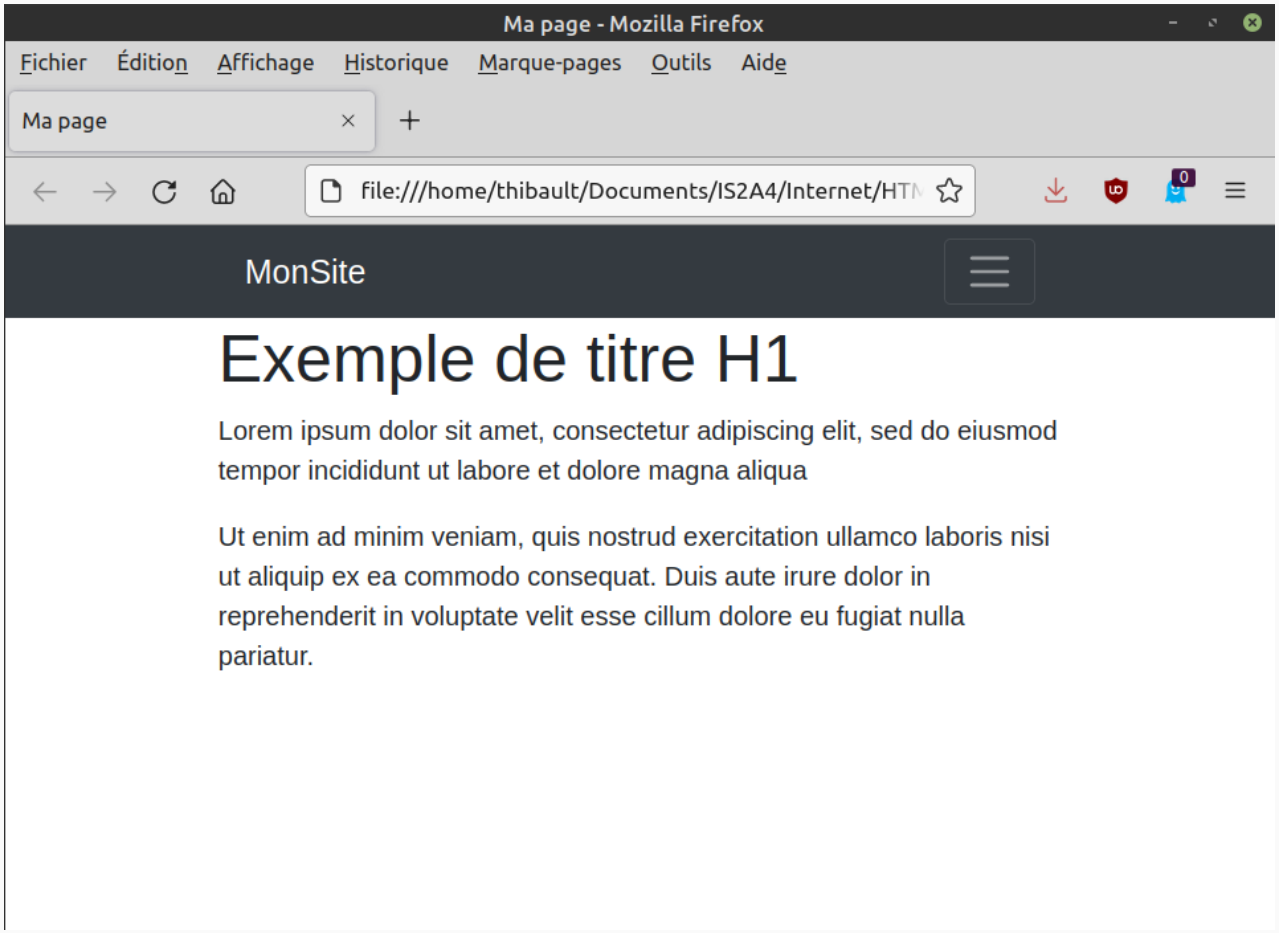
Exemple

HTML

```
<div class="bg-dark">
    <div class="container">
        <header>
            <nav class="col navbar navbar-lg-expand navbar-dark">
                <a class="navbar-brand" href="#">MonSite</a>
                <button class="navbar-toggler" type="button" data-toggle=
                "collapse" data-target="#navbarContent">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div id="navbarContent" class="collapse navbar-collapse">
                    <ul class="navbar-nav">
                        <li class="nav-item active">
                            <a class="nav-link" href="#">Accueil</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="#">Cours</a>
                        </li>
                    </ul>
                </div>
            </nav>
        </header>
    </div>
</div>
```

Exemple

Rendu



Design de barre

Différents design
Vous aurez peut être noté les classes *bg-light* ou *bg-dark* qui apparaissent dans le code. Cela permet de définir différent design de barre.

Autre solution
On peut aussi utiliser l'attribut *style* de l'élément *nav*.

<nav class="col navbar navbar-lg-expand navbar-dark"
style="background-color:#e32020">

Mise en avant

Pourquoi ? Comment ?
Il peut être nécessaire de mettre en avant des éléments d'un site, et ce pour plein de raison. En réalité, la multiplicité des raisons fait qu'il peut y avoir énormément de façon différentes de mettre en avant du contenu de façon pertinente.

Jumbotron

Pas le hero que nous méritons, mais celui dont on a besoin

La classe *Jumbotron* est une sous classe de la classe *hero* dont nous avons parlé plus haut. Elle est dédié à la mise en avant du contenu en tête de page. Bien sûr on pourrait l'utiliser pour mettre en valeur d'autres types de contenus, mais cela serait s'exposer à des effets de bors indésirables (ce qui arrive toujours quand on bidouille avec des choses qui ne sont pas faites pour ce qu'on veut réaliser).

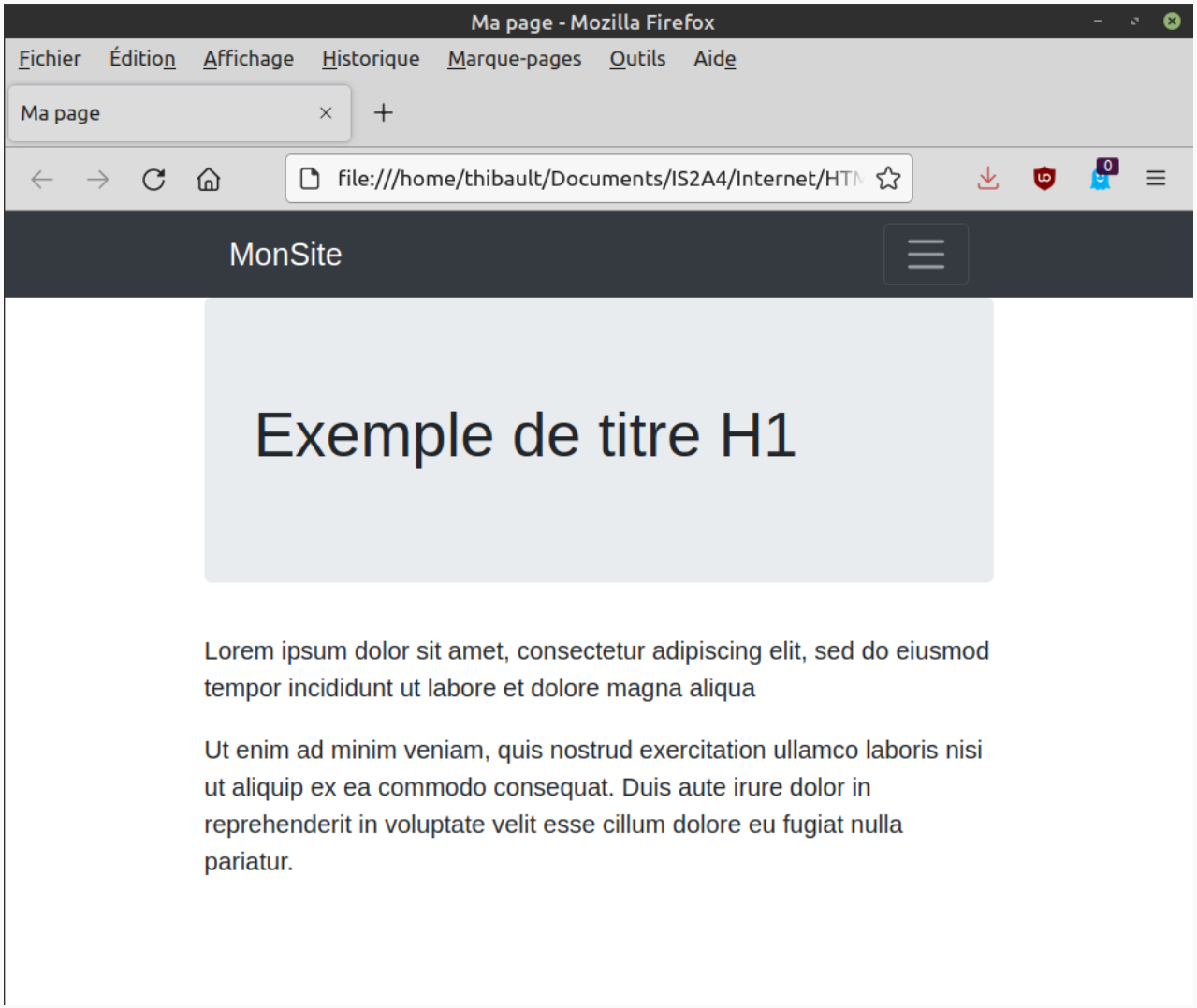
Exemple

HTML

```
<div class="jumbotron">
    <div class="row">
        <div class="col">
            <h1>Exemple de titre H1</h1>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-12 col-md-6 col-lg-4">
        <p>Lorem ipsum dolor sit amet, consectetur
        adipiscing elit, sed do eiusmod tempor
        incididunt ut labore et dolore magna aliqua</p>
    </div>
    <div class="col-12 col-md-6 col-lg-8">
        <p>Ut enim ad minim veniam, quis nostrud
        exercitation ullamco laboris nisi ut aliquip
        ex ea commodo consequat. Duis aute irure dolor
        in reprehenderit in voluptate velit esse
        cillum dolore eu fugiat nulla pariatur.</p>
    </div>
</div>
```

Exemple

Rendu



Card

Principe
Le composant *card* est un conteneur flexible qui permet de mettre en forme du contenu en le distingant du reste de la page par une bordure, et qui comporte un en-tête, un pied-de-page et un corps.

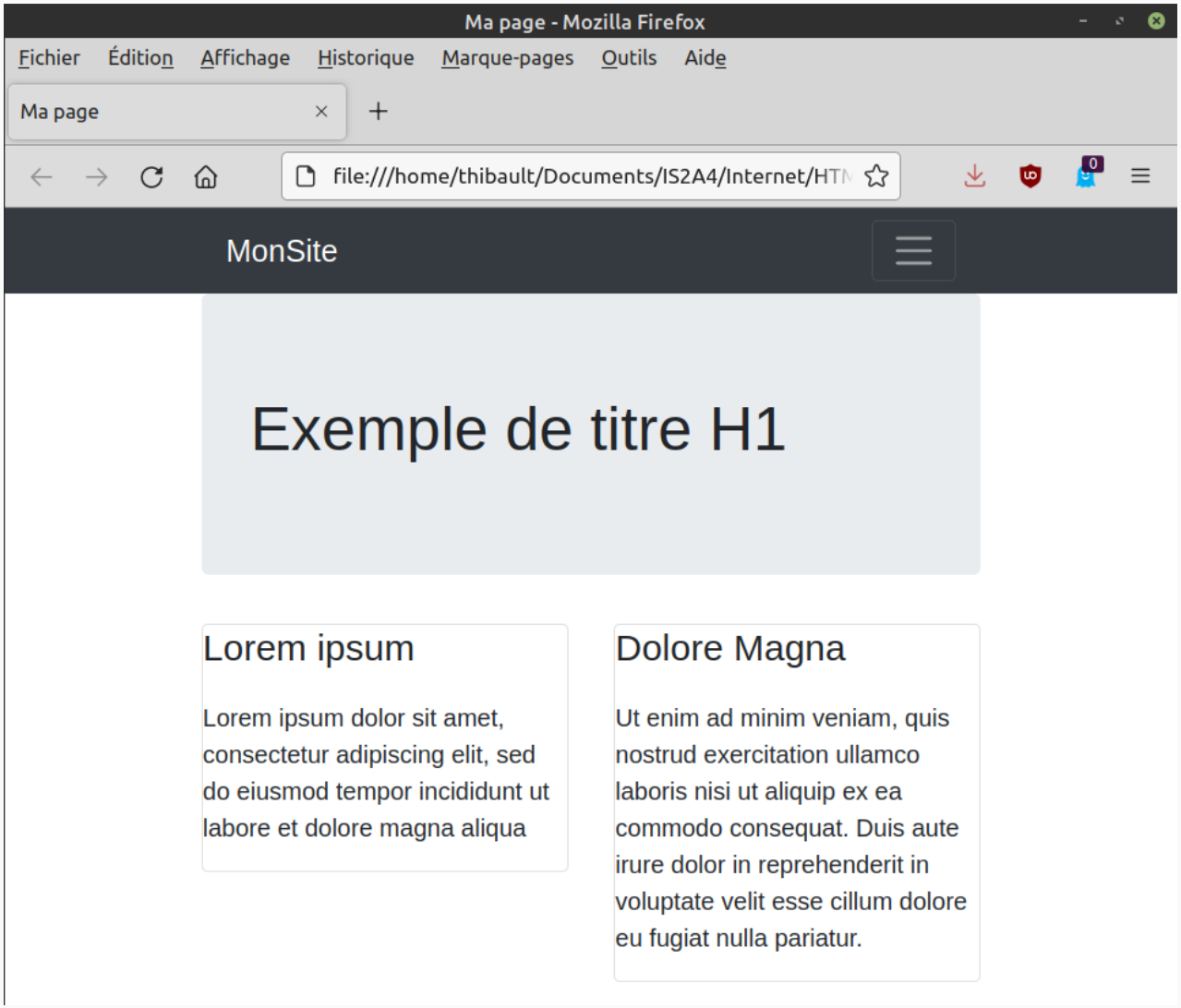
Exemple

HTML

```
<div class="row">
    <div class="col">
        <div class="card">
            <div class="card-title">
                <h4>Lorem ipsum</h4>
            </div>
            <div class="card-bordy">
                <p>Lorem ipsum dolor sit amet, consectetur adipiscing
                elit, sed do eiusmod tempor incididunt    ut labore et
                dolore magna aliqua</p>
            </div>
        </div>
    </div>
    <div class="col">
        <div class="card">
            <div class="card-title">
                <h4>Dolore Magna</h4>
            </div>
            <div class="card-bordy">
                <p>Ut enim ad minim veniam, quis nostrud exercitation
                ullamco laboris nisi ut aliquip ex ea commodo consequat.
                Duis aute irure dolor in reprehenderit in voluptate
                velit esse cillum dolore eu fugiat nulla pariatur.</p>
            </div>
        </div>
    </div>
</div>
```

Exemple

Rendu



Lead

Principe
La classe **lead** s'applique à un élément paragraphe et permet de le mettre en avant, par exemple en tête d'une section ou d'un article. En général, elle s'applique au paragraphe qui sert d'introduction ou de chapeau.

Exemple

HTML

```
<div class="row">
    <p class="lead">Lorem ipsum dolor sit amet,
    consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore magna aliqua</p>
</div>
<div class="row">
    <div class="col">
        <p>Lorem ipsum dolor sit amet, consectetur
        adipiscing elit, sed do eiusmod tempor
        incididunt    ut labore et dolore magna aliqua
        </p>
    </div>
    <div class="col">
        <p>Ut enim ad minim veniam, quis nostrud
        exercitation ullamco laboris nisi ut aliquip
        ex ea commodo consequat. Duis aute irure dolor
        in reprehenderit in voluptate velit esse
        cillum dolore eu fugiat nulla pariatur.</p>
    </div>
</div>
```

Exemple

Rendu

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Alert

Principe

Le composant **alert** sert à mettre en avant un texte qui vient ponctuellement prévenir l'utilisateur de quelque chose. C'est par définition quelque chose qui sera lié à un événement. et qui disparaîtra ensuite.

Il existe nativement plusieurs type qui sont détaillés dans la slide suivante.

Syntaxe

```
<div class="alert alert-primary" role="alert">
    This is a primary alert—check it out!
</div>
```

Alert

This is a primary alert—check it out!

This is a secondary alert—check it out!

This is a success alert—check it out!

This is a danger alert—check it out!

This is a warning alert—check it out!

This is a info alert—check it out!

This is a light alert—check it out!

This is a dark alert—check it out!

Buttons

Principe

De manière très similaire à ce qui existe pour *alert*, il existe des designs prédéfinis pour les boutons grâce aux classes **btn** et **btn-...** (avec '...' un style prédéfini).

Syntaxe

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-link">Link</button>
```

Rendu

Button group

Principe

Il est possible de créer des groupes de boutons, ce qui peut être utile pour des raisons de sémantique ou d'esthétique.

Syntaxe

```
<div class="btn-group" role="group" aria-label="Basic example">
  <button type="button" class="btn btn-secondary">&lt;</button>
  <button type="button" class="btn btn-secondary">^</button>
  <button type="button" class="btn btn-secondary">&gt;</button>
</div>
```

Rendu

Collapse

Principe
Nous avons vu comment fonctionne *collapse* avec les barres de navigation. En fait, *collapse* peut s'utiliser sur bien d'autres éléments. Il peut être utile de créer une div de classe *collapse* pour, par exemple, le l'afficher que lors d'une certaine action.

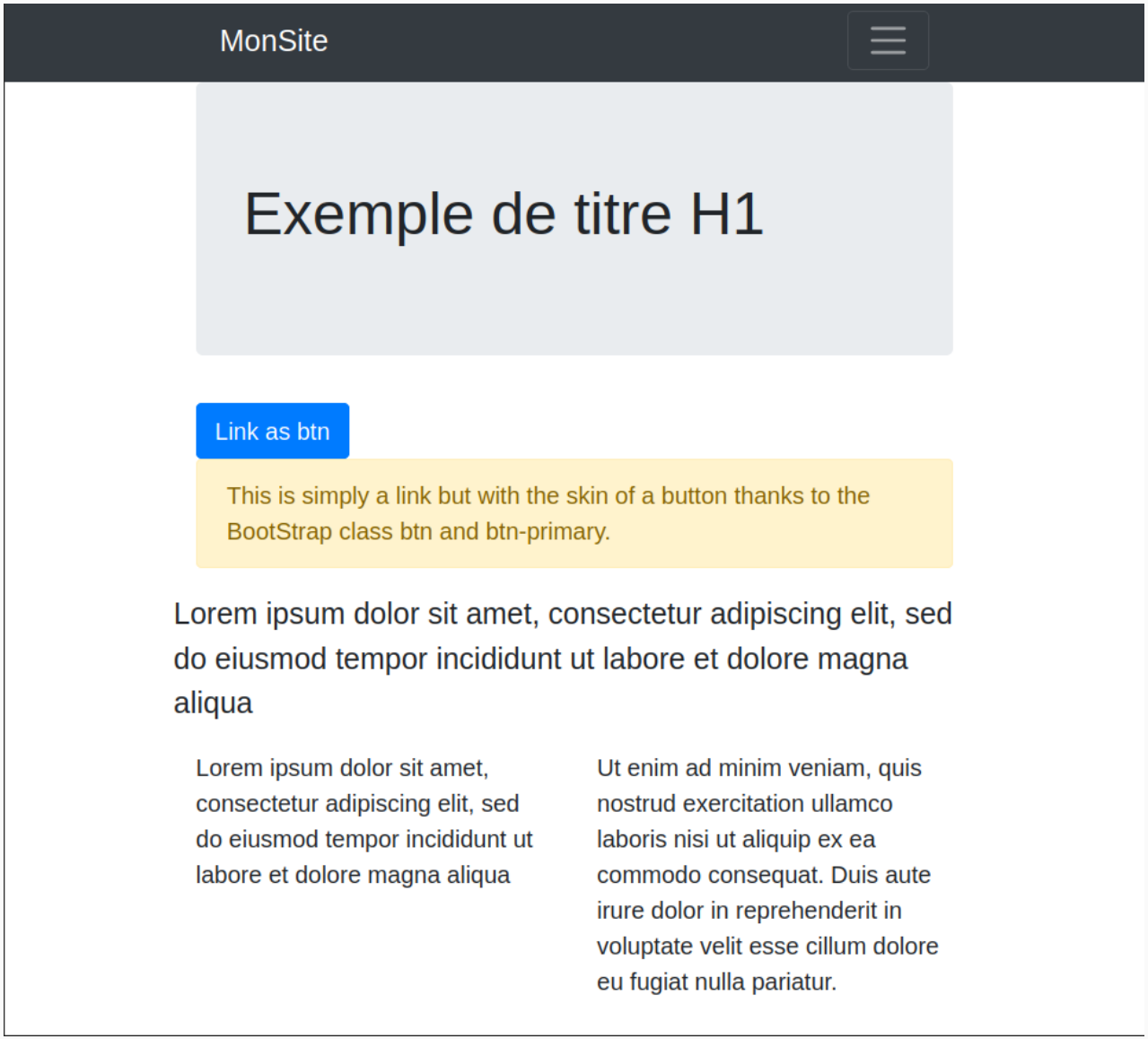
Exemple

HTML

```
<a class="btn btn-primary" data-toggle="collapse" href="#collapseExample"
 role="button" aria-expanded="false" aria-controls="collapseExample">
    Link as btn
</a>
<div class="collapse" id="collapseExample">
    <div class="alert alert-warning">
        This is simply a link but with the skin of a button thanks to
        the BootStrap class btn and btn-primary.
    </div>
</div>
```

Exemple

Rendu



Progress Bar

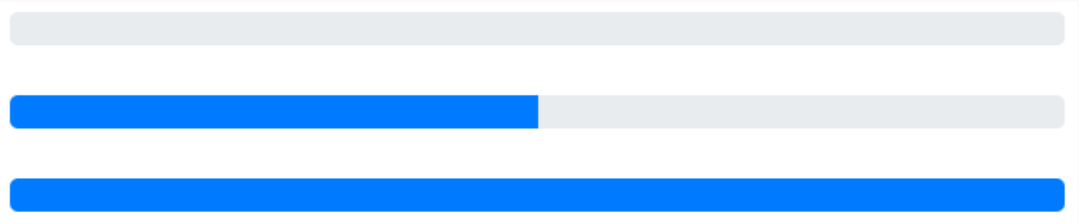
Principe

Une façon très simple de montrer une progression, que ça soit en fixe sur un CV ou en dynamique pour la progression dans un formulaire par exemple.

Syntaxe

```
<div class="progress">
    <div class="progress-bar" role="progressbar" style="width: 0%"
    aria-valuenow="0" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress">
    <div class="progress-bar" role="progressbar" style="width: 50%"
    aria-valuenow="50" aria-valuemin="0" aria-valuemax="100"></div>
</div>
<div class="progress">
    <div class="progress-bar" role="progressbar" style="width: 100%"
    aria-valuenow="100" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

Rendu



Typographie

Alignement de texte

Principe
Il est possible de modifier l'alignement du texte dans un élément simplement en utilisant les classes Bootstrap dont le CSS a déjà été définit.

Classes

- *text-left*
- *text-center*
- *text-right*

Responsive
Cet alignement peut s'adapter à la taille de l'écran utilisé en utilisant les modificateurs de classes déjà vus précédemment. On peut par exemple écrire *text-sm-left* ou *text-md-center*.

Gras et italique

Graisse
Il existe différentes épaisseurs de trait nativement inclues dans Bootstrap, qu'on appelle en utilisant une classe dans un élément *div* ou *span* par exemple :

- font-weight-bolder
- font-weight-bold
- font-weight-normal
- font-weight-light
- font-weight-lighter

Italique
Tout simplement *font-italic*.

Liste inline

Utilité

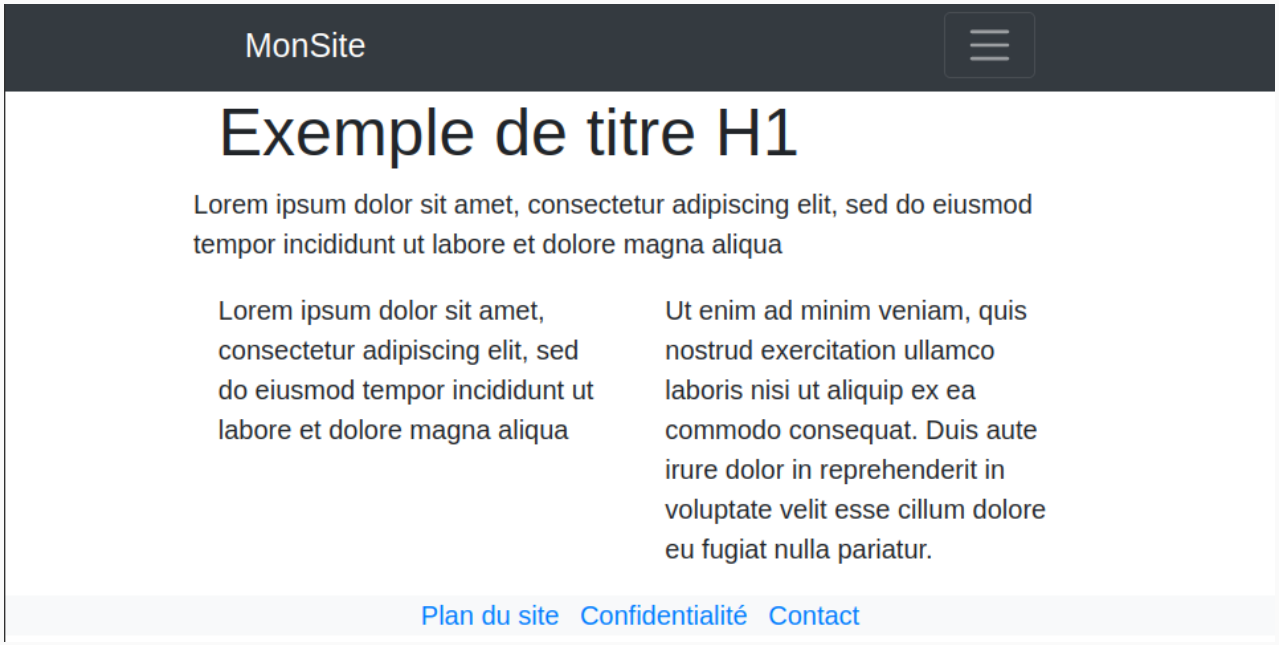
La classe **liste-inline** de Bootstrap permet de représenter des listes sur une seule ligne, très pratique par exemple pour les liens d'un *footer*.

Syntaxe

```
<div class="bg-light text-center">
    <div class="container">
        <footer>
            <ul class="list-inline">
                <li class="list-inline-item"><a href="#">Plan du site</a></li>
                <li class="list-inline-item"><a href="#">Confidentialité</a></li>
                <li class="list-inline-item"><a href="#">Contact</a></li>
            </ul>
        </footer>
    </div>
</div>
```

Exemple

Rendu



Citation

Exemple

> A well-known quote, contained in a blockquote element.
>
> — Someone famous in *Source Title*

```
<figure>
  <blockquote class="blockquote">
    <p>A well-known quote, contained in a blockquote element.</p>
  </blockquote>
  <figcaption class="blockquote-footer">
    Someone famous in <cite title="Source Title">Source Title</cite>
  </figcaption>
</figure>
```

Pour en savoir plus

La doc Bootstrap !
`https://getbootstrap.com/docs/5.1/getting-started/introduction/`

This is it ☺