

Prenez en main Bootstrap

Par bestmomo



OPENCLASSROOMS

www.openclassrooms.com

*Licence Creative Commons 6 2.0
Dernière mise à jour le 6/03/2014*

Sommaire

Sommaire	2
Lire aussi	3
Prenez en main Bootstrap	5
Partie 1 : Premiers pas	5
Mise en route	6
Un framework ?	6
Les intérêts	6
Les inconvénients	6
Découverte de Bootstrap	7
Origine de Bootstrap	7
Contenu du kit	7
Évolution de Bootstrap	7
Installation	8
Installation de Bootstrap	8
Les CDN	11
Une trame pour démarrer	11
Le template de base	11
Un template de démarrage	13
Les Media Queries	15
Une grille ?	17
Présentation	18
Le principe d'une grille	18
Terminologie	18
La grille de Bootstrap	18
Organisation de la grille	19
La grille en pratique	25
Le conteneur	25
Une seule rangée	26
Plusieurs rangées	27
Sauter des colonnes	27
Imbrication d'éléments	29
Ordre des colonnes	30
Un petit TP	31
Mise en page	31
Premier cas	31
Second cas	33
Troisième cas	34
Quatrième cas	36
Un peu de pratique	40
Combiner les formats et exemple de page	40
Combinaison de classes « col-* »	40
Page d'exemple	43
Cas pratiques et classes « responsive »	49
Exercice 1	49
Exercice 2	56
Partie 2 : Mise en forme des éléments	60
Éléments de base	61
Listes, descriptions et tableaux	61
Les listes	65
Descriptions	66
Les tableaux	67
Formulaires	72
Formulaire de base	72
La page d'exemple	75
Formulaire horizontal	82
Formulaire en ligne	83
Les classes input-group, input-group-addon et input-group-btn à l'œuvre	84
Dimension des contrôles	86
Stylisation selon l'état	87
Un générateur de formulaire	90
Boutons, icônes et images	91
La page d'exemple	91
Des jolis boutons	95
De jolies icônes	101
Mise en forme des images	105
Les composants intégrés	106
La navigation	107
La page d'exemple	107
Une barre de navigation	108
La pagination	114
Aide à la navigation	117
Effets typographiques et thumbnails	118
La page d'exemple	118
Jumbotron	119

Des libellés colorés	120
Des badges	122
Thumbnails	123
Listes groupées, panneaux et médias	126
La page d'exemple	126
Les listes groupées	127
Les panneaux	132
Les médias	136
Alertes et barres de progression	139
Alertes	139
Barres de progression	142
Partie 3 : Mise en forme avancée	147
Les plugins jQuery	147
Généralités	147
La librairie jQuery	147
Les bibliothèques Javascript de Bootstrap	149
La page d'exemple	152
Menu déroulant	158
Menu déroulant dans la barre de navigation	158
Menu déroulant dans un bouton	159
Activation par Javascript	165
Fenêtre modale	166
Un premier exemple simple	166
Un pied de page	167
Une transition	168
Garder le fond de l'écran	169
Injecter une page HTML	169
Largeur de la fenêtre modale	172
La page d'exemple et l'accessibilité	173
Activation avec Javascript	174
Onglets	176
Un premier exemple simple	176
Un autre aspect	176
Transition progressive	176
Empilage	177
Justification	177
Désactiver un lien	178
Un menu déroulant dans un onglet	178
Onglets latéraux	179
La page d'exemple	180
Utilisation de Javascript	181
Boutons	183
Bouton bascule	183
Boutons effet "checkbox"	183
Boutons effet "radio"	183
Boutons pour attente de processus	184
La page d'exemple	185
Carrousel	186
La page d'exemple	186
Un premier exemple simple	186
Des titres pour les images	187
Un indicateur de l'image affichée	188
Des boutons de défilement	189
Améliorer la présentation	190
Les options du carrousel	191
Les événements	194
Info-bulles et Popover	195
La page d'exemple	195
Info-bulle	196
Popover	198
Effet accordéon	200
Un exemple simple	200
Des bordures pour bien délimiter les éléments	201
Afficher une seule zone	203
De jolies en-têtes	204
Effet accordéon sur une liste groupée	205
Commande en Javascript	206
Utilisation des événements	207
Une barre de navigation rétractable	209
Le Scrollspy	210
Mise en page avec le Scrollspy	210
Un mouvement fluide	214
Activation par Javascript	215
Configurer Bootstrap	215
Configuration simplifiée	216
Une page d'exemple	216
Configurer en ligne sur le site de Bootstrap	220
LESS, un langage de style dynamique	223
Les sources des fichiers LESS de Bootstrap	223
Des variables	224
Inclusion de fichier	225
Les mixins	225
Sélecteurs emboîtés	226

Opérations	227
Fonctions	228
Un exemple synthétique	228
Utilisation de LESS dans Bootstrap	231
Un programme pour LESS	231
Exemple simple de configuration	232
Réflexions sur la configuration de Bootstrap avec LESS	233
Les mixins	235
Utilisation dynamique de LESS	239
Un thème pour Bootstrap	240
Le thème optionnel	240
D'autres thèmes	241
Composer son thème	243

Prenez en main Bootstrap

Par [bestmomo](#)

Mise à jour : 06/03/2014

Difficulté : Intermédiaire  Durée d'étude : 20 jours



5 visites depuis 7 jours, classé 15/808

Vous créez des pages web et vous passez beaucoup (trop) de temps avec le CSS ?

Alors **Bootstrap** est fait pour vous, et ce cours va vous guider dans la découverte de cette puissante boîte à outils. **Bootstrap** est un framework qui peut rendre votre vie plus facile pour créer l'architecture d'une page web. Mais **Bootstrap** va bien plus loin qu'offrir du code **CSS** déjà bien organisé et structuré. Il offre aussi des plugins **jQuery** de qualité pour enrichir vos pages.

Vous êtes prêts ? Bon, on y va, mais il serait bien, pour profiter pleinement de ce cours, que possédiez quelques connaissances préalables :

- La base indispensable est de posséder de bonnes notions en **HTML** et **CSS**. Vous avez des lacunes ? Comblez-les avec [le cours de Mathieu Nebra](#).
- Pour comprendre la mise en œuvre des plugins **jQuery**, vous aurez besoin des quelques bases dans ce domaine. Il vous en manque ? Alors vous pouvez réparer ça en lisant [le cours de Michel Martin](#).
- Si vous voulez modifier **Bootstrap** pour l'adapter à vos besoins il vous faudra connaître **LESS**. Là aussi vous avez besoin d'un coup de main ? Il existe également [un cours](#).



Ce cours a été élaboré avec la version 3.1 de Bootstrap.

Allez on y va !

Partie 1 : Premiers pas

Dans cette première partie, nous allons voir ce qu'est un *framework* et en quoi il peut nous être utile. Nous allons voir comment installer **Bootstrap** et commencer à l'utiliser pour organiser une page web avec un système de *grille* simple et efficace.

Après une présentation générale et une illustration des principales possibilités de structuration de page avec **Bootstrap** nous verrons des cas pratiques d'application.

Mise en route

Pour utiliser efficacement Bootstrap, vous devez déjà être convaincu de son utilité, vous devez aussi savoir l'installer. Ce chapitre est destiné à vous faire découvrir en quoi Bootstrap peut vous aider et comment vous devez le mettre en œuvre pour l'utiliser efficacement. Nous aborderons également les Media Queries que Bootstrap utilise pour que nos pages s'adaptent automatiquement au support que nous utilisons pour les visualiser.

Un framework ?

Avant toute chose, il faut définir ce qu'est un framework. Il s'agit d'un ensemble de composants structurés qui sert à créer les fondations et à organiser le code informatique pour faciliter le travail des programmeurs, que ce soit en terme de productivité ou de simplification de la maintenance. Il en existe beaucoup pour les applications web qui ciblent de nombreux langages : Java, Python, Ruby, PHP...

Il existe des frameworks côté serveur (désignés *backend* en anglais), et d'autres côté client (désignés *frontend* en anglais). Bootstrap fait partie de cette deuxième catégorie. Les frameworks CSS sont spécialisés, comme leur nom l'indique, dans les CSS ! C'est-à-dire qu'ils nous aident à mettre en forme les pages web : organisation, aspect, animation... Ils sont devenus à la mode et il en existe un certain nombre :

- [elasticss](#)
- [Knacss](#)
- [Blueprint](#)
- [960 Grid System](#)
- [YUI](#)
- [52Framework](#)
- [inuitcss](#)
- [Elements CSS framework](#)
- [BlueTrip](#)
- [ez-css](#)
- etc.

Bootstrap est un framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation... Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées. Le framework le plus proche de Bootstrap est sans doute [Foundation](#) qui est présenté comme « *The most advanced responsive front-end framework in the world* ». Cette absence de modestie est-elle de mise ? Je pense que c'est surtout une affaire de goût et de préférence personnelle.

Les intérêts

- Les navigateurs sont pleins de fantaisies et ont des comportements très différents malgré leur lente convergence vers les standards. Les frameworks sont *cross-browser*, c'est à dire que la présentation est similaire quel que soit le navigateur utilisé et d'une parfaite compatibilité ;
- Les frameworks CSS font gagner du temps de développement parce qu'ils nous proposent les fondations de la présentation;
- Les frameworks CSS normalisent la présentation en proposant un ensemble homogène de styles ;
- Les frameworks CSS proposent en général une grille pour faciliter le positionnement des éléments ;
- Les frameworks CSS offrent souvent des éléments complémentaires : boutons esthétiques, barres de navigation, etc. ;
- La grande diffusion de nouveaux moyens de visualisation du web (smartphones, tablettes...) impose désormais la prise en compte de tailles d'écran très variées ; les frameworks CSS prennent généralement en compte cette contrainte.

Les inconvénients

- Pour utiliser efficacement un framework il faut bien le connaître, ce qui implique un temps d'apprentissage ;
- La normalisation de la présentation peut devenir lassante en lissant les effets visuels.

Par rapport aux deux inconvénients énoncés, Bootstrap est d'une prise en main rapide même pour un débutant et est totalement configurable parce qu'il est construit avec [LESS](#) (que nous verrons un peu plus loin dans le cours).

Découverte de Bootstrap

Origine de Bootstrap

Vous connaissez forcément Twitter, un des principaux réseaux sociaux qui inondent la planète de liens virtuels entre les humains devenus des noyaux cybernétiques. Le projet Bootstrap a été créé au départ par Mark Otto et Jacob Thornton pour répondre à des besoins internes de développement de cette entreprise au niveau de l'uniformisation de l'aspect des pages web. Il s'agissait juste de stylisation CSS, mais le framework s'est ensuite enrichi de composants Javascript.

Il a ensuite été publié en 2011 en devenant rapidement populaire parce qu'il est venu se positionner dans un espace vacant du développement. Son système de grille de 12 colonnes est devenu une référence. D'autre part sa mise en œuvre est aisée et se limite à référencer quelques librairies, comme nous allons bientôt le voir.

Il a été mis à disposition du public sous licence Apache. Le framework en est actuellement à la version 3. Celle-ci a pris un virage particulier en intégrant l'aspect « responsive » par défaut, alors qu'auparavant cette fonctionnalité faisait l'objet d'un fichier séparé. Cette version est même déclarée comme « mobile-first ». Avec l'utilisation croissante d'appareils mobiles, le framework s'est adapté pour offrir une solution censée couvrir tous les besoins.

Contenu du kit

Bootstrap propose :

- Une mise en page basée sur une grille de 12 colonnes bien pratique. Bien sûr, si vous avez besoin de plus de 12 colonnes, ou de moins, il est toujours possible de modifier la configuration ;
- L'utilisation de [Normalize.css](#) ;
- Du code fondé sur HTML5 et CSS3 ;
- Une bibliothèque totalement open source sous [licence Apache](#) ;
- Du code qui tient compte du format d'affichage des principaux outils de navigation (*responsive design*) : smartphones, tablettes... ;
- Des plugins jQuery de qualité ;
- Un résultat *cross-browser* (la prise en charge de IE7 a été abandonnée avec la version 3), donc une garantie de compatibilité maximale ;
- Une bonne documentation ;
- La garantie d'une évolution permanente ;
- Une mine de ressources variées sur le web ;
- Une architecture basée sur [LESS](#), un outil bien pratique qui étend les possibilités de CSS.

Vous pouvez trouver toutes les informations sur Bootstrap directement [sur le site dédié](#).



C'est quoi un [normalize.css](#) ?

Les navigateurs n'adoptent pas tous les mêmes valeurs par défaut pour les styles des éléments HTML. Cela peut générer quelques surprises au rendu des pages web selon le navigateur utilisé. D'autre part certains navigateurs présentent des défauts de prise en compte de certains éléments. [Normalize](#) est un petit fichier CSS qui établit des règles pour avoir un rendu identique quel que soit le navigateur utilisé. Au lieu d'agir brutalement comme les reset CSS qui remettent toutes les valeurs à zéro, [normalize](#) agit intelligemment en conservant ce qui est utile et en jouant finement sur les éléments. Vous pouvez trouver le détail des règles appliquées avec leur explication [ici](#).

Évolution de Bootstrap

Bootstrap est un framework très populaire qui évolue très rapidement avec l'arrivée fréquente de nouvelles versions. C'est à la fois un avantage (il s'améliore de plus en plus) et un inconvénient (le code qu'on a écrit pour une mise en page n'est rapidement

plus valable pour les nouvelles versions). Le passage à la version 3 a été une petite révolution avec de très nombreux changements, en particulier une refonte complète de la grille. Un site écrit avec la version 2 doit être totalement réécrit pour cette nouvelle version, mais ce n'est évidemment pas une obligation. L'évolution du framework s'est faite essentiellement en direction des appareils nomades qui constituent peu à peu le parc le plus important d'appareils pour surfer sur Internet.

La principale source d'information pour connaître les changements des nouvelles versions est [le blog officiel](#). Il est aussi intéressant de s'informer en amont sur [la page GitHub du projet](#) pour connaître les demandes des utilisateurs (Pull Requests) et les problèmes rencontrés (Issues).

Ce cours a été élaboré à partir de la version 3.1 et tous les renseignements et exemples donnés ont été testés sur cette version.

Installation

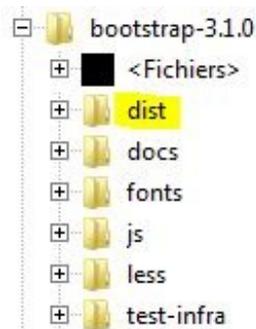
Installation de Bootstrap

L'installation de Bootstrap est simple : cliquez sur le bouton de téléchargement [sur le site du framework](#). Vous avez à disposition trois boutons :

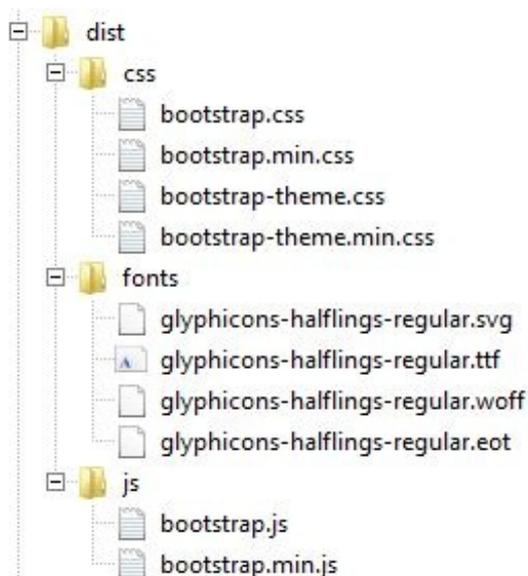
- "Download Bootstrap" : permet de récupérer juste les fichiers nécessaires au fonctionnement de Bootstrap,
- "Download source" : permet de récupérer en plus tous les fichiers source,
- "Download Sass" : c'est un portage de Bootstrap en Sass pour les utilisateurs de projets qui utilisent Sass (Rails, Compass...).

Vous pouvez aussi aller consulter directement le code source sur [github](#), ou même l'installer avec [Bower](#) si vous n'aimez pas la simplicité 🤖.

Quand vous téléchargez la librairie avec le bouton "Download source", vous obtenez un fichier zippé contenant un répertoire bootstrap-3.1.0, qui contient lui-même un certain nombre de fichiers et de dossiers, comme le montre la figure suivante.



Les fichiers utiles pour simplement utiliser Bootstrap se situent dans le dossier `dist` (« distribution »), ce sont les seuls fichiers que vous obtenez si vous utilisez le bouton "Download Bootstrap" :



Le dossier « dist » contient les fichiers utiles à l'utilisation de Bootstrap

Voyons un peu ces fichiers :

- `bootstrap.css` comporte les classes de base de Bootstrap ;
- `bootstrap.min.css` comporte les mêmes classes de base que `bootstrap.css` mais est compilé ;
- `bootstrap-theme.css` contient des règles de styles particulières pour créer un thème spécifique pour Bootstrap ;
- `bootstrap-theme.min.css` est identique à `bootstrap-theme.css` mais est compilé ;
- `glyphicons-halflings-regular.svg` comporte la collection d'icônes au format `svg` ;
- `glyphicons-halflings-regular.ttf` comporte la collection d'icônes au format `ttf` ;
- `glyphicons-halflings-regular.woff` comporte la collection d'icônes au format `woff` ;
- `glyphicons-halflings-regular.eot` comporte la collection d'icônes au format `eot` ;
- `bootstrap.js` contient le code JavaScript des composants de Bootstrap ;
- `bootstrap.min.js` contient le même code JavaScript mais est compilé ;

Les fichiers `min` (`bootstrap.min.css`, `bootstrap.min.js` et `bootstrap-theme.min.css`) contiennent le même code que leurs équivalents (`bootstrap.css`, `bootstrap.js` et `bootstrap-theme.css`) mais ont été épurés des commentaires et compilés pour les alléger et accélérer ainsi leur chargement.



D'accord, mais que mettre sur un site pour que Bootstrap fonctionne ?

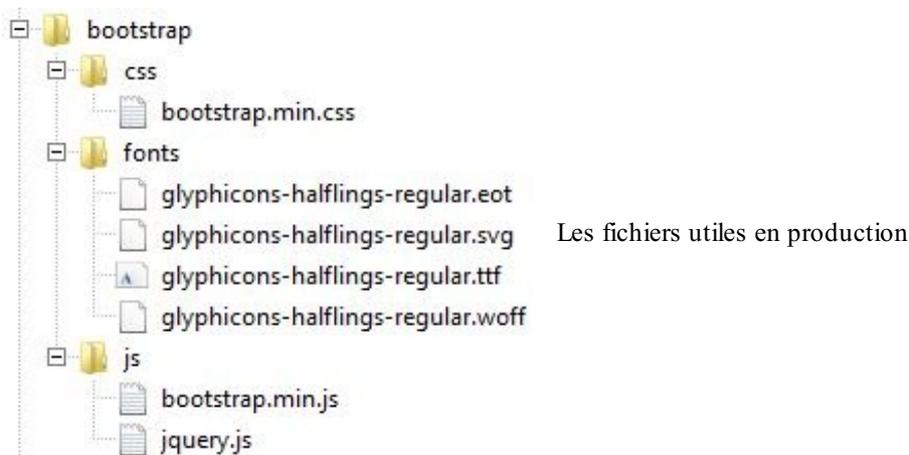
Il faut distinguer les deux situations :

1. Pendant la phase de développement, il est intéressant de pouvoir naviguer dans le code de Bootstrap, il faut donc mettre les fichiers non compressés.
2. Lorsque votre site est en ligne, vous n'avez plus à vous inquiéter du code et seule compte la vitesse de chargement, il faut donc cette fois utiliser les versions compressées (`min`).

La figure suivante montre les fichiers utiles pendant la phase de développement (les fichiers `bootstrap.js` et `jquery.js` ne sont à prévoir que si vous utilisez des plugins jQuery et le dossier `fonts` uniquement si vous utilisez les icônes).



La figure suivante montre les fichiers utiles lorsque le site est en ligne (les fichiers `bootstrap.min.js` et `jquery.js` ne sont à prévoir que si vous utilisez des plugins jQuery et le dossier `fonts` uniquement si vous utilisez les icônes).



Dans tous les exemples de ce cours, je pars du principe qu'un répertoire `bootstrap` a été créé à la racine du site avec les trois répertoires `css`, `fonts` et `js` comportant les fichiers cités précédemment.

Pour que Bootstrap fonctionne il faut le déclarer dans les pages HTML, qui doivent d'ailleurs être impérativement au format HTML 5, il faut donc prévoir le bon DOCTYPE :

Code : HTML

```
<!DOCTYPE html>
...
```

Il faut ensuite déclarer au minimum le fichier `bootstrap.min.css` (ou `bootstrap.css`) dans l'en-tête de la page web :

Code : HTML

```
<head>
...
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
</head>
```

À partir de là toutes les classes sont accessibles... Évidemment vous devez adapter le lien selon la localisation de vos fichiers.

Si vous utilisez des composants JavaScript, vous devez également référencer la librairie de Bootstrap ainsi que jQuery (la librairie jQuery ne fait pas partie des fichiers téléchargés avec Bootstrap et doit être récupéré indépendamment [sur http://jquery.com/](http://jquery.com/)) :

Code : HTML

```
<script src="bootstrap/js/jquery.js"></script>
<script src="bootstrap/js/bootstrap.min.js"></script>
```



Et le fichier de thème ?

Ce fichier est un exemple d'amélioration de l'esthétique de Bootstrap, il n'est pas utile pour son fonctionnement. Si vous observez l'apparence des composants de Bootstrap, vous allez constater qu'il sont « plats », sans aucun relief. Dans la version 2, ce n'était pas le cas. Pour des raisons de performances, il a été décidé d'offrir cette amélioration visuelle dans un fichier séparé. Si vous avez la curiosité de regarder ce fichier, vous verrez qu'il contient des définitions de dégradés et d'ombrages. Vous trouvez le résultat de l'application de ce thème sur [cette page d'exemple](#). Nous reparlerons de ce fichier dans le chapitre de la configuration de Bootstrap.

Les CDN

CDN est l'acronyme de « *Content delivery network* » ; c'est un réseau de serveurs qui met à disposition des librairies. Il devient ainsi inutile de prévoir ces librairies sur son propre serveur, il suffit de « pointer » vers eux. Il y a des avantages à utiliser un CDN :

- Libération de ressources et de la bande passante sur son propre serveur ;
- Parallélisation des téléchargements (un CDN est sur plusieurs serveurs) ;
- Accélération du chargement ;
- Diminution de la latence ;
- Actualisation automatique des librairies ;
- Amélioration du référencement...



Y-a-t-il des inconvénients ?

D'après certains, l'utilisation d'un CDN, qui impose une requête DNS supplémentaire, ne serait judicieuse que si l'on a beaucoup de librairies à charger. D'autre part si vous modifiez le fichier CSS de Bootstrap pour l'adapter à vos besoins, vous ne pourrez plus bénéficier des avantages d'un CDN. Mais si vous voulez profiter de cette possibilité pour Bootstrap, utilisez [bootstrapcdn](#). Vous pouvez évidemment utiliser un CDN également pour jQuery, par exemple [chez Google](#).

Évidemment si vous utilisez des CDN, l'installation en est d'autant simplifiée. Vous n'avez qu'à adapter les appels des librairies :

Code : HTML

```
<link href="//netdna.bootstrapcdn.com/bootstrap/3.1.0/css/bootstrap.min.css"
rel="stylesheet">
...
<script
src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
<script
src="//netdna.bootstrapcdn.com/bootstrap/3.1.0/js/bootstrap.min.js"></script>
```

Une trame pour démarrer Le template de base

Vous trouvez aussi sur le site de Bootstrap [un template de base](#) et des exemples comportant l'essentiel des éléments pour vous aider à initier un projet. Pour le moment vous n'allez pas encore comprendre l'utilité de tous ces éléments, mais vous y reviendrez par la suite lorsque vous aurez fait le tour de ce cours.

Dans le template de base, vous trouvez en premier cette ligne :

Code : HTML

```
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

Cette ligne concerne uniquement les mobiles. On demande que l'affichage occupe tout l'espace disponible avec une taille de 1, autrement dit sans zoom. Vous pouvez aller encore plus loin en interdisant le zoom ou en le limitant à certaines valeurs.

On trouve ensuite la déclaration du fichier CSS dans sa version minifiée :

Code : HTML

```
<link href="css/bootstrap.min.css" rel="stylesheet">
```

En fin de page (pour ne pas ralentir le chargement) se trouvent les appels JavaScript (utiles si vous utilisez des plugins jQuery, comme nous allons le voir dans ce cours) :

Code : HTML

```
<script src="https://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
```

Avec un test conditionnel, on trouve aussi des références :

Code : HTML

```
<!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// --
>
<!--[if lt IE 9]>
<script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
<script
src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
<![endif]-->
```

Le but de ces appels est de permettre aux navigateurs ne prenant pas en charge HTML5 et les Media Queries CSS3 de le faire ; ils visent essentiellement IE8. Pour obtenir plus de renseignements sur `respond.js` vous pouvez aller sur [la page GitHub](#). Méfiez-vous en particulier si vous utilisez un CDN pour charger vos styles CSS, parce que `respond.js` utilise AJAX et vous pouvez buter sur le problème d'accès à plusieurs domaines.

Le but du fichier `html5shiv` est de créer les éléments (avec `createElement`) de type bloc du HTML 5 (header, section, aside...) qui sont ignorés par IE8.

En résumé on obtient ce code (en simplifiant les références et en supposant que tous les fichiers sont situés dans un dossier `bootstrap` et des sous-dossiers fonctionnels) :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file://
-->
    <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
    <script
src="https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://code.jquery.com/jquery.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>
  </body>
```

```
</html>
```

Un template de démarrage

Vous pouvez trouver également sur le site 18 templates d'exemple. Les éléments nécessaires à la compréhension de ces exemples seront exposés tout au long de ce cours. Voyons toutefois le [premier exemple](#), qui est le plus simple :

Code : HTML

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="shortcut icon" href="../../docs-
assets/ico/favicon.png">

    <title>Starter Template for Bootstrap</title>

    <!-- Bootstrap core CSS -->
    <link href="../../dist/css/bootstrap.css" rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="starter-template.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements
and media queries -->
    <!--[if lt IE 9]>
    <script src="../../assets/js/html5shiv.js"></script>
    <script src="../../assets/js/respond.min.js"></script>
    <![endif]-->
  </head>

  <body>

    <div class="navbar navbar-inverse navbar-fixed-top"
role="navigation">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="#">Project name</a>
        </div>
        <div class="collapse navbar-collapse">
          <ul class="nav navbar-nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#about">About</a></li>
            <li><a href="#contact">Contact</a></li>
          </ul>
        </div><!--/.nav-collapse -->
      </div>
    </div>

    <div class="container">

      <div class="starter-template">
        <h1>Bootstrap starter template</h1>

```

```

    <p class="lead">Use this document as a way to quickly start
any new project.<br> All you get is this text and a mostly barebones
HTML document.</p>
  </div>

</div><!-- /.container -->

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster
-->
<script src="https://code.jquery.com/jquery-
1.10.2.min.js"></script>
<script src="../../dist/js/bootstrap.min.js"></script>
</body>
</html>

```

On retrouve ce qu'on a vu précédemment avec la ligne pour les mobiles et l'appel au fichier CSS de Bootstrap. On trouve la référence d'un autre fichier CSS : `starter-template.css`. Voyons ce que contient ce fichier :

Code : CSS

```

body {
  padding-top: 50px;
}
.starter-template {
  padding: 40px 15px;
  text-align: center;
}

```

Nous verrons, lorsque nous parlerons du composant "Barre de navigation" de Bootstrap, qu'il faut parfois réserver un espace en haut de la page (avec `padding-top: 50px`) pour éviter que le texte du début ne se retrouve sous la barre. On trouve aussi ici la classe `starter-template` qui se contente de fixer un espace interne et un alignement centré du texte. Cette classe ne sert que pour le texte de présentation du template peut être évidemment supprimée ou modifiée selon votre convenance.

On trouve aussi dans la page une trame de la barre de navigation (je vous explique tout ça en détail dans ce cours) :

Code : HTML

```

<div class="navbar navbar-inverse navbar-fixed-top"
role="navigation">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Project name</a>
    </div>
    <div class="collapse navbar-collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </div><!--/.nav-collapse -->
  </div>
</div>

```

Le contenu de la page est ensuite inséré dans une DIV comportant la classe container :

Code : HTML

```
<div class="container">
  <div class="starter-template">
    <h1>Bootstrap starter template</h1>
    <p class="lead">Use this document as a way to quickly start any
new project.<br> All you get is this text and a mostly barebones
HTML document.</p>
  </div>
</div><!-- /.container -->
```

Nous verrons également l'intérêt de cette classe et son utilisation. Considérez les exemples de templates fournis sur le site de Bootstrap à la fois comme des guides de démarrage et des aides à la compréhension du framework. Évitez de les prendre tels quels sans en comprendre tous les éléments. À l'issue de ce cours vous aurez tout en main pour le faire...

Les Media Queries

Les Media Queries sont destinées à simplifier la création de pages web pour les rendre consultables sur des supports variés (tablettes, smartphones...). Cette section n'est qu'une introduction rapide à ce domaine qui mériterait un cours à lui tout seul.

Avec les Media Queries on peut cibler :

- La résolution ;
- Le type de media ;
- La taille de la fenêtre (width et height) ;
- La taille de l'écran (device-width et device-height) ;
- Le nombre de couleurs ;
- Le ratio de la fenêtre (par exemple le 16/9) ;
- etc.



Les plus courageux peuvent consulter directement [les spécifications du W3C](#).

La liste est longue, malheureusement peu de navigateurs actuels sont capables de digérer tout ça. On se limite en général à la taille de l'affichage, l'orientation et parfois le ratio.



Je ne comprends pas la différence entre taille de l'écran et taille de la fenêtre !

Ah ! Bonne réflexion. Cette distinction n'a aucun sens pour les écrans d'ordinateurs mais trouve tout son intérêt pour les smartphones. La taille de l'écran est la surface physique réelle alors que la taille de la fenêtre est celle dont le smartphone "pense" disposer. Ne me demandez pas pourquoi ce n'est pas la même valeur, je n'en sais rien. Par exemple pour un iPhone 4 la surface réelle est de 640x960 px et la surface de la fenêtre est de 320x480 px. Autrement dit un iPhone est pessimiste sur ses possibilités d'affichage !

Mais ce n'est pas le seul souci ! Le navigateur embarqué sur un smartphone a lui aussi une certaine idée de la surface d'affichage dont il dispose (le viewport). Par exemple, pour continuer avec l'exemple de l'iPhone 4, qui utilise Safari, ce dernier pense disposer d'une largeur de 980 px ! Sur ces appareils il est aussi possible d'utiliser le zoom et notre mise en page risque de s'en ressentir. On peut se demander comment s'en sortir dans tout ce bazar... Heureusement on dispose de la balise META viewport pour fixer certaines valeurs, voici ce qui est préconisé dans le template de Bootstrap :

Code : HTML

```
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

- On ouvre la fenêtre à la largeur de l'écran : `width=device-width`
- On règle le zoom : `initial-scale=1.0`

Vous pouvez aussi fixer d'autres valeurs. Par exemple, empêcher l'utilisateur de zoomer. Il existe [un site](#) intéressant pour tester le rendu sur mobile.

Syntaxe des Media Queries

Une media query est une expression dont la valeur est soit vraie, soit fausse. Voici un exemple dans du code CSS :

Code : CSS

```
@media (max-width: 767px) {  
  ...  
}
```

`@media` est une règle apparue avec le CSS2. Elle permet à la base de cibler un media : écran, imprimante, projecteur... Cette règle est reprise et étendue avec le CSS3. L'expression située entre parenthèses est évaluée, et si elle est « vraie », les règles situées dans le bloc sont prises en compte.

Ici l'expression est `(max-width: 767px)`, elle se comprend facilement. Elle est « vraie » si le support de visualisation de la page a une largeur d'affichage inférieure ou égale à 767px.



Oui, mais on n'a pas spécifié de média là !

Oui parce qu'on veut prendre en compte tous les médias existants ! Voici une syntaxe équivalente :

Code : CSS

```
@media all and (max-width: 767px) { ... }
```

On en profite pour voir qu'on peut combiner des éléments avec des opérateurs logiques.

Voici une autre expression avec le même opérateur logique :

Code : CSS

```
@media (min-width: 768px) and (max-width: 979px) { ... }
```

Ici, on veut prendre en compte les règles du bloc si l'affichage se situe entre 768px et 979px.

Si on veut appliquer des règles lorsque l'orientation est « portrait », voici la syntaxe à utiliser :

Code : CSS

```
@media (orientation: portrait) { ... }
```

Comme vous pouvez le voir, la syntaxe est simple, ce qui l'est moins ce sont les règles à définir. Mais quand vous utilisez Bootstrap, vous n'avez pas à vous soucier de tout ça... Si vous explorez le code de Bootstrap, vous trouverez beaucoup d'utilisation des Media Queries, comme par exemple ici, où est définie une règle de dimension maximale pour la classe `container` lorsque l'affichage est d'au moins 992 pixels :

Code : CSS

```
@media (min-width: 992px) {  
  .container {  
    max-width: 970px;  
  }  
  ...  
}
```

Nous verrons plus loin dans ce cours que Bootstrap définit quatre types de medias selon la dimension de l'affichage. Voici la syntaxe utilisée :

Code : CSS

```
/* Extra small devices (phones, up to 480px) */  
/* No media query since this is the default in Bootstrap */  
  
/* Small devices (tablets, 768px and up) */  
@media (min-width: @screen-tablet) { ... }  
  
/* Medium devices (tablets, 992px and up) */  
@media (min-width: @screen-desktop) { ... }  
  
/* Large devices (large desktops, 1200px and up) */  
@media (min-width: @screen-large-desktop) { ... }
```

En résumé :

- Bootstrap est un framework CSS qui comporte en plus des plugins jQuery pour composer des pages web.
- Bootstrap est un framework récent qui a connu un développement et une popularité très rapide.
- Bootstrap s'installe facilement en référençant quelques fichiers sur son serveur ou même en passant directement par des CDN.
- Bootstrap propose des templates de démarrage pour éviter de partir avec une page blanche.
- Bootstrap intègre des medias queries pour adapter les pages web à tous les support de visualisation.

Une grille ?

L'organisation spatiale des pages web est l'une des premières préoccupations lorsque l'on crée un site web. Prévoit-on une bannière ? Faut-il un espace pour un menu à gauche ou en haut ? Y aura-t-il des blocs sur un des côtés pour recevoir certaines fonctionnalités comme la connexion ou des infos ? Faut-il prévoir un bas de page ?

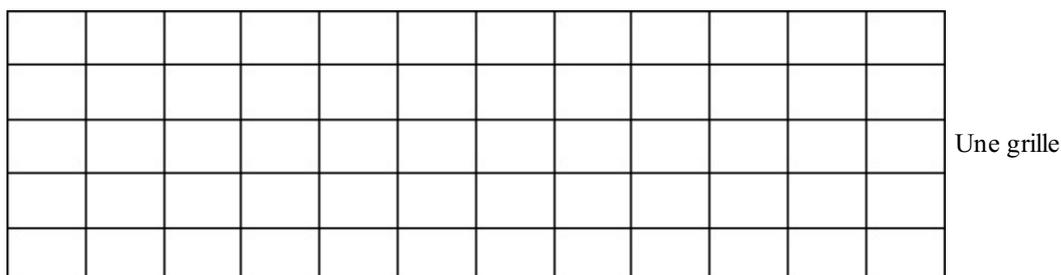
Bootstrap ne répond évidemment pas à ces questions, mais il peut grandement vous faciliter la tâche, avec son système de grille, pour obtenir le résultat que vous souhaitez.

Petite visite guidée...

Présentation

Le principe d'une grille

Une grille est tout simplement un découpage en cellules de mêmes dimensions (voir figure suivante).



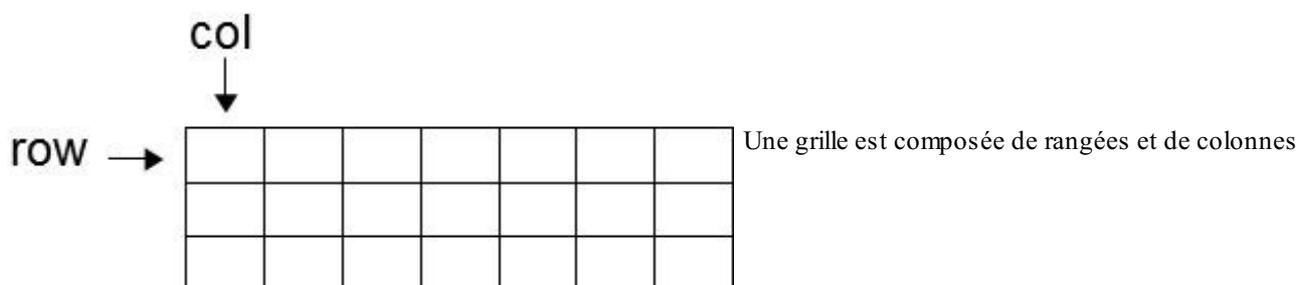
On peut alors décider d'organiser du contenu en utilisant pour chaque élément une ou plusieurs cellules, comme à la figure suivante.



La grille de Bootstrap comporte 12 colonnes comme dans l'illustration précédente. Vous commencez sans doute à comprendre l'utilité de cette organisation.

Terminologie

Une grille est découpée en rangées (appelées *row*, parce que tout est en anglais) et colonnes (*col*), comme à la figure suivante.



La grille de Bootstrap

La grille de Bootstrap n'est pas aussi idéale que celle présentée précédemment. Le découpage en colonnes est tout simplement une division en pourcentage de la largeur de la fenêtre de visualisation et correspond donc à ce qu'on vient de voir. En revanche, il n'en est pas vraiment de même pour les rangées. Ces dernières ont la hauteur de leur contenu (voir figure suivante).

	Petit élément			Gros élément	

Les rangées ont la hauteur de leur contenu

Moralité, une rangée prend la hauteur du plus gros élément qu'elle contient. Puisque la largeur des colonnes est contrainte, le flux des données s'écoule verticalement, ce qui est un comportement HTML classique. Il faudra en tenir compte lors de la mise en page.

Organisation de la grille

Bootstrap est essentiellement un fichier CSS. Il comporte de nombreuses classes que l'on peut utiliser directement dans les balises HTML.

La première classe à connaître est `row`, qui représente une rangée. Il faut ensuite définir le nombre de colonnes pour chaque élément en sachant qu'il y en a au maximum 12. Pour définir le nombre de colonnes utilisées pour chaque élément, on dispose de quatre batteries de 12 classes :

- `col-xs-1` ou `col-sm-1` ou `col-md-1` ou `col-lg-1`
- `col-xs-2` ou `col-sm-2` ou `col-md-2` ou `col-lg-2`
- ...
- `col-xs-12` ou `col-sm-12` ou `col-md-12` ou `col-lg-12`



Pourquoi 4 sortes de classes pour les colonnes ?

Je vous ai déjà dit que Bootstrap est « responsive », ce qui veut dire qu'il s'adapte à la taille de l'écran. Il permet une visualisation aussi bien sur un écran géant que sur un smartphone. Mais que se passe-t-il pour les éléments d'une page web lorsque la fenêtre diminue ou s'élargit ? On peut envisager deux hypothèses : les éléments se redimensionnent en restant positionnés, ou alors ils s'empilent quand la fenêtre devient plus étroite et se positionnent côte à côte quand elle s'élargit. Voici à la figure suivante l'exemple de [la page d'accueil du site OpenClassrooms](#) qui illustre ce phénomène avec trois situations (écran large, écran moyen et petit écran) :

Infos **OC** OPENCLASSROOMS

Préinscrivez-vous au premier MOOC sur la création de site web en HTML5/CSS3 ! [En savoir plus](#)

Français ▾

Cours ▾ Forums Librairie Plus ▾ Recherche

Inscription Connexion

Rejoignez-les

LES COURS LES PLUS OUVERTS DU WEB

800 cours gratuits en ligne. Une communauté de 600 000 professeurs et étudiants.

INTERNET a bouleversé NOTRE FAÇON

Que voulez-vous apprendre aujourd'hui ? Exemple : HTML5, physique quantique, référencement

DU 11 SEPTEMBRE AU 2 OCTOBRE 2013

2 LIVRES ACHETÉS, LE 3^E OFFERT* PARI :

[En savoir plus](#)

*UN LIVRE OFFERT PARI LES 5 TITRES CI-DESSUS. VOIR CONDITIONS DE L'OFFRE.

Nouveaux cours

- Arduino pour bien commencer en électronique et en programmation**
OpenClassrooms
S. Landraut et H. Weisslinger
- Apprenez le fonctionnement des réseaux TCP/IP**
INTECH INFO
E. Lalitte et R. Guichard
- Nous descendons tous de Gengis Khan !**
OpenClassrooms
M. Nebra et M. Launay
- Des applications ultra-rapides avec Node.js**
OpenClassrooms
M. Nebra

Aspect sur grand écran



Préinscrivez-vous au premier MOOC sur la création de site web en HTML5/CSS3 ! [En savoir plus](#) Français ▾

Inscription Connexion

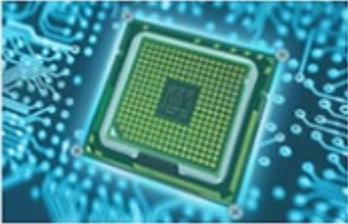
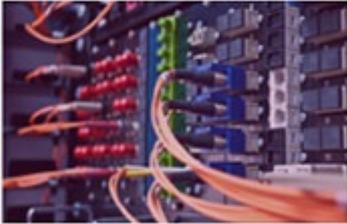
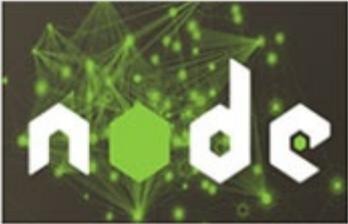
Cours ▾ Forums Librairie Etudes Emploi Recherche

INTERNET
à *bouleversé*
NOTRE FAÇON

LES COURS LES PLUS OUVERTS DU WEB

800 cours gratuits en ligne. Une communauté de 600 000 professeurs et étudiants.

▣ Nouveaux cours ▣

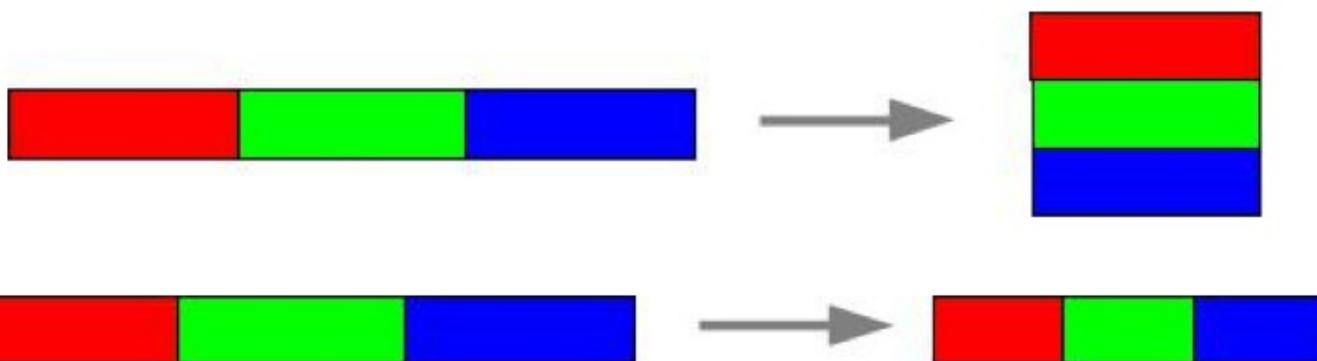
 <p>Arduino pour bien commencer en électronique et en programmation</p> <p>OpenClassrooms S. Landrault et H. Weisslinger</p>	 <p>Apprenez le fonctionnement des réseaux TCP/IP</p> <p>INTECH INFO E. Lalitte et R. Guichard</p>	 <p>Nous descendons tous de Gengis Khan !</p> <p>OpenClassrooms M. Nebra et M. Launay</p>
 <p>Des applications ultra-rapides avec Node.js</p>	<p>Améliorez la visibilité de votre site grâce au référencement</p>	<p>Développez votre site web avec le framework Symfony2</p>

Aspect sur écran moyen



Aspect sur petit écran

Plutôt que de ratatiner les éléments au risque de les rendre illisibles, le choix a été fait de les empiler petit à petit quand la fenêtre devient plus étroite. Une autre option pourrait consister à supprimer des éléments pas vraiment utiles, mais nous reviendrons plus tard sur ce point. Pour le moment on va juste se demander comment on peut choisir entre les deux options présentées à la figure suivante.



Empilement ou réduction ?

C'est ici qu'interviennent les 4 sortes de classes vues précédemment pour les colonnes. Bootstrap considère 4 sortes de médias : les petits, genre smartphones (moins de 768 pixels), les moyens, genre tablettes (moins de 992 pixels), les écrans moyens (moins de 1200 pixels) et enfin les gros écrans (plus de 1200 pixels). Vous trouverez à la figure suivante un tableau pour illustrer les différences de réaction selon la catégorie.

	Petit écran (smartphone)	Écran réduit (tablette)	Écran moyen (desktop)	Grand écran (desktop)
				
Comportement	Redimensionnement	Redimensionnement	Redimensionnement	Empilage puis redimensionnement
Classe	col-xs-*	col-sm-*	col-md-*	col-lg-*
Valeur de référence	< 768 px	> 768 px	> 992 px	> 1200 px

Le nom des classes est intuitif : xs pour *x-small*, sm pour *small*, md pour *medium* et lg pour *large*.



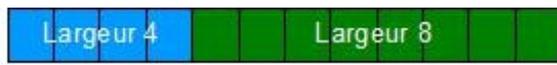
Un petit exemple ?

Pour avoir un élément de 4 colonnes de large accolé avec un élément de 8 colonnes de large sur un smartphone, on a :

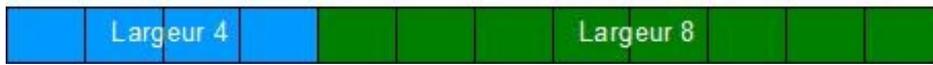
Code : HTML

```
<div class="row">
  <div class="col-xs-4">Largeur 4</div>
  <div class="col-xs-8">Largeur 8</div>
</div>
```

La figure suivante illustre ce code avec l'effet quand on passe sur un écran plus grand.



Un élément de 4 colonnes à côté d'un



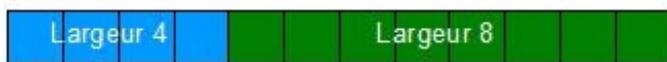
élément de 8 colonnes

Voici la version tablette :

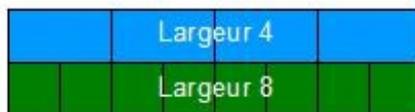
Code : HTML

```
<div class="row">
  <div class="col-sm-4">Largeur 4</div>
  <div class="col-sm-8">Largeur 8</div>
</div>
```

La figure suivante illustre ce code avec l'effet quand on passe sur un écran plus petit ou plus grand.



L'affichage change en fonction de la taille



de l'écran



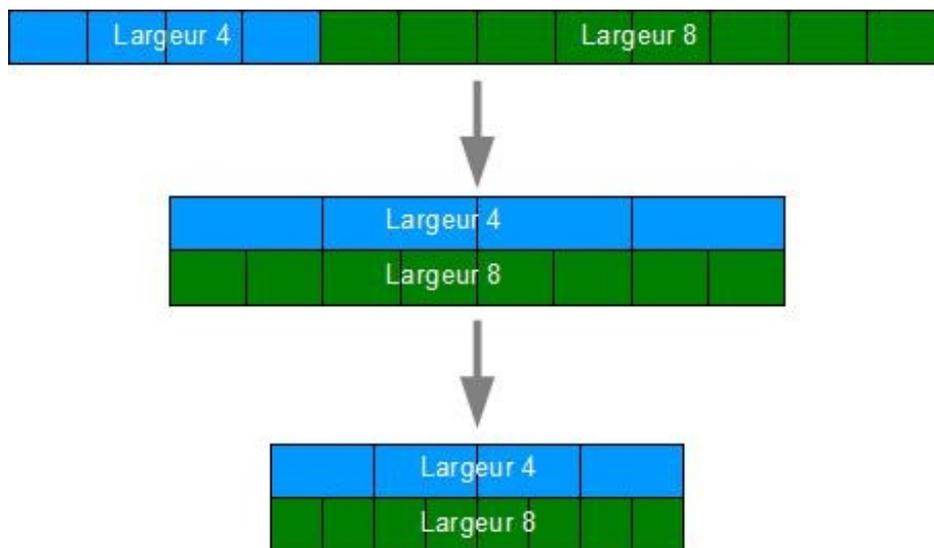
Vous remarquez que lors de l'empilage, les colonnes prennent toute la place disponible.

Et enfin, voici la version grand écran :

Code : HTML

```
<div class="row">
  <div class="col-lg-4">Largeur 4</div>
  <div class="col-lg-8">Largeur 8</div>
</div>
```

La figure suivante illustre ce code avec l'effet quand on passe sur un écran de plus en plus petit.



La version grand écran

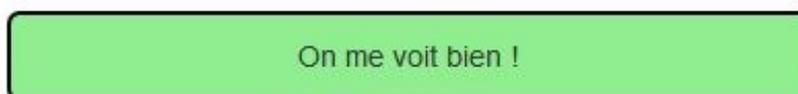
La largeur des éléments de la grille est calculée en pourcentage selon la fenêtre de visualisation. Rien n'interdit évidemment de mélanger des classes des 3 catégories pour créer des effets particuliers avec certains éléments qui doivent s'empiler et d'autre pas...

Après cette petite mise en jambe pour vous présenter la notion de grille dans Bootstrap, on va passer en revue les possibilités dans le détail. Mais pour visualiser tous les exemples qui vont suivre, je vous propose d'ajouter un peu de style, histoire d'afficher les éléments de façon explicite. Pour éviter de polluer le code HTML on va mettre ce style spécifique au cours dans un fichier CSS particulier :

Code : CSS

```
body {
  padding-top: 10px;
}
[class*="col-"], footer {
  background-color: lightgreen;
  border: 2px solid black;
  border-radius: 6px;
  line-height: 40px;
  text-align: center;
}
```

Le seul but de ces règles de style est de faire apparaître nettement les éléments à l'écran. Une petite marge interne en haut pour le corps, pour éviter que tout soit collé en haut de l'écran. Des bordures (avec des coins arrondis pour l'esthétique) et un fond coloré pour distinguer les éléments. Une hauteur fixée à 40 pixels, parce qu'étant donné que la hauteur des rangées dépend du contenu, on aurait un rendu hétéroclite. La figure suivant présente l'aspect obtenu.



Le résultat final

La grille en pratique

Le conteneur

La grille de Bootstrap doit être placée dans un conteneur. Bootstrap propose les classes `container` et `container-fluid`. Leur utilisation était auparavant optionnelle. Il est désormais clairement indiqué dans la documentation qu'il faut la mettre en œuvre systématiquement si on veut obtenir des alignements et des espacements corrects. La classe `container` contient et centre la grille sur une largeur fixe, qui s'adapte en fonction de la largeur de l'écran. La classe `container-fluid` permet à la grille d'occuper toute la largeur. Dans les exemples de ce chapitre, je vais utiliser systématiquement `container` pour avoir une visualisation plus facile des éléments. Ce conteneur a une largeur maximale selon le média concerné, comme indiqué au tableau suivant.

	Petit écran (smartphone)	Ecran réduit (tablette)	Ecran moyen (desktop)	Grand écran (desktop)
				
Conteneur maximum	Automatique	750 px	970 px	1170 px
Valeur de référence	< 768 px	>= 768 px	>= 992 px	>= 1200 px

Le centrage du conteneur est fait de façon classique avec de petites marges internes de 15 pixels et les marges droite et gauche automatiques :

Code : CSS

```
.container {
  padding-right: 15px;
  padding-left: 15px;
  margin-right: auto;
  margin-left: auto;
}
```

Ce sont les mêmes règles pour la classe `container-fluid`. Ce qui est ajouté pour la classe `container` est la limite de largeur spécifiée par média :

Code : CSS

```
@media (min-width: 768px) {
  .container {
    max-width: 750px;
  }
  ...
  @media (min-width: 992px) {
    .container {
      max-width: 970px;
    }
    ...
    @media (min-width: 1200px) {
      .container {
        max-width: 1170px;
      }
      ...
    }
  }
}
```

Une seule rangée

Dans ce premier exemple, on va déclarer une seule rangée avec deux éléments qui occupent tout l'espace :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
```

```

<div class="col-lg-4">4 colonnes</div>
<div class="col-lg-8">8 colonnes</div>
</div>
</body>
</html>

```

Ce qui nous donne la figure suivante.



Une seule rangée avec deux éléments qui occupent tout l'espace

Mise en page rapide et simple non ?

Plusieurs rangées

Pour obtenir plusieurs rangées, il suffit d'utiliser plusieurs fois la classe `row`. Un petit exemple avec 3 rangées pour voir l'effet :

Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-lg-1">1 col</div>
        <div class="col-lg-2">2 colonnes</div>
        <div class="col-lg-3">3 colonnes</div>
        <div class="col-lg-6">6 colonnes</div>
      </div>
      <div class="row">
        <div class="col-lg-12">12 colonnes</div>
      </div>
      <div class="row">
        <div class="col-lg-4">4 colonnes</div>
        <div class="col-lg-8">8 colonnes</div>
      </div>
    </body>
  </html>

```

Ce qui nous donne la figure suivante.



Trois rangées

Sauter des colonnes

Bootstrap permet aussi de sauter des colonnes. Il y a quelques classes d'offset pour ça :

- `col-*-offset-1`
- `col-*-offset-2`
- ...
- `col-*-offset-11`

Ces classes se contentent d'ajouter une marge gauche. Par exemple :

Code : CSS

```
.col-lg-offset-6 {
  margin-left: 50%;
}
```

Dans ce cas, 6 colonnes représentent la moitié de la fenêtre donc 50%. Voici un exemple :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-lg-3">3 colonnes</div>
        <div class="col-lg-6">6 colonnes</div>
        <div class="col-lg-3">3 colonnes</div>
      </div>
      <div class="row">
        <div class="col-lg-3">3 colonnes</div>
        <div class="col-lg-offset-6 col-lg-3">3 colonnes</div>
      </div>
    </div>
  </body>
</html>
```

Voici le résultat à la figure suivante.



On a sauté des colonnes

On a sauté 6 colonnes grâce à la classe `col-lg-offset-6`. On peut évidemment multiplier les sauts :

Code : HTML

```
<div class="container">
  <div class="row">
    <div class="col-lg-2 col-lg-offset-1">2 colonnes</div>
    <div class="col-lg-4 col-lg-offset-2">4 colonnes</div>
```

```
<div class="col-lg-2 col-lg-offset-1">2 colonnes</div>
</div>
</div>
```

Voici le résultat à la figure suivante.



On a sauté des colonnes plusieurs fois

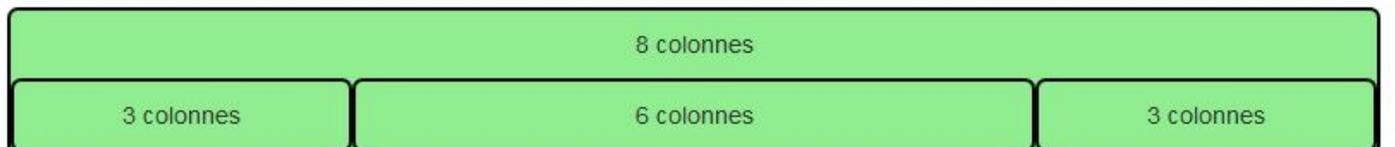
Imbrication d'éléments

On doit souvent imbriquer des éléments dans une page web, est-ce possible avec Bootstrap ? Autrement dit inclure un row dans un col. Eh bien, on va tester ça tout de suite :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-lg-8">8 colonnes
          <div class="row">
            <div class="col-lg-3">3 colonnes</div>
            <div class="col-lg-6">6 colonnes</div>
            <div class="col-lg-3">3 colonnes</div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Voici le résultat à la figure suivante.



On a imbriqué des rangées

On dirait bien que ça fonctionne !

Les imbrications peuvent s'enchaîner pour répondre à des besoins complexes :

Code : HTML

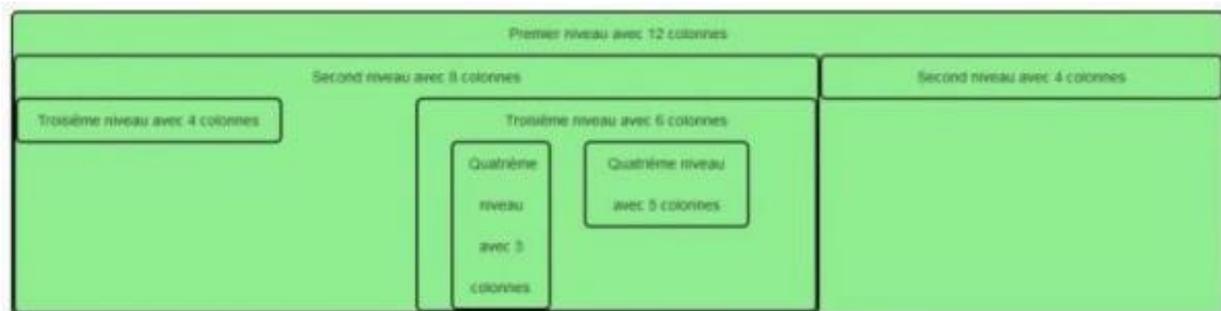
```
<div class="container">
  <div class="row">
    <div class="col-md-12">Premier niveau avec 12 colonnes
```

```

<div class="row">
  <div class="col-md-8">Second niveau avec 8 colonnes
  <div class="row">
    <div class="col-md-4">Troisième niveau avec 4 colonnes</div>
    <div class="col-md-6 col-md-offset-2">Troisième niveau avec
6 colonnes
    <div class="row">
      <div class="col-md-3 col-md-offset-1">Quatrième niveau
avec 3 colonnes</div>
      <div class="col-md-5 col-md-offset-1">Quatrième niveau
avec 5 colonnes</div>
    </div>
  </div>
</div>
<div class="col-md-4">Second niveau avec 4 colonnes</div>
</div>
</div>

```

Voici le résultat à la figure suivante.



imbrication complexe avec des sauts de colonnes

Ordre des colonnes

Très logiquement, les colonnes s'affichent dans l'ordre du flux. Il peut arriver qu'on veuille bouleverser cet ordre. Regardez cet exemple :

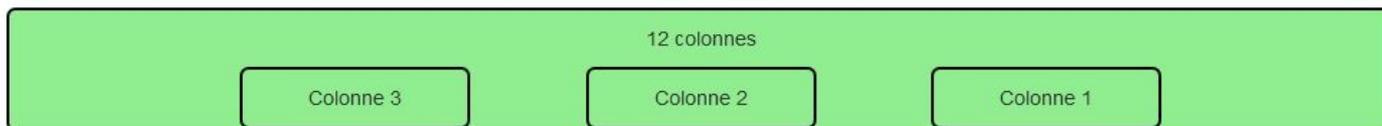
Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-lg-12">12 colonnes
          <div class="row">
            <div class="col-lg-2 col-lg-push-8">Colonne 1</div>
            <div class="col-lg-2 col-lg-push-3">Colonne 2</div>
            <div class="col-lg-2 col-lg-pull-2">Colonne 3</div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

Et voici le résultat à la figure suivante.

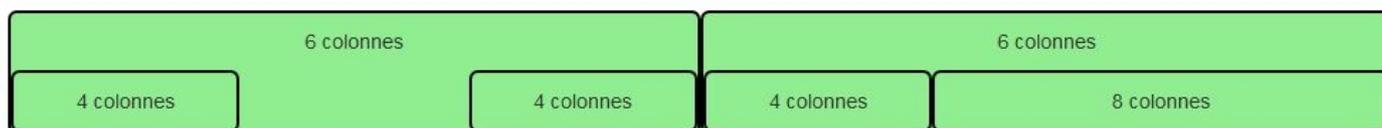


On peut choisir l'ordre des colonnes

La classe `col-lg-push-*` permet de décaler une colonne vers la droite et la classe `col-lg-pull-*` fait l'inverse.

Un petit TP

Pour vous entraîner, je vous propose d'obtenir le résultat visible à la figure suivante.



Essayez d'obtenir ce résultat

Donc deux zones séparées avec un offset de 2 colonnes dans la partie gauche.

Voici la solution :

Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-lg-6">6 colonnes
          <div class="row">
            <div class="col-lg-4">4 colonnes</div>
            <div class="col-lg-offset-4 col-lg-4">4 colonnes</div>
          </div>
        </div>
        <div class="col-lg-6">6 colonnes
          <div class="row">
            <div class="col-lg-4">4 colonnes</div>
            <div class="col-lg-8">8 colonnes</div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

```

Mise en page

L'intérêt principal d'une grille est de réaliser une mise en page. Nous allons voir quelques exemples pour structurer correctement une page.

Premier cas

Commençons par quelque chose de simple avec une en-tête, un menu à gauche, une section à droite et un pied de page :

Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <header class="row">
        <div class="col-lg-12">
          Entete
        </div>
      </header>
      <div class="row">
        <nav class="col-lg-2">
          Menu
        </nav>
        <section class="col-lg-10">
          Section
        </section>
      </div>
      <footer class="row">
        Pied de page
      </footer>
    </div>
  </body>
</html>

```

Le résultat se trouve à la figure suivante.



Une mise en page simple

J'ai utilisé les classes pour grand écran avec les classe `col-lg-*`, ce qui fait qu'à la réduction, je me retrouve avec un empilage dès que je passe en dessous de 1200 pixels (voir figure suivante).



Empilage à la réduction de l'affichage

Si je veux que ma mise en forme reste stable aussi pour les écrans moyens, comment faire ? Il me suffit d'utiliser les classes `col-md-*` à la place de `col-lg-*`, ce qui a pour effet de déplacer la limite à 992 pixels ([tester](#)). De la même façon, l'utilisation des

classes `col-sm-*` déplace la limite à 768 pixels (`tester`)...



J'ai le même résultat sans utiliser la classe `col-lg-12`, est-elle vraiment utile ?

On obtient effectivement la même mise en page, mais les classes `col-*` ont un padding droite et gauche de 15 pixels, et si on n'utilise pas la classe `col-lg-12`, on va avoir une incohérence au niveau de l'affichage du contenu.

Second cas

Considérons maintenant un cas un peu plus riche :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <header class="row">
        <div class="col-sm-12">
          Entete
        </div>
      </header>
      <div class="row">
        <nav class="col-sm-2">
          Menu
        </nav>
        <section class="col-sm-10">
          Section
          <div class="row">
            <article class="col-md-10">
              Article
            </article>
            <aside class="col-md-2">
              Aside
            </aside>
          </div>
        </section>
      </div>
      <footer class="row">
        <div class="col-sm-12">
          Pied de page
        </div>
      </footer>
    </div>
  </body>
</html>
```

Le résultat se trouve à la figure suivante.



Une mise en page plus riche

On a conservé l'en-tête et le pied de page, ainsi que le menu, mais on a intégré dans le contenu un article et un aside. L'ensemble est prévu pour fonctionner sur desktop (grand et petit). Pour les tablettes, on se retrouve avec un empilage pour le contenu de la section (utilisation de classes `col-md-*` pour `article` et `aside`), comme à la figure suivante.



Résultat sur une tablette

Et comme on n'a rien prévu de spécial pour les smartphones, tout s'empile (voir figure suivante).



Résultat sur un

smartphone

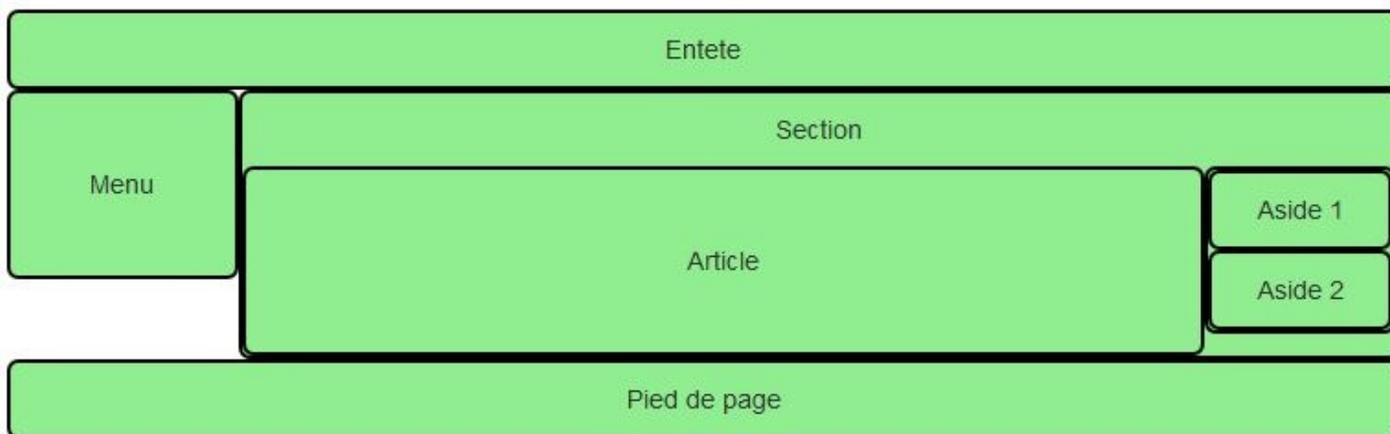
Troisième cas

Enrichissons encore la mise en page :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
    <style type="text/css">
      /* Style pour l'exemple*/
      article.col-sm-10, nav.col-sm-2 {
        line-height: 100px;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <header class="row">
        <div class="col-lg-12">
          Entete
        </div>
      </header>
      <div class="row">
        <nav class="col-sm-2">
          Menu
        </nav>
        <section class="col-sm-10">
          Section
          <div class="row">
            <article class="col-sm-10">
              Article
            </article>
            <div class="col-sm-2">
              <div class="row">
                <aside>
                  Aside 1
                </aside>
                <aside>
                  Aside 2
                </aside>
              </div>
            </div>
          </div>
        </section>
      </div>
      <footer class="row">
        <div class="col-lg-12">
          Pied de page
        </div>
      </footer>
    </div>
  </body>
</html>
```

Le résultat se trouve à la figure suivante.



Une mise en page plus complexe

Maintenant nous avons 2 *asides* à côté de l'article. Pour y arriver, il faut 2 niveaux d'imbrication. On a aussi un empilage complet à la réduction, comme le montre la figure suivante.



Résultat à la réduction



Pourquoi l'en-tête et le pied de page ont maintenant la classe `col-lg-12`, alors que dans l'exemple précédent on avait `col-sm-12` ?

Comme la largeur est de 12 colonnes, tout ce qui nous intéresse est le centrage, que ces 2 classes possèdent en commun, sans influence du média utilisé. On peut donc utiliser indifféremment ces 2 classes dans notre cas.

Quatrième cas

Terminons avec un cas plus fourni :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
```

```

<link href="assets/css/bootstrap.css" rel="stylesheet">
<link href="assets/css/tuto.css" rel="stylesheet">
<style type="text/css">
  /* Style pour l'exemple*/
  footer {
    border: 0;
  }
  article.col-sm-6, nav.col-sm-2 {
    line-height: 60px;
  }
</style>
</head>
<body>
  <div class="container">
    <header class="row">
      <div class="col-lg-12">
        Entete
      </div>
    </header>
    <div class="row">
      <nav class="col-sm-2">
        Menu
      </nav>
      <section class="col-sm-10">
        Section
        <div class="row">
          <div class="col-sm-10">
            <div class="row">
              <article class="col-sm-6">
                Article
              </article>
              <article class="col-sm-6">
                Article
              </article>
            </div>
          </div>
          <div class="col-sm-2">
            <div class="row">
              <aside>
                Aside 1
              </aside>
              <aside>
                Aside 2
              </aside>
            </div>
          </div>
        </div>
      </section>
      <section class="col-sm-offset-2 col-sm-10">
        Section
        <div class="row">
          <article class="col-sm-6">
            Article
          </article>
          <article class="col-sm-6">
            Article
          </article>
          <article class="col-sm-6">
            Article
          </article>
        </div>
      </section>
    </div>
  </body>

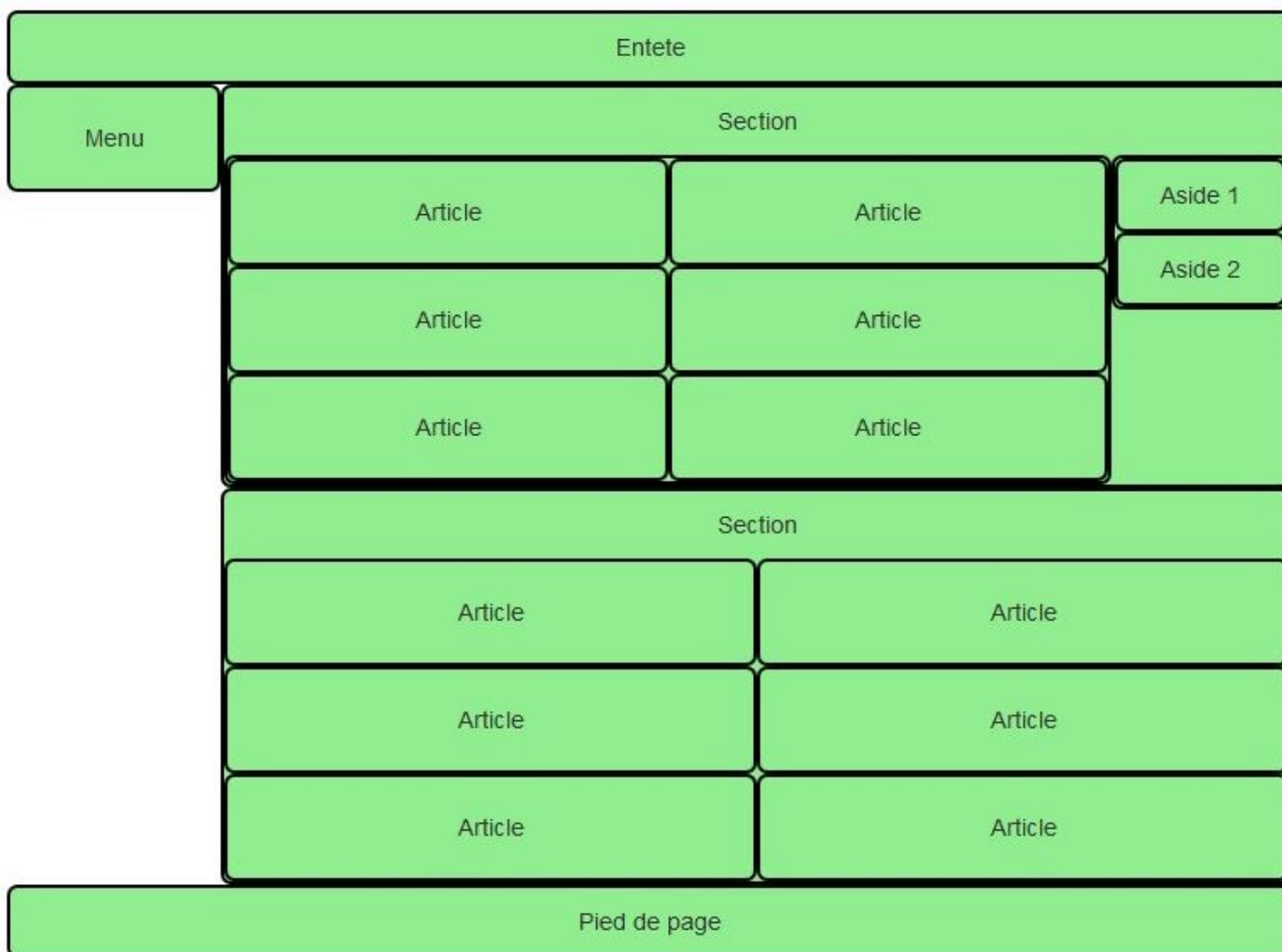
```

```

        Article
    </article>
    <article class="col-sm-6">
        Article
    </article>
    <article class="col-sm-6">
        Article
    </article>
    <article class="col-sm-6">
        Article
    </article>
    </div>
</section>
</div>
<footer class="row">
    <div class="col-lg-12">
        Pied de page
    </div>
</footer>
</div>
</body>
</html>

```

Le résultat se trouve à la figure suivante.



Une mise en page très chargée

Dans les 2 sections, j'ai mis 6 articles d'une largeur de 6 colonnes qui se positionnent parfaitement. Il n'est pas utile de créer 3 rangées. Dans la deuxième section, je n'ai pas mis d'aside pour montrer un mélange de compositions. Le décalage de la

deuxième section pour tenir compte de la largeur du menu se fait tout simplement avec une classe d'offset. En ce qui concerne le choix des classes pour les colonnes je me suis arrangé pour que la première section garde sa structure quand on rétrécit.

Vous avez pu voir avec ces quelques exemples qu'on peut obtenir facilement une structure de page aussi complexe que l'on veut. Le point le plus délicat demeure le choix des classes pour les colonnes en fonction du résultat que l'on désire selon les médias utilisés. Dans tous les cas, il faut tester les différents formats parce que le rendu ne donne pas forcément du premier coup ce que l'on avait prévu. Si vous n'y parvenez pas avec une seule classe, alors la solution est d'en combiner plusieurs, c'est ce que nous allons voir bientôt...

En résumé :

- Bootstrap propose un grille pour positionner tous les éléments des pages web.
- La grille est versatile et permet de nombreuses combinaisons comme des inclusions, des sauts de colonnes.
- La grille rend possible également une adaptation selon les dimensions du support de visualisation en réorganisant les éléments ou en masquant certains.

Un peu de pratique

Nous avons vu en détail les possibilités de la grille de Bootstrap. Nous avons vu également comment faire des mises en pages adaptées à différents types de supports. Il est temps maintenant de mettre en application toutes ces connaissances avec des applications pratiques.

Combiner les formats et exemple de page

Nous allons maintenant utiliser ce que nous avons vu dans les chapitres précédents pour construire une page pratique.

Combinaison de classes « col-* »

Nous avons vu des mises en page utilisant sélectivement les classes pour les colonnes prévues pour les différents formats de supports. Nous allons à présent envisager leur combinaison pour gérer certains cas. Dans la page à réaliser, je veux avoir des petites photos côte à côte. Supposons que je parte de cette structure pour cette partie de la page :

Code : HTML

```
<!DOCTYPE HTML>
<html>

  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      [class*="col"] { margin-bottom: 20px; }
      img { width: 100%; }
      body { margin-top: 10px; }
    </style>
  </head>

  <body>

    <div class="container">
      <section class="row">
        <div class="col-lg-2"></div>
        <div class="col-lg-2"></div>
      </section>
    </div>

  </body>
</html>
```

Le rendu sur grand écran est parfait avec 6 photos sur la largeur (voir figure suivante).



Affichage sur grand écran

Mais ça se gâte quand je rétrécis la fenêtre, puisque je sais qu'en dessous de 1200 pixels les éléments s'empilent. Du coup je me retrouve avec une image sur la largeur, et comme je les ai prévues en basse résolution, elle pixellise, comme à la figure suivante.



Affichage au-dessous de 1200 pixels



L'idéal serait d'avoir :

- Sur moyen et grand écran : 6 images sur la largeur
- Sur tablette : 4 images sur la largeur
- Sur smartphone : 3 images sur la largeur

Comment réaliser cela ? Tout simplement en combinant les classes `col-*` :

Code : HTML

```
<div class="container">
  <section class="row">
    <div class="col-xs-4 col-sm-3 col-md-2"></div>
    <div class="col-xs-4 col-sm-3 col-md-2"></div>
  </section>
</div>
```

Je vais avoir ainsi :

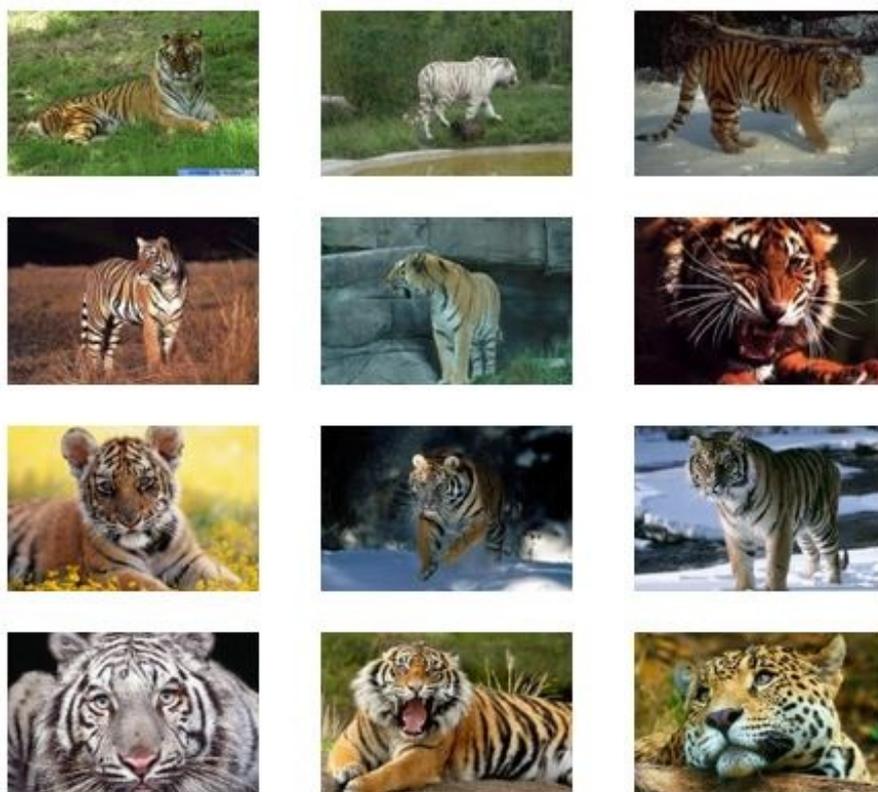
- Sur un écran de plus de 992 pixels de large : classes `col-md-2` actives
- Sur un écran entre 768 et 992 pixels de large : classes `col-sm-3` actives
- Sur un écran de moins de 768 pixels de large : classes `col-xs-4` actives

Et voici à la figure suivante le résultat sur écran moyen (classes `col-sm-3` actives).



Affichage sur écran moyen

Et à la figure suivante, le résultat sur petit écran (classes `col-xs-4` actives).



Affichage sur petit écran

J'ai donc obtenu facilement une adaptation de la présentation de mes photos en fonction du support utilisé pour les visualiser.

Page d'exemple

Construisons maintenant une page complète qui intègre la partie que nous venons de traiter :

Code : HTML

```
<!DOCTYPE HTML>
```

```

<html>

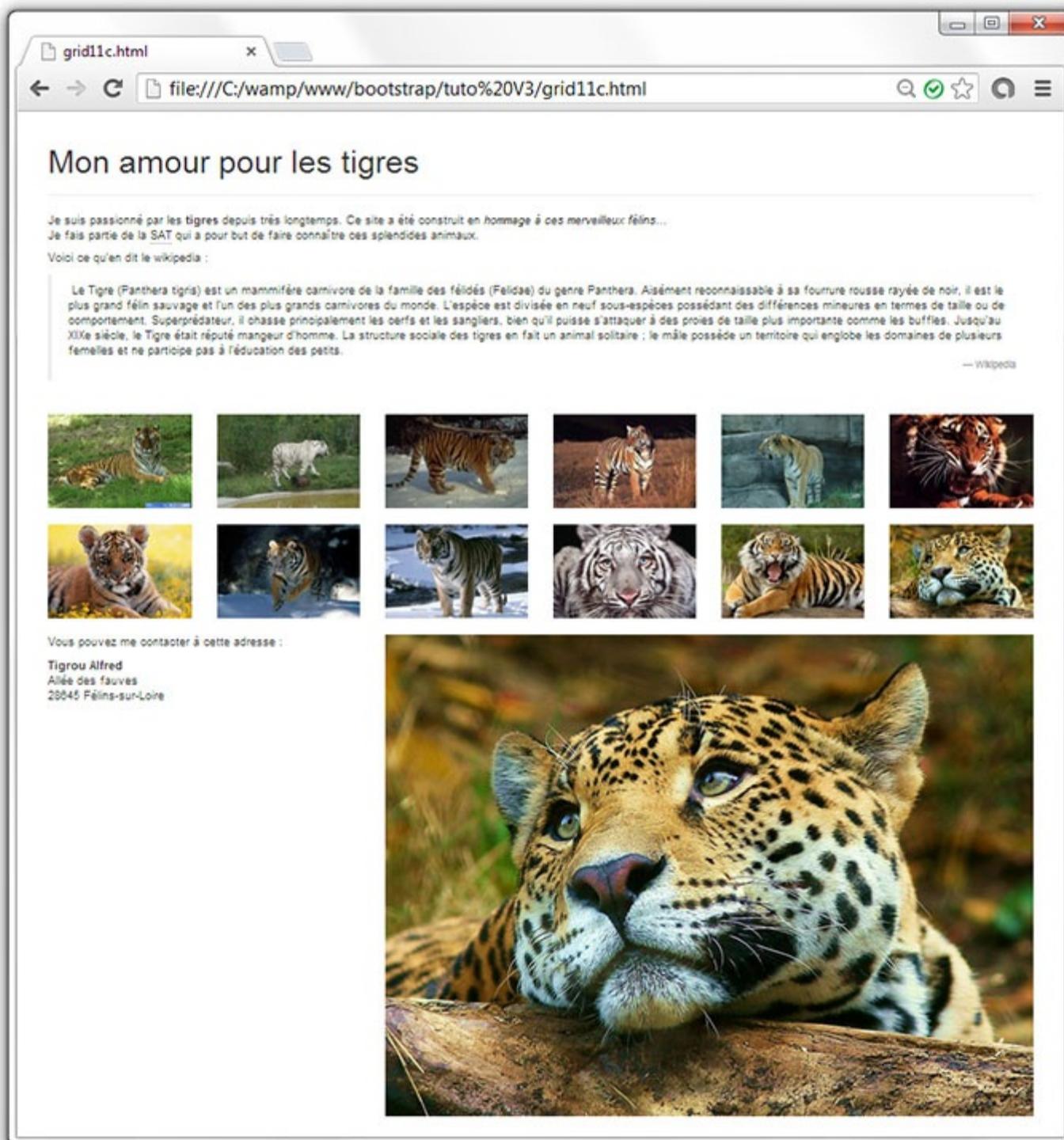
  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      [class*="col"] { margin-bottom: 20px; }
      img { width: 100%; }
    </style>
  </head>

  <body>
    <div class="container">
      <header class="page-header">
        <h1>Mon amour pour les tigres</h1>
      </header>
      <section class="row">
        <div class="col-lg-12">
          <p>
            Je suis passionné par les <strong>tigres</strong> depuis
            très longtemps. Ce site a été construit en <em>hommage à ces
            merveilleux félins...</em><br>
            Je fais partie de la <abbr title="Société des Amoureux
            des Tigres">SAT</abbr> qui a pour but de faire connaître ces
            splendides animaux.
          </p>
          <p>Voici ce qu'en dit le wikipedia :</p>
          <blockquote>
            Le Tigre (Panthera tigris) est un mammifère carnivore de
            la famille des félidés (Felidae) du genre Panthera.
            Aisément reconnaissable à sa fourrure rousse rayée de noir, il est
            le plus grand félin sauvage et l'un des plus grands carnivores du
            monde.
            L'espèce est divisée en neuf sous-espèces possédant des différences
            mineures en termes de taille ou de comportement. Superprédateur,
            il chasse principalement les cerfs et les sangliers, bien qu'il
            puisse s'attaquer à des proies de taille plus importante comme les
            buffles.
            Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La
            structure sociale des tigres en fait un animal solitaire ; le mâle
            possède
            un territoire qui englobe les domaines de plusieurs femelles et ne
            participe pas à l'éducation des petits.<br>
            <small class="pull-right">Wikipedia</small><br>
          </blockquote>
        </div>
      </section>
      <section class="row">
        <div class="col-xs-4 col-sm-3 col-md-2"></div>
        <div class="col-xs-4 col-sm-3 col-md-2"></div>

```

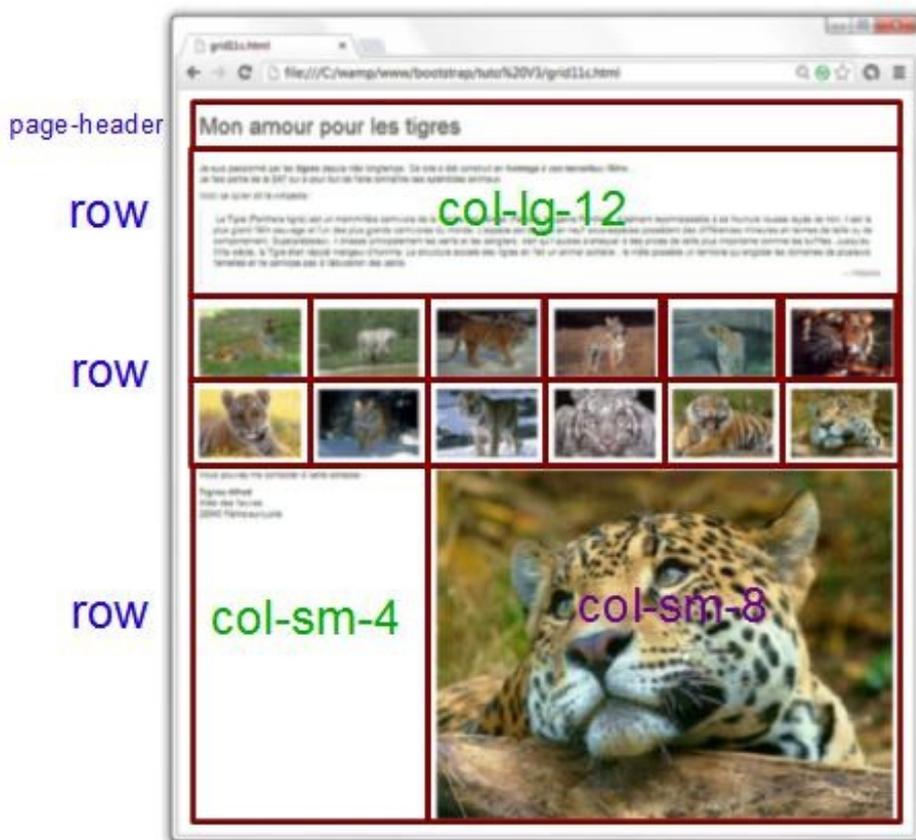
```
<div class="col-xs-4 col-sm-3 col-md-2"></div>
</section>
<section class="row">
  <aside class="col-sm-4">
    <address>
      <p>Vous pouvez me contacter à cette adresse :</p>
      <strong>Tigrou Alfred</strong><br>
      Allée des fauves<br>
      28645 Félin-sur-Loire<br>
    </address>
  </aside>
  <div class="col-sm-8">
    
  </div>
</section>
</div>
</body>
</html>
```

Voici à la figure suivante le résultat obtenu.



La page d'exemple

Une simple composition avec un en-tête et 3 rangées. L'en-tête ne comporte que le titre avec une classe un peu particulière que nous allons voir plus loin. La première rangée comporte un seul élément. La deuxième en revanche en comporte 12 et correspond à ce que nous avons vu précédemment. Voici à la figure suivante une visualisation du découpage.



La page d'exemple décryptée

La classe `page-header`

Cette classe fixe quelques valeurs :

Code : CSS - classe `page-header`

```
.page-header {
  padding-bottom: 9px;
  margin: 40px 0 20px;
  border-bottom: 1px solid #eee;
}
```

Une marge haute de 40 pixels et basse de 20 pixels. Une ligne inférieure de 1 pixel de couleur grise, avec un écart de 9 pixels entre le contenu et cette ligne. Donc une approche sympathique pour un en-tête de page. Il suffit d'y placer un titre comme je l'ai fait ici :

Code : HTML

```
<header class="page-header">
  <h1>Mon amour pour les tigres</h1>
</header>
```

Voici le résultat à la figure suivante.

Mon amour pour les tigres

La classe « `page-header` »

Quelques mises en valeur

Dans le premier paragraphe, j'ai utilisé quelques mises en valeur :

Code : HTML

```
<p>Je suis passionné par les <strong>tigres</strong> depuis très
longtemps. Ce site a été construit en <em>hommage à ces merveilleux
félins...</em><br>
Je fais partie de la <abbr title="Société des Amoureux des
Tigres">SAT</abbr> qui a pour but de faire connaître ces splendides
animaux.
</p>
```

Observez à la figure suivante la qualité du popup lorsque l'on passe le curseur au-dessus de l'abréviation SAT.

Je suis passionné par les **tigres** depuis très longtemps. Ce site a été construit en *hommage à ces merveilleux félins...*
Je fais partie de la **SAT** qui a pour but de faire connaître ces splendides animaux.

Voici ce qu'en dit le Société des Amoureux des Tigres

Apparence d'une abréviation

Une citation

Le chapitre suivant comporte une citation avec utilisation de la balise blockquote :

Code : HTML

```
<blockquote>Le Tigre (Panthera tigris) est un mammifère
carnivore...<br>
<small class="pull-right">Wikipedia</small><br>
</blockquote>
```

Le résultat est sobre et élégant (voir figure suivante).

Voici ce qu'en dit le wikipedia :

Le Tigre (*Panthera tigris*) est un mammifère carnivore de la famille des félidés (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands carnivores du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement. Superprédateur, il chasse principalement les cerfs et les sangliers, bien qu'il puisse s'attaquer à des proies de taille plus importante comme les buffles. Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle possède un territoire qui englobe les domaines de plusieurs femelles et ne participe pas à l'éducation des petits.

— Wikipedia

Apparence d'une citation

La référence inférieure « Wikipedia » a été renvoyée à droite avec la classe `pull-right` qui se contente de rendre flottant à droite :

Code : CSS - Classe pull-right

```
.pull-right {
  float: right !important;
}
```

Une adresse

Dans la partie inférieure gauche figure une adresse. J'ai simplement utilisé la balise `<address>` :

Code : HTML

```
<address>
  <p>Vous pouvez me contacter à cette adresse :</p>
  <p><strong>Tigrou Alfred</strong><br>
    Allée des fauves<br>
    28645 Félin-sur-Loire<br>
</address>
```

Pour un résultat satisfaisant, visible à la figure suivante.

Vous pouvez me contacter à cette adresse :

Tigrou Alfred
Allée des fauves
28645 Félin-sur-Loire

Apparence d'une adresse

Juste un peu de style ajouté

Pour donner plus d'harmonie à cette page, j'ai été obligé d'ajouter une marge basse de 20px à tous les `col` et j'ai obligé les images à occuper tout l'espace disponible :

Code : CSS

```
[class*="col"] {
  margin-bottom: 20px;
}
img {
  width: 100%;
}
```

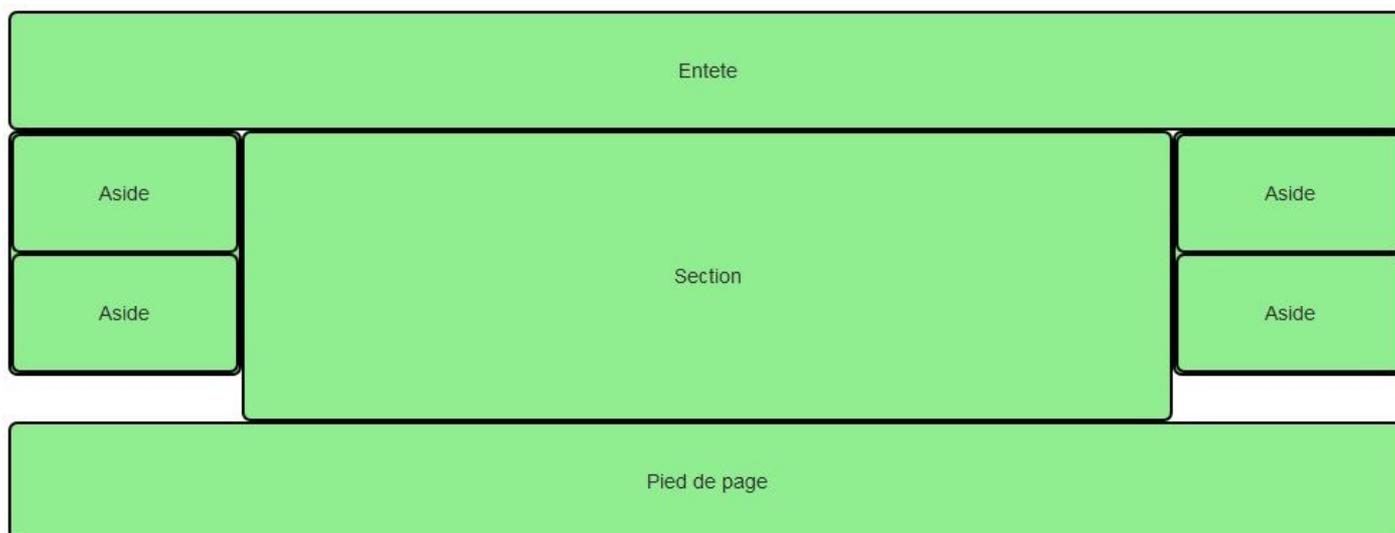
C'est le seul élément de style ajouté à cette page.

Cas pratiques et classes « responsive »

Maintenant, vous avez tous les éléments en main pour construire des pages ! Je vous propose donc quelques exercices pratiques pour vous entraîner.

Exercice 1

Le but est d'obtenir une mise en page pour grand écran identique à celle de la figure suivante.



Résultat à obtenir

Réfléchissez un peu avant de regarder la solution.

Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
    <!-- Un peu de style pour la visualisation -->
    <style type="text/css">
      .col-lg-8 { line-height: 200px; }
      .col-lg-12 { line-height: 80px; }
    </style>
  </head>
  <body>
    <div class="container">

      <header class="row">
        <div class="col-lg-12">
          Entete
        </div>
      </header>

      <div class="row">

        <div class="col-lg-2">
          <div class="row">
            <aside class="col-lg-12">
              Aside
            </aside>
            <aside class="col-lg-12">
              Aside
            </aside>
          </div>
        </div>

        <section class="col-lg-8">
          Section
        </section>

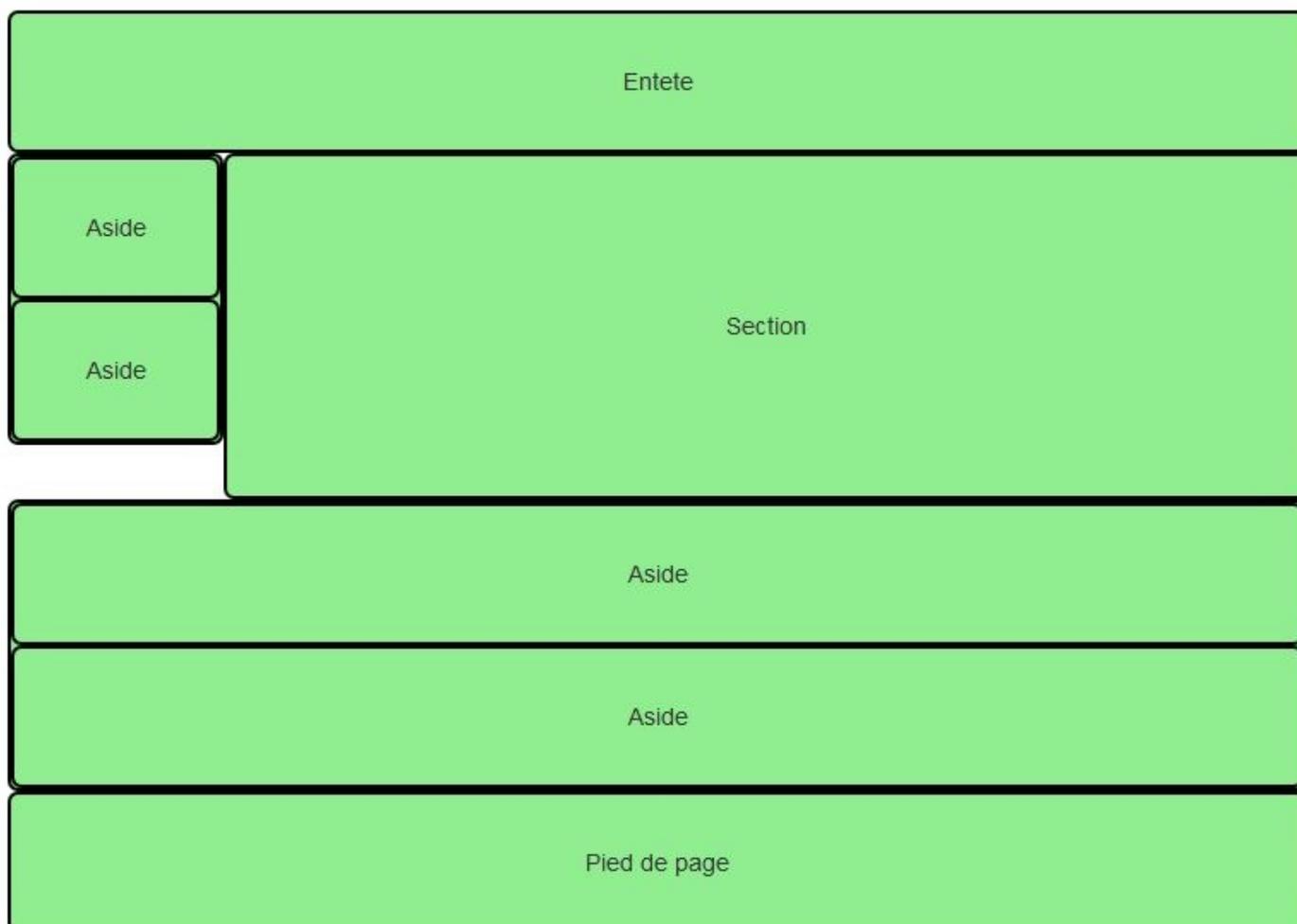
        <div class="col-lg-2">
          <div class="row">
            <aside class="col-lg-12">
              Aside
            </aside>
            <aside class="col-lg-12">
              Aside
            </aside>
          </div>
        </div>
      </div>
    </body>
  </html>

```

```
        Aside
      </aside>
    </div>
  </div>

  <div>
    <div>
      <div class="row">
        <div class="col-lg-12">
          Pied de page
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

Maintenant la question à se poser c'est : comment va réagir cette structure quand on va réduire la fenêtre ? Comme j'ai mis des classes `col-lg-*`, tout va s'empiler sous 1200 pixels. Si cet effet me convient, c'est parfait. Mais supposons que, sur tablette, je veuille une structure similaire à celle de la figure suivante.



Affichage désiré sur tablette

Comment procéder ? Voici une solution :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
```

```

<link href="assets/css/bootstrap.css" rel="stylesheet">
<link href="assets/css/tuto.css" rel="stylesheet">
<!-- Un peu de style pour la visualisation -->
<style type="text/css">
  .col-md-8, .col-sm-10 { line-height: 200px; }
  .col-md-12 { line-height: 80px; }
</style>
</head>
<body>
  <div class="container">

    <header class="row">
      <div class="col-md-12">
        Entete
      </div>
    </header>

    <div class="row">

<div class="col-sm-2">
      <div class="row">
        <aside class="col-md-12">
          Aside 1
        </aside>
        <aside class="col-md-12">
          Aside 2
        </aside>
      </div>
    </div>

<section class="col-sm-10 col-md-8">
      Section
    </section>

<div class="clearfix visible-sm"></div>

    <div class="col-md-2">
      <div class="row">
        <aside class="col-md-12">
          Aside 3
        </aside>
        <aside class="col-md-12">
          Aside 4
        </aside>
      </div>
    </div>

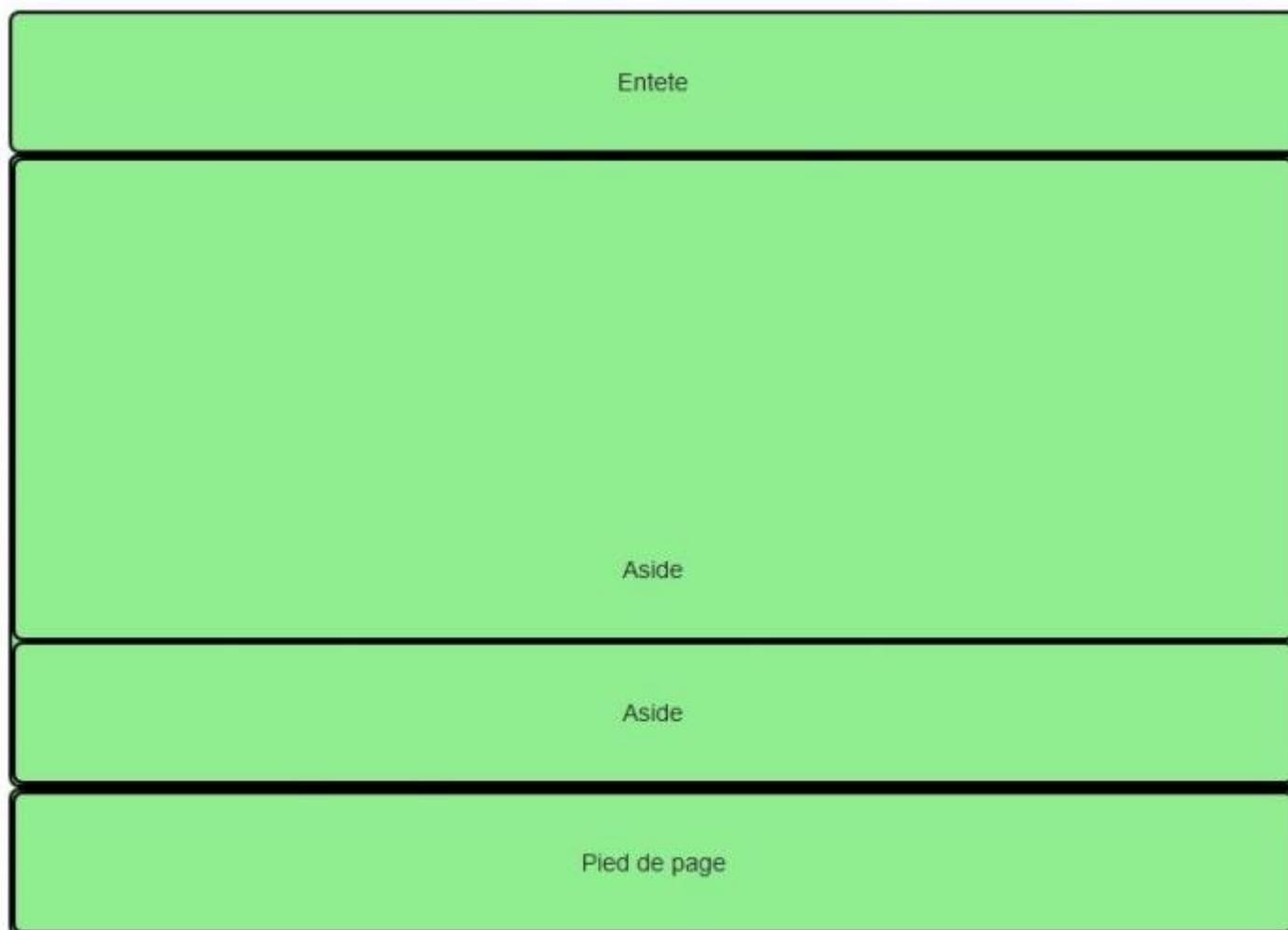
    <div>

    <footer class="row">
      <div class="col-md-12">
        Pied de page
      </div>
    </footer>

  </div>
</body>
</html>

```

J'ai remplacé les classes `col-lg-*` par des classes `col-md-*` pour avoir un comportement de base pour écrans moyens et grands. J'ai ensuite introduit des classes `col-sm-*` pour avoir le comportement désiré sur tablette. En revanche, la ligne 38 mérite quelques commentaires particuliers. Pour le média tablette, vous avez des éléments flottants qui se succèdent avec des comportements qui ne sont pas forcément ceux que l'on souhaite. La figure suivante vous montre le résultat sans cette ligne de code.



Résultat sans la classe « clearfix »

Que fait cette classe `clearfix` ? Regardons le code :

Code : CSS

```
.clearfix:after {  
  clear: both;  
}
```

Autrement dit, on réinitialise les éléments flottants qui suivent pour les ramener dans le flux normal, et éviter qu'ils viennent recouvrir les éléments précédents. Nous allons voir maintenant la classe **visible-sm...**

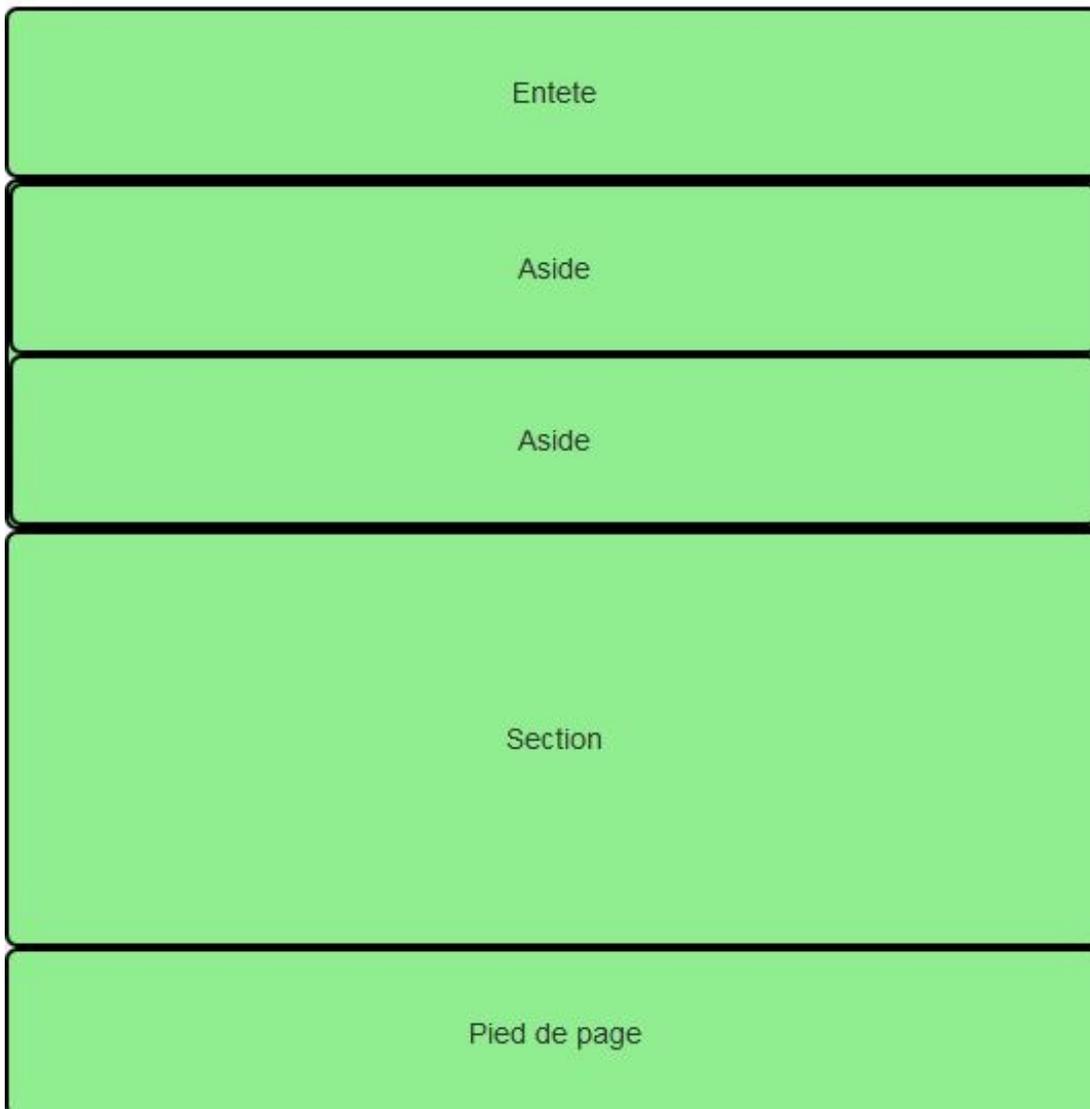
Les classes « responsive »

Pour poursuivre l'exercice précédent, il faut d'abord évoquer quelques classes très utiles qui vont nous permettre non plus de positionner différemment les éléments, mais carrément de les faire disparaître ou apparaître. Vous trouverez un tableau très bien fait [dans la documentation](#), visible à la figure suivante.

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
.visible-xs	Visible	Hidden	Hidden	Hidden
.visible-sm	Hidden	Visible	Hidden	Hidden
.visible-md	Hidden	Hidden	Visible	Hidden
.visible-lg	Hidden	Hidden	Hidden	Visible
.hidden-xs	Hidden	Visible	Visible	Visible
.hidden-sm	Visible	Hidden	Visible	Visible
.hidden-md	Visible	Visible	Hidden	Visible
.hidden-lg	Visible	Visible	Visible	Hidden

Les classes responsives

Nous avons déjà utilisé une de ces classes. Continuons à en utiliser pour poursuivre notre exercice en supprimant 2 `asides` pour les smartphones (voir figure suivante).



Affichage désiré sur

smartphone

Il suffit de changer une ligne, la ligne 40 :

Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
    <!-- Un peu de style pour la visualisation -->
    <style type="text/css">
      .col-md-8, .col-sm-10 { line-height: 200px; }
      .col-md-12 { line-height: 80px; }
    </style>
  </head>
  <body>
    <div class="container">

      <header class="row">
        <div class="col-md-12">
          Entete
        </div>
      </header>

      <div class="row">

        <div class="col-sm-2">
          <div class="row">
            <aside class="col-md-12">
              Aside
            </aside>
            <aside class="col-md-12">
              Aside
            </aside>
          </div>
        </div>

        <section class="col-sm-10 col-md-8">
          Section
        </section>

        <div class="clearfix visible-sm"></div>

        <div class="hidden-xs col-md-2">
          <div class="row">
            <aside class="col-md-12">
              Aside
            </aside>
            <aside class="col-md-12">
              Aside
            </aside>
          </div>
        </div>

      </div>

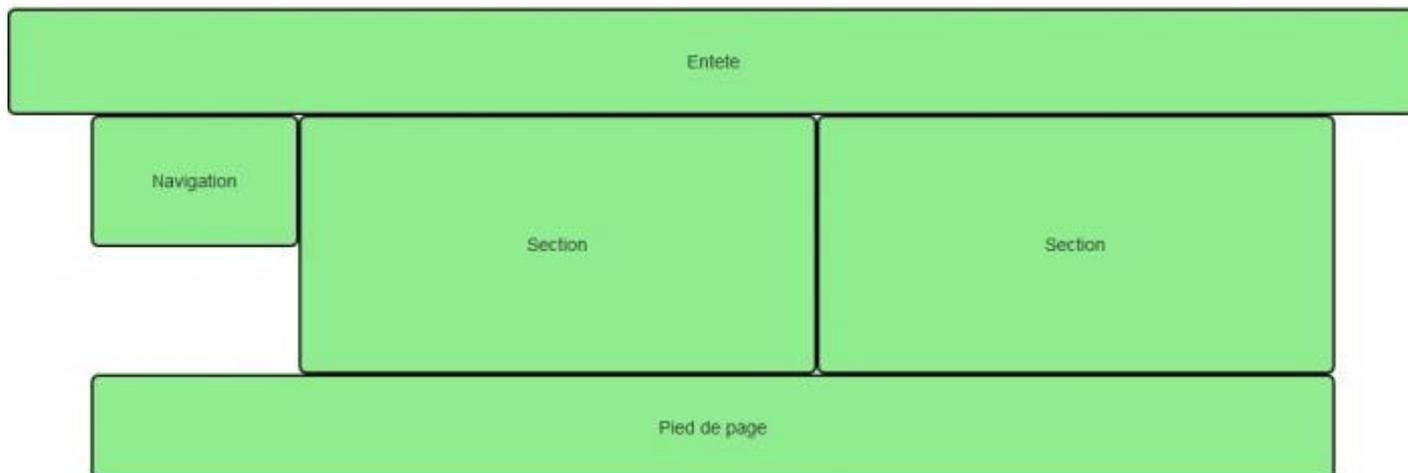
      <footer class="row">
        <div class="col-md-12">
          Pied de page
        </div>
      </footer>

    </div>
  </body>
</html>

```

Exercice 2

Voyons maintenant un autre cas. Je veux obtenir le résultat visible à la figure suivante.



Résultat à obtenir

Donc une entête qui prend toute la largeur de l'écran, une navigation à gauche, deux sections accolées et un pied de page, le tout pour grand écran. Voici une solution :

Code : HTML

```
<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
    <!-- Un peu de style pour la visualisation -->
    <style type="text/css">
      .col-lg-2 { line-height: 100px; }
      .col-lg-5 { line-height: 200px; }
      .col-lg-12 { line-height: 80px; }
    </style>
  </head>
  <body>

    <header>
      <div class="col-lg-12">
        Entete
      </div>
    </header>

    <div class="container">

      <div class="row">

        <nav class="col-lg-2">
          Navigation
        </nav>

        <section class="col-lg-5">
          Section
        </section>
        <section class="col-lg-5">
          Section
        </section>

      </div>

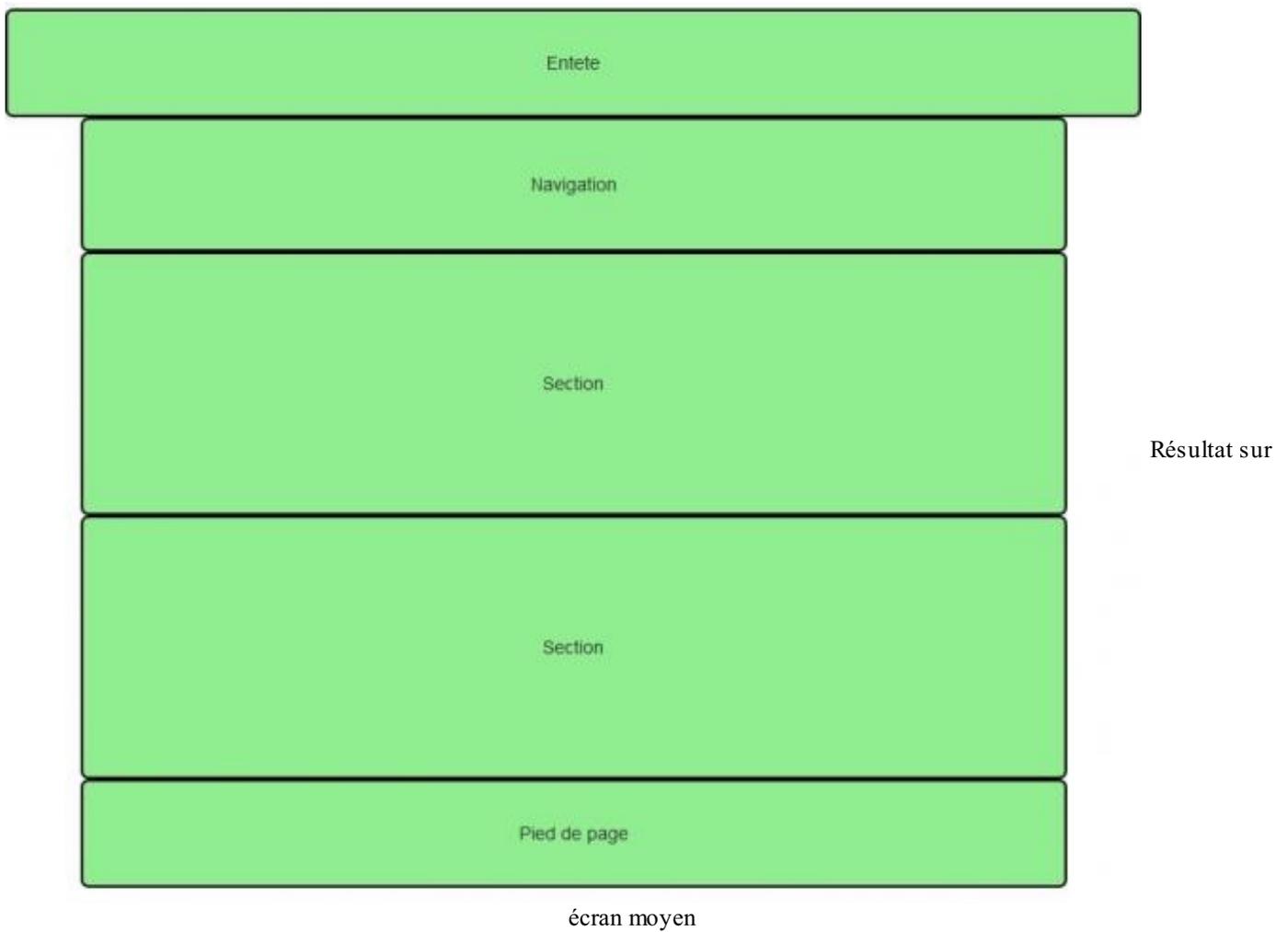
      <footer class="row">
```

```
<div class="col-lg-12">
  Pied de page
</div>
</footer>

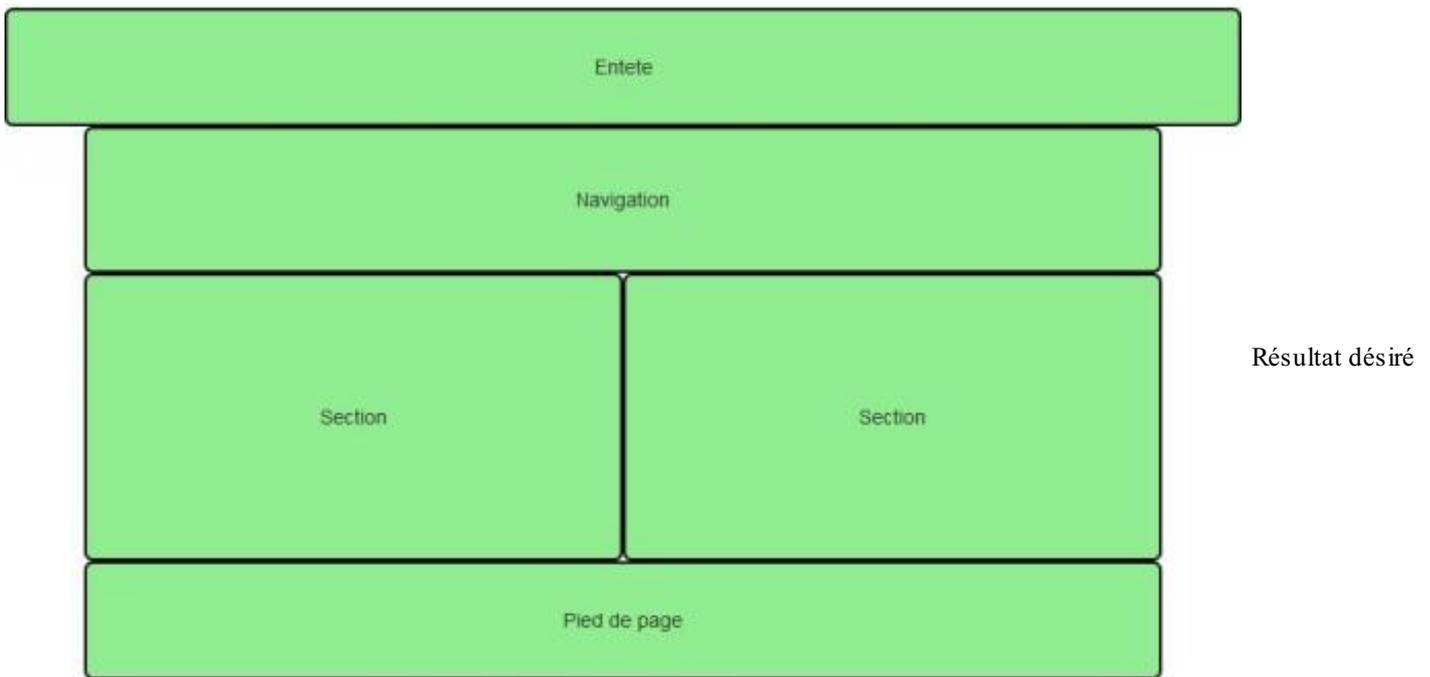
</div>

</body>
</html>
```

Maintenant je me dis que mon application serait bien aussi sur écran moyen. Si je ne fais rien, j'obtiens le résultat visible à la figure suivante.



Ça ne me plaît pas trop, j'aimerais que les 2 sections restent accolées, comme à la figure suivante.



sur écran moyen

Voici une solution :

Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
    <!-- Un peu de style pour la visualisation -->
    <style type="text/css">
      .col-lg-2 { line-height: 100px; }
      .col-lg-5 { line-height: 200px; }
      .col-lg-12 { line-height: 80px; }
    </style>
  </head>
  <body>

    <header>
      <div class="col-lg-12">
        Entete
      </div>
    </header>

    <div class="container">

      <div class="row">

<nav class="col-md-12 col-lg-2">
  Navigation
</nav>

<section class="col-md-6 col-lg-5">
  Section
</section>
<section class="col-md-6 col-lg-5">
  Section
</section>

      </div>

```

```

    <footer class="row">
      <div class="col-lg-12">
        Pied de page
      </div>
    </footer>

  </div>

</body>
</html>

```

Après réflexion, je me dis que mon application irait bien aussi sur tablette, l'empilement des éléments me convient, mais j'aimerais avoir une en-tête différente. Après une nouvelle réflexion, j'en veux aussi une différente sur smartphone. Avec les classes « responsive » vues précédemment, c'est facile à réaliser :

Code : HTML

```

<!DOCTYPE html>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/tuto.css" rel="stylesheet">
    <!-- Un peu de style pour la visualisation -->
    <style type="text/css">
      .col-lg-2 { line-height: 100px; }
      .col-lg-5 { line-height: 200px; }
      .col-lg-12 { line-height: 80px; }
    </style>
  </head>
  <body>

    <header>
    <div class="visible-lg col-lg-12">
      Entete large
    </div>
    <div class="visible-md col-sm-12">
      Entete moyenne
    </div>
    <div class="visible-sm col-12">
      Entete tablette
    </div>
    <div class="visible-xs col-12">
      Entete smartphone
    </div>
  </header>

  <div class="container">

    <div class="row">

      <nav class="col-md-12 col-lg-2">
        Navigation
      </nav>

      <section class="col-md-6 col-lg-5">
        Section
      </section>
      <section class="col-md-6 col-lg-5">
        Section
      </section>

    </div>

    <footer class="row">

```

```
        <div class="col-lg-12">
            Pied de page
        </div>
    </footer>

</div>

</body>
</html>
```

Tester

En résumé :

- Il est possible avec la grille de régler très finement le rendu des pages selon les supports de visualisation.
- La plupart des éléments typographiques du HTML trouvent automatiquement une mise en forme esthétique avec Bootstrap.
- On a vu sur des exemples pratiques qu'il est facile de faire des mises en page même complexes sans trop d'effort.

Partie 2 : Mise en forme des éléments

Dans cette deuxième partie, nous allons voir ensemble comment mettre en forme des éléments de base comme les listes, les tableaux, les formulaires, les boutons, les médias...

Nous verrons aussi comment Bootstrap permet de créer facilement une barre de navigation élégante.

Nous verrons également ce qui crée la "touche" visuelle de Bootstrap : certains effets typographiques, les panneaux et les alertes.

Eléments de base

Un page web contient du texte et des images, mais aussi un certain nombre d'éléments très fréquents : listes, tableaux, formulaires, icônes, boutons. Créer une harmonie dans la présentation des tous ces éléments n'est pas une tâche aisée.

Nous allons voir comment Bootstrap permet de créer un rendu visuel cohérent pour que tous ces éléments cohabitent de façon esthétique.

Listes, descriptions et tableaux

Voici le deuxième jet de la page web de test :

Code : HTML

```
<!DOCTYPE HTML>
<html>

  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      [class*="col"] { margin-bottom: 20px; }
      img { width: 100%; }
    </style>
  </head>

  <body>
    <div class="container">
      <header class="page-header">
        <h1>Mon amour pour les tigres</h1>
      </header>
      <section class="row">
        <div class="col-lg-12">
          <p>
            Je suis passionné par les <strong>tigres</strong> depuis
            très longtemps. Ce site a été construit en <em>hommage à ces
            merveilleux félins...</em><br>
            Je fais partie de la <abbr title="Société des Amoureux
            des Tigres">SAT</abbr> qui a pour but de faire connaître ces
            splendides animaux.
          </p>
          <p>Voici ce qu'en dit le wikipedia :</p>
          <blockquote>
            Le Tigre (Panthera tigris) est un mammifère carnivore
            de la famille des félidés (Felidae) du genre Panthera.
            Aisément reconnaissable à sa fourrure rousse rayée de noir, il est
            le plus grand félin sauvage et l'un des plus grands carnivores du
            monde.
            L'espèce est divisée en neuf sous-espèces possédant des différences
            mineures en termes de taille ou de comportement. Superprédateur,
            il chasse principalement les cerfs et les sangliers, bien qu'il
            puisse s'attaquer à des proies de taille plus importante comme les
            buffles.
            Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La
            structure sociale des tigres en fait un animal solitaire ;
            le mâle possède un territoire qui englobe les domaines de plusieurs
            femelles et ne participe pas à l'éducation des petits.<br>
          </blockquote>
        </div>
      </section>
    </div>
  </body>
</html>
```

```

        <small class="pull-right">Wikipedia</small><br>
    </blockquote>
</div>
</section>
<section class="row">
    <div class="col-xs-4 col-sm-3 col-md-2"></div>
    <div class="col-xs-4 col-sm-3 col-md-2"></div>
</section>
<div class="row">
    <section class="col-sm-4">
        <h5>Voici les sous-espèces des tigres :</h5>
        <br>
        <ul>
            <li>Tigre de Sibérie</li>
            <li>Tigre de Chine méridionale</li>
            <li>Tigre de Bali</li>
            <li>Tigre de d'Indochine</li>
            <li>Tigre de Malaisie</li>
            <li>Tigre de Java</li>
            <li>Tigre de Sumatra</li>
            <li>Tigre du Bengale</li>
            <li>Tigre de la Caspienne</li>
        </ul>
    </section>
    <section class="col-sm-8">
        
    </section>
</div>
<div class="row">
    <section class="col-sm-8 table-responsive">
        <table class="table table-bordered table-striped table-
condensed">
            <caption>
                <h4>Les menaces pour les tigres</h4>
            </caption>
            <thead>
                <tr>
                    <th>Lieu</th>
                    <th>Menace</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Grand Mekong</td>
                    <td>Demande croissante de certaines parties de
l'animal pour la médecine chinoise traditionnelle
et fragmentation des habitats du fait du développement non durable
d'infrastructures</td>
                </tr>
            </tbody>
        </table>
    </section>
</div>

```

```

        </tr>
        <tr>
            <td>Île de Sumatra</td>
            <td>Production d'huile de palme et de pâtes à
papiers</td>
        </tr>
        <tr>
            <td>Indonésie et Malaisie</td>
            <td>Pâte à papier, l'huile de palme et le
caoutchouc</td>
        </tr>
        <tr>
            <td>États-Unis</td>
            <td>Les tigres captifs représentent un danger pour
les tigres sauvages</td>
        </tr>
        <tr>
            <td>Europe</td>
            <td>Gros appétit pour l'huile de palme</td>
        </tr>
        <tr>
            <td>Népal</td>
            <td>Commerce illégal de produits dérivés de
tigres</td>
        </tr>
    </tbody>
</table>
</section>
<section class="col-sm-4">
    <address>
        <p>Vous pouvez me contacter à cette adresse :</p>
        <strong>Tigrou Alfred</strong><br>
        Allée des fauves<br>
        28645 Félines-sur-Loire<br>
    </address>
</section>
</div>
</div>
</body>
</html>

```

Ce qui donne la figure suivante.

Mon amour pour les tigres

Je suis passionné par les tigres depuis très longtemps. Ce site a été construit en hommage à ces merveilleux félins...
Je fais partie de la SAT qui a pour but de faire connaître ces splendides animaux.

Voici ce qu'en dit le wikipedia :

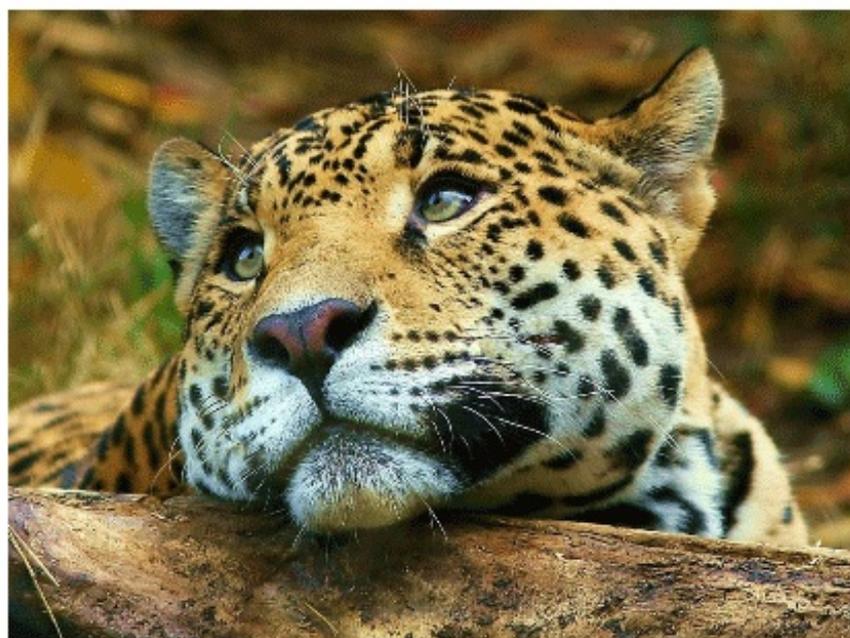
Le Tigre (*Panthera tigris*) est un mammifère carnivore de la famille des félinés (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands carnivores du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement. Superprédateur, il chasse principalement les cerfs et les sangliers, bien qu'il puisse s'attaquer à des proies de taille plus importante comme les buffles. Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle possède un territoire qui englobe les domaines de plusieurs femelles et ne participe pas à l'éducation des petits.

— Wikipedia



Voici les sous-espèces des tigres :

- Tigre de Sibérie
- Tigre de Chine méridionale
- Tigre de Bali
- Tigre de d'Indochine
- Tigre de Malaisie
- Tigre de Java
- Tigre de Sumatra
- Tigre du Bengale
- Tigre de la Caspienne



La

Les menaces pour les tigres

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc
États-Unis	Les tigres captifs représentent un danger pour les tigres sauvages
Europe	Gros appétit pour l'huile de palme
Népal	Commerce illégal de produits dérivés de tigres

Vous pouvez me contacter à cette adresse :

Tigrou Alfred
Allée des fauves
28045 Félin-sur-Loire

page d'exemple

Tester

La présentation s'est enrichie d'une rangée et de nouveaux éléments. Voici à la figure suivante une visualisation du découpage.

page-header **Mon amour pour les tigres**

row **col-lg-12**

row

row **col-sm-4** **col-sm-8**

row **col-sm-8** **col-sm-4**

Je suis passionné par les tigres depuis très longtemps. Ce site a été construit en hommage à ces merveilleux félins...
 le fera partie de la SAT qui a pour but de faire connaître ces splendides animaux.
 Voici ce qu'en dit le Wikipédia

Le Tigre (*Panthera tigris*) est un mammifère carnivore de la famille des félins, originaire du genre *Panthera*. Anciennement reconnue comme à sa fourrure, cette espèce de félin est le plus grand félin existant au monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de coloration. Super-prédateur, il chasse principalement les cerfs et les sangliers, bien qu'il ait été accusé à plusieurs reprises de tuer des humains, notamment les bœufs. Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle possède un territoire qui englobe les domaines de plusieurs femelles et ne participe pas à l'éducation des petits.

— Wikipédia



Voici les sous-espèces des tigres :

- Tigre de Sibérie
- Tigre de Chine méridionale
- Tigre de Bali
- Tigre de d'Indochine
- Tigre de Malaisie
- Tigre de Java
- Tigre de Sumatra
- Tigre du Bengale
- Tigre de la Caspienne

Les menaces pour les tigres

Lieu	Menace
Asie du Sud-Est	Commerce illégal de certaines parties de tigre pour la médecine traditionnelle chinoise et fragmentation des habitats du fait du développement non durable d'infrastructures
Asie de Sumatra	Production d'huile de palme et de cacao à grande échelle
Indonésie et Malaisie	Pêche à papier, fraude de papier
État d'Israël	Les tigres captifs représentent un danger pour les tigres sauvages
Europe	Gros appât pour fraude de palme
Népal	Commerce illégal de produits dérivés de tigre

Une adresse ma intention à votre adresse :
 Tigres d'Inde
 Allée des Feuilles
 33045 Pîtres-sur-Loire

La page d'exemple décryptée

Les listes

Style de base

Bootstrap propose une mise en forme élégante des listes. J'en ai prévu une sur la page :

Code : HTML

```
<h5>Voici les sous-espèces des tigres :</h5>
<br>
<ul>
  <li>Tigre de Sibérie</li>
  <li>Tigre de Chine méridionale</li>
  <li>Tigre de Bali</li>
  <li>Tigre de d'Indochine</li>
  <li>Tigre de Malaisie</li>
  <li>Tigre de Java</li>
  <li>Tigre de Sumatra</li>
  <li>Tigre du Bengale</li>
  <li>Tigre de la Caspienne</li>
</ul>
```

Ce qui donne le résultat visible à la figure suivante.

Voici les sous-espèces des tigres :

- Tigre de Sibérie
 - Tigre de Chine méridionale
 - Tigre de Bali
 - Tigre de d'Indochine
 - Tigre de Malaisie
 - Tigre de Java
 - Tigre de Sumatra
 - Tigre du Bengale
 - Tigre de la Caspienne
- Une liste

Sans style

Il existe la classe `list-unstyled`, à appliquer à la balise ``, ce qui donne donc `<ul class="list-unstyled">`, qui supprime les puces. Voici à la figure suivante le résultat sur la liste de la page.

Voici les sous-espèces des tigres :

Tigre de Sibérie
 Tigre de Chine méridionale
 Tigre de Bali
 Tigre de d'Indochine
 Tigre de Malaisie
 Tigre de Java
 Tigre de Sumatra
 Tigre du Bengale
 Tigre de la Caspienne

Une liste sans style

En ligne

Il peut arriver de vouloir une liste... en ligne. Bon je n'appelle plus ça une liste mais pourquoi pas ? Il suffit d'utiliser la classe `list-inline`, donc `<ul class="list-inline">`. Voilà à la figure suivante ce que ça donne sur la page.

Voici les sous-espèces des tigres :

Tigre de Sibérie Tigre de Chine méridionale
 Tigre de Bali Tigre de d'Indochine Tigre de Malaisie
 Tigre de Java Tigre de Sumatra Tigre du Bengale
 Tigre de la Caspienne

Une liste en ligne

Descriptions

Classique

Les descriptions sont aussi bien traitées par Bootstrap. Voici un exemple :

Code : HTML

```
<dl>
  <dt>Tigre</dt>
  <dd>Animal d'Asie, grand et puissant, au pelage jaune rayé de
  noir, de la famille des félins</dd>
```

```

<dt>Panthère</dt>
<dd>Félin d'Afrique et d'Asie au pelage jaune tâcheté de
noir</dd>
<dt>Guépard</dt>
<dd>Félin carnassier rapide d'Asie et d'Afrique au pelage tâcheté
de noir</dd>
<dt>Chat</dt>
<dd>Petit félin, au pelage soyeux, en général domestique</dd>
</dl>

```

Et l'effet visuel à la figure suivante.

Tigre

Animal d'Asie, grand et puissant, au pelage jaune rayé de noir, de la famille des félins

Panthère

Félin d'Afrique et d'Asie au pelage jaune tâcheté de noir

Guépard

Félin carnassier rapide d'Asie et d'Afrique au pelage tâcheté de noir

Chat

Petit félin, au pelage soyeux, en général domestique

Une description

verticale

Horizontal

Pour obtenir une présentation horizontale, il suffit d'utiliser la classe `dl-horizontal`, ce qui donne `<dl class="dl-horizontal">`. Voici à la figure suivante l'effet sur la même liste.

Tigre Animal d'Asie, grand et puissant, au pelage jaune rayé de noir, de la famille des félins

Panthère Félin d'Afrique et d'Asie au pelage jaune tâcheté de noir

Guépard Félin carnassier rapide d'Asie et d'Afrique au pelage tâcheté de noir

Chat Petit félin, au pelage soyeux, en général domestique

Une description horizontale

Les tableaux

Plusieurs mises en forme sont prévues pour les tableaux, j'en ai retenu une pour le tableau de la page :

Code : HTML

```

<table class="table table-bordered table-striped table-condensed">
  <caption>
    <h4>Les menaces pour les tigres</h4>
  </caption>
  <thead>
    <tr>
      <th>Lieu</th>
      <th>Menace</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Grand Mekong</td>
      <td>Demande croissante de certaines parties de l'animal

```

```

pour la médecine chinoise traditionnelle
et fragmentation des habitats du fait du développement non durable
d'infrastructures</td>
</tr>
<tr>
<td>Île de Sumatra</td>
<td>Production d'huile de palme et de pâtes à
papiers</td>
</tr>
<tr>
<td>Indonésie et Malaisie</td>
<td>Pâte à papier, l'huile de palme et le
caoutchouc</td>
</tr>
<tr>
<td>États-Unis</td>
<td>Les tigres captifs représentent un danger pour les
tigres sauvages</td>
</tr>
<tr>
<td>Europe</td>
<td>Gros appétit pour l'huile de palme</td>
</tr>
<tr>
<td>Népal</td>
<td>Commerce illégal de produits dérivés de tigres</td>
</tr>
</tbody>
</table>

```

Ce qui nous donne le résultat visible à la figure suivante.

Les menaces pour les tigres

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc
États-Unis	Les tigres captifs représentent un danger pour les tigres sauvages
Europe	Gros appétit pour l'huile de palme
Népal	Commerce illégal de produits dérivés de tigres

Un

tableau stylisé

On va analyser les classes utilisées...

Classe « *table* »

C'est la classe de base pour les tableaux, elle fixe quelques styles et crée des lignes de séparation horizontales :

Code : CSS

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

Ce qui donne la figure suivante.

Les menaces pour les tigres

Lieu	Menace	
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures	Effet
Île de Sumatra	Production d'huile de palme et de pâtes à papiers	

de la classe table

Classe « *table-bordered* »

C'est la classe pour avoir de jolies bordures avec des coins arrondis :

Code : CSS

```
<table class="table table-bordered">
```

Ce qui donne la figure suivante.

Les menaces pour les tigres

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers

Effet de la classe table-bordered

Classe « *table-striped* »

Cette classe ajoute un fond avec un gris très léger pour les lignes impaires (1, 3, ...) :

Code : CSS

```
<table class="table table-bordered table-striped">
```

Je ne vous mets pas d'illustration parce que franchement, le gris est si léger qu'on ne voit rien. Quand je regarde la couleur utilisée, je trouve la valeur #F9F9F9, ce qui est effectivement un gris très très léger... Vous pouvez remarquer aussi que, lorsqu'on passe le curseur au-dessus du tableau, la couleur de fond de la ligne correspondante change avec la valeur #F5F5F5, ce qui est déjà un gris assez léger. Je pense que des valeurs un peu plus marquées auraient été souhaitables. Mais rien n'empêche de les changer ! Pour la page d'exemple, je suis passé par une autre approche qui apparaîtra dans un chapitre ultérieur.

Classe « *table-condensed* »

Cette classe permet de condenser le tableau pour qu'il occupe moins de place verticale :

Code : CSS

```
<table class="table table-bordered table-striped table-condensed">
```

Ce qui donne la figure suivante.

Les menaces pour les tigres

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc

Effet

Vous avez remarqué qu'on peut cumuler l'application des classes sans problème, comme je l'ai fait dans cette démonstration progressive.

Tableau adaptable avec la classe « table-responsive »

Que se passe-t-il pour un tableau quand on réduit la largeur de l'affichage ? Les colonnes du tableau se réduisent en conséquence jusqu'à rendre l'apparence ni très esthétique, ni très lisible de la figure suivante.

Les menaces pour les tigres

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc

Effet sans la classe table-responsive

En englobant le tableau avec la classe `table-responsive`, on fait apparaître une barre de défilement (voir figure suivante).

Code : HTML

```
<section class="col-sm-8 table-responsive">
  <table class="table table-bordered table-striped table-condensed">
```

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc
États-Unis	Les tigres captifs représentent un danger pour les tigres sauvages
Europe	Gros appétit pour l'huile de palme
Népal	Commerce illégal de produits dérivés de tigres

Barre de défilement avec la classe

table-responsive

Un peu de couleur dans les tableaux

Vous avez la possibilité de colorer le fond des lignes des tableaux avec les classes `info`, `success`, `danger`, `warning` et `active`. Voici le tableau de notre exemple, mais avec un peu de couleur :

Code : HTML

```
<table class="table table-bordered table-striped table-condensed">
  <caption>
    <h4>Les menaces pour les tigres</h4>
  </caption>
  <thead>
    <tr>
      <th>Lieu</th>
      <th>Menace</th>
    </tr>
  </thead>
  <tbody>
    <tr class="success">
      <td>Grand Mekong</td>
      <td>Demande croissante de certaines parties de l'animal pour
la médecine chinoise traditionnelle et
fragmentation des habitats du fait du développement non durable
d'infrastructures</td>
    </tr>
    <tr class="danger">
      <td>Île de Sumatra</td>
      <td>Production d'huile de palme et de pâtes à papiers</td>
    </tr>
    <tr class="warning">
      <td>Indonésie et Malaisie</td>
      <td>Pâte à papier, l'huile de palme et le caoutchouc</td>
    </tr>
    <tr class="active">
      <td>États-Unis</td>
      <td>Les tigres captifs représentent un danger pour les
tigres sauvages</td>
    </tr>
    <tr class="success">
      <td>Europe</td>
      <td>Gros appétit pour l'huile de palme</td>
    </tr>
    <tr class="danger">
      <td>Népal</td>
      <td>Commerce illégal de produits dérivés de tigres</td>
    </tr>
  </tbody>
</table>
```

Ce qui donne la figure suivante.

Les menaces pour les tigres

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc
États-Unis	Les tigres captifs représentent un danger pour les tigres sauvages
Europe	Gros appétit pour l'huile de palme
Népal	Commerce illégal de produits dérivés de tigres

De la couleur dans les tableaux

Mais évidemment, cette possibilité de colorisation prend tout son sens dans une gestion dynamique de données, pour faire apparaître par exemple des enregistrements possédant certaines caractéristiques.

Formulaires

Formulaire de base

Si on n'utilise aucune classe particulière dans une balise `<form>`, on obtient une disposition en ligne. Prenons l'exemple d'un formulaire simple :

Code : HTML

```
<form>
  <legend>Légende</legend>
  Text : <input type="text">
  Textarea : <textarea id="textarea"></textarea>
  Select :
    <select>
      <option>Option 1</option>
      <option>Option 2</option>
      <option>Option 3</option>
    </select>
  <button>Envoyer</button>
</form>
```

Nous aurons alors un rendu similaire à celui de la figure suivante.

Légende

Text : Textarea :

Select : Option 1 ▾

Un formulaire sans style

Par défaut les contrôles s'alignent horizontalement. J'aimerais que les contrôles occupent toute la largeur disponible. Il faut utiliser la classe `form-control` pour obtenir cet effet (voir figure suivante) :

Code : HTML

```
<form class="col-lg-6">
  <legend>Légende</legend>
  Text : <input type="text" class="form-control">
  Textarea : <textarea id="textarea" class="form-control"></textarea>
  Select :
  <select class="form-control">
    <option>Option 1</option>
    <option>Option 2</option>
    <option>Option 3</option>
  </select>
  <button>Envoyer</button>
</form>
```

Légende

Text :

Textarea :

Effet de la classe

Select :

Envoyer

`form-control`

On se rend compte que cette classe ne se contente pas d'élargir les contrôles, elle procède aussi à une mise en forme esthétique. Arrangeons encore ce formulaire en utilisant des balises `<label>` et écartons un peu le bouton :

Code : HTML

```
<form class="col-lg-6">
  <legend>Légende</legend>
  <label for="texte">Text : </label>
  <input id="text" type="text" class="form-control">
  <label for="textarea">Textarea : </label>
  <textarea id="textarea" type="text" class="form-control"></textarea>
  <label for="select">Select : </label>
  <select id="select" class="form-control">
    <option>Option 1</option>
    <option>Option 2</option>
    <option>Option 3</option>
  </select>
  <br>
  <button>Envoyer</button>
</form>
```

Le résultat se trouve à la figure suivante.

Légende

Text :

Textarea :

Ajout de balises

Select :

Envoyer

<label>

On a gagné en esthétique. On peut encore améliorer l'espacement en utilisant la classe `form-group` :

Code : HTML

```
<form class="col-lg-6">
  <legend>Légende</legend>
  <div class="form-group">
    <label for="texte">Text : </label>
    <input id="text" type="text" class="form-control">
  </div>
  <div class="form-group">
    <label for="textarea">Textarea : </label>
    <textarea id="textarea" type="textarea" class="form-
control"></textarea>
  </div>
  <div class="form-group">
    <label for="select">Select : </label>
    <select id="select" class="form-control">
      <option>Option 1</option>
      <option>Option 2</option>
      <option>Option 3</option>
    </select>
  </div>
  <button>Envoyer</button>
</form>
```

On obtient alors le résultat visible à la figure suivante.

Légende

Text :

Textarea :

Effet de la classe

Select :

form-group

Le formulaire est bien aéré et j'ai pu enlever la balise `
` qui me servait à espacer le bouton.

La page d'exemple

Voici la troisième version de notre page consacrée aux tigres :

Code : HTML

```

<!DOCTYPE HTML>
<html>

  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      body { background-color:#DDD; }
      [class*="col"] { margin-bottom: 20px; }
      img { width: 100%; }
      .well {
        background-color:#CCC;
        padding: 20px;
      }
    </style>
  </head>

  <body>
    <div class="container">
      <header class="page-header">
        <h1>Mon amour pour les tigres</h1>
      </header>
      <section class="row">
        <div class="col-lg-12">
          <p>
            Je suis passionné par les <strong>tigres</strong> depuis
            très longtemps. Ce site a été construit en <em>hommage à ces
            merveilleux félins...</em><br>
            Je fais partie de la <abbr title="Société des Amoureux
            des Tigres">SAT</abbr> qui a pour but de faire connaître ces
            splendides animaux.
          </p>

```

```

    <p>Voici ce qu'en dit le wikipedia :</p>
    <blockquote>
      Le Tigre (Panthera tigris) est un mammifère carnivore
      de la famille des félidés (Felidae) du genre Panthera.
      Aisément reconnaissable à sa fourrure rousse rayée de noir, il est
      le plus grand félin sauvage et l'un des plus grands carnivores du
      monde.
      L'espèce est divisée en neuf sous-espèces possédant des différences
      mineures en termes de taille ou de comportement. Superprédateur,
      il chasse principalement les cerfs et les sangliers, bien qu'il
      puisse s'attaquer à des proies de taille plus importante comme les
      buffles.
      Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La
      structure sociale des tigres en fait un animal solitaire ;
      le mâle possède un territoire qui englobe les domaines de plusieurs
      femelles et ne participe pas à l'éducation des petits.<br>
      <small class="pull-right">Wikipedia</small><br>
    </blockquote>
  </div>
</section>
<section class="row">
  <div class="col-xs-4 col-sm-3 col-md-2"></div>
  <div class="col-xs-4 col-sm-3 col-md-2"></div>
</section>
<div class="row">
  <section class="col-sm-4">
    <h5>Voici les sous-espèces des tigres :</h5>
    <br>
    <ul>
      <li>Tigre de Sibérie</li>
      <li>Tigre de Chine méridionale</li>
      <li>Tigre de Bali</li>
      <li>Tigre de d'Indochine</li>
      <li>Tigre de Malaisie</li>
      <li>Tigre de Java</li>
      <li>Tigre de Sumatra</li>
      <li>Tigre du Bengale</li>
      <li>Tigre de la Caspienne</li>
    </ul>
  </section>
  <section class="col-sm-8">
    
  </section>
</div>
<div class="row">
  <section class="col-sm-12">
    <table class="table table-bordered table-striped table-
condensed">

```

```

<caption>
<h4>Les menaces pour les tigres</h4>
</caption>
<thead>
  <tr>
    <th>Lieu</th>
    <th>Menace</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td>Grand Mekong</td>
    <td>Demande croissante de certaines parties de
l'animal pour la médecine chinoise traditionnelle et fragmentation
des habitats
du fait du développement non durable d'infrastructures</td>
  </tr>
  <tr>
    <td>Île de Sumatra</td>
    <td>Production d'huile de palme et de pâtes à
papiers</td>
  </tr>
  <tr>
    <td>Indonésie et Malaisie</td>
    <td>Pâte à papier, l'huile de palme et le
caoutchouc</td>
  </tr>
  <tr>
    <td>États-Unis</td>
    <td>Les tigres captifs représentent un danger pour
les tigres sauvages</td>
  </tr>
  <tr>
    <td>Europe</td>
    <td>Gros appétit pour l'huile de palme</td>
  </tr>
  <tr>
    <td>Népal</td>
    <td>Commerce illégal de produits dérivés de
tigres</td>
  </tr>
</tbody>
</table>
</section>
</div>
<div class="row">
  <section class="col-sm-8">
    <form class="well">
      <legend>Si vous voulez me laisser un message</legend>
      <h4>Comment m'avez-vous trouvé ?</h4>
      <fieldset>
        <label for="ami" class="radio">
          <input type="radio" name="origine" value="ami"
id="ami">
          Par un ami
        </label>
        <label for="web" class="radio">
          <input type="radio" name="origine" value="web"
id="web">
          Sur le web
        </label>
        <label for="hasard" class="radio">
          <input type="radio" name="origine" value="hasard"
id="hasard">
          Par hasard
        </label>
        <label for="autre" class="radio">
          <input type="radio" name="origine" value="autre"
id="autre">
          Autre...
      </fieldset>
    </form>
  </section>
</div>

```

```
        </label>
        <label for="textarea">Votre message :</label>
        <textarea id="textarea" rows="4" class="form-
control"></textarea>
        <p class="help-block">Vous pouvez agrandir la
fenêtre</p>
        <button class="btn btn-primary"
type="submit">Envoyer</button>
    </fieldset>
</form>
</section>
<section class="col-sm-4">
    <address>
        <p>Vous pouvez me contacter à cette adresse :</p>
        <strong>Tigrou Alfred</strong><br>
        Allée des fauves<br>
        28645 Félin-sur-Loire<br>
    </address>
</section>
</div>
</div>
</body>
</html>
```

Pour cette troisième version, j'ai ajouté un peu de style et un formulaire, ce qui donne les figures suivantes.

base02.html x

file:///C:/wamp/www/bootstrap/tuto%20V3/base02.html

Mon amour pour les tigres

Je suis passionné par les **tigres** depuis très longtemps. Ce site a été construit en *hommage* à ces merveilleux félins...
Je fais partie de la SAT qui a pour but de faire connaître ces splendides animaux.

Voici ce qu'en dit le wikipedia :

Le Tigre (*Panthera tigris*) est un mammifère carnivore de la famille des félidés (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands carnivores du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement. Superprédateur, il chasse principalement les cerfs et les sangliers, bien qu'il puisse s'attaquer à des proies de taille plus importante comme les buffles. Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle possède un territoire qui englobe les domaines de plusieurs femelles et ne participe pas à l'éducation des petits.

— Wikipedia

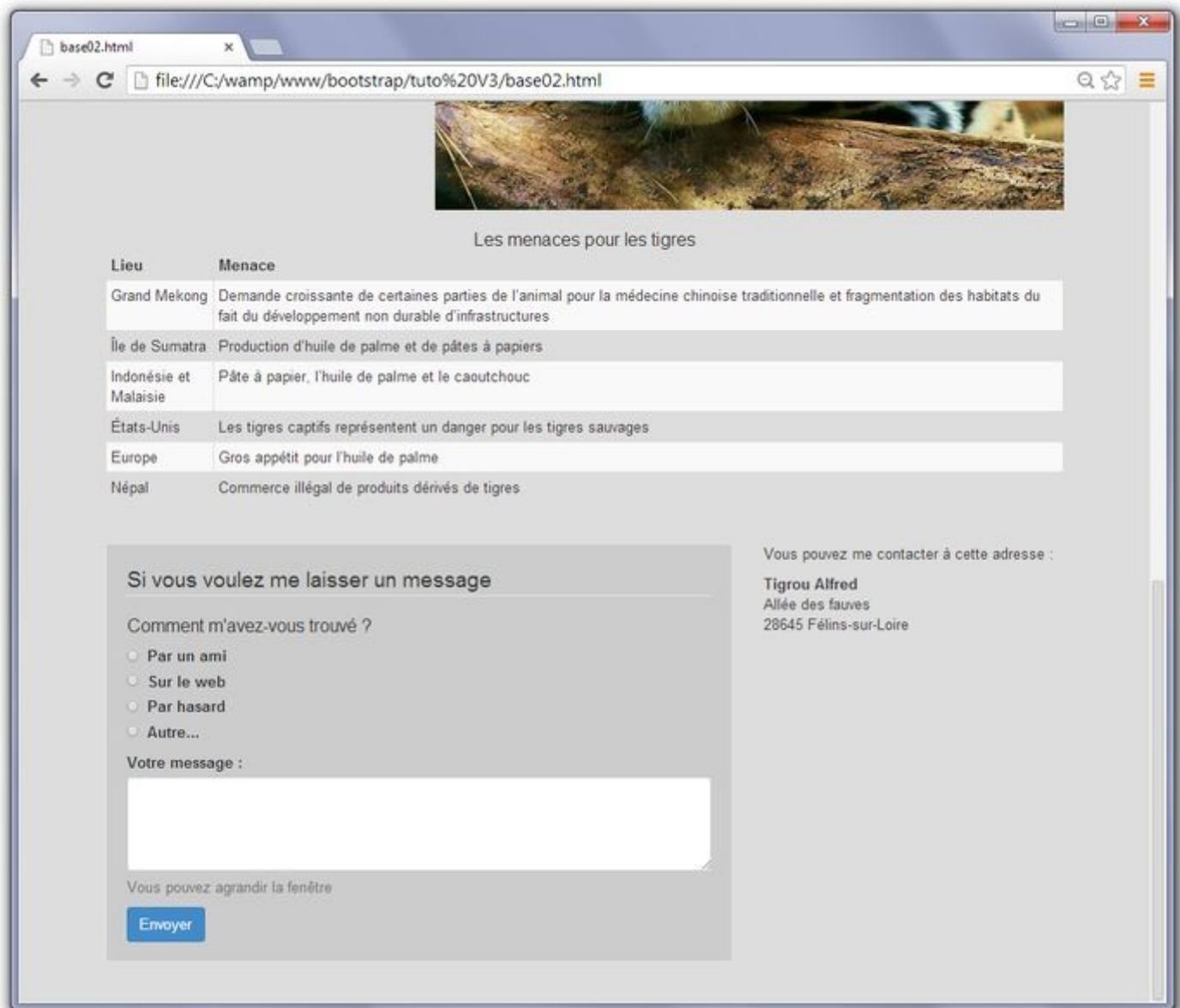


Voici les sous-espèces des tigres :

- Tigre de Sibérie
- Tigre de Chine méridionale
- Tigre de Bali
- Tigre de d'Indochine
- Tigre de Malaisie
- Tigre de Java
- Tigre de Sumatra
- Tigre du Bengale
- Tigre de la Caspienne



La page d'exemple, première partie



La page d'exemple, deuxième partie

Tester

Au niveau du style, juste une petite touche :

Code : CSS

```
body { background-color:#DDD; }
[class*="col"] { margin-bottom: 20px; }
img { width: 100%; }
.well {
  background-color:#CCC;
  padding: 20px;
}
```

Le fond d'un léger gris met bien en valeur le texte et les photos. Quant au formulaire (concerné par la classe `well`), c'est l'objet de ce chapitre...

Les classes « radio » et « checkbox »

Bootstrap propose l'utilisation d'une classe particulière pour les cases à cocher (checkbox) et les zones d'option (radio). Nous

l'avons utilisé dans le formulaire de la page :

Code : HTML

```
<form class="well">
  <legend>Si vous voulez me laisser un message</legend>
  <h4>Comment m'avez-vous trouvé ?</h4>
  <fieldset>
    <label for="ami" class="radio">
      <input type="radio" name="origine" value="ami" id="ami">
        Par un ami
    </label>
    <label for="web" class="radio">
      <input type="radio" name="origine" value="web" id="web">
        Sur le web
    </label>
    <label for="hasard" class="radio">
      <input type="radio" name="origine" value="hasard" id="hasard">
        Par hasard
    </label>
    <label for="autre" class="radio">
      <input type="radio" name="origine" value="autre" id="autre">
        Autre...
    </label>
    <label for="textarea">Votre message :</label>
    <textarea id="textarea" class="form-control"
rows="4"></textarea>
    <p class="help-block">Vous pouvez agrandir la fenêtre</p>
    <button class="btn btn-primary" type="submit">Envoyer</button>
  </fieldset>
</form>
```

La classe « well », un peu de style, un joli bouton et un texte d'aide

Voilà une classe que j'aime bien, elle provoque une jolie stylisation. J'ai apporté ce petit complément au formulaire de la page :

Code : CSS

```
<form class="well">
```

On peut ajuster la dimension en ajoutant la classe `well-lg` pour agrandir ou `well-sm` pour réduire. J'ai aussi peaufiné en ajoutant des règles de style à la page :

Code : CSS

```
<style type="text/css">
body { background-color:#DDD; }
[class*="col"] { margin-bottom: 20px; }
img { width: 100%; }
.well {
background-color:#CCC;
padding: 20px;
}
</style>
```

La dernière touche a consisté à améliorer l'aspect du bouton avec 2 classes que nous verrons plus loin dans ce cours :

Code : HTML

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

Il ne reste plus qu'à rendre plus discret le texte d'aide situé au-dessous de la zone de texte, ce que nous accomplissons avec la classe `help-block` (voir figure suivante) :

Code : HTML

```
<p class="help-block">Vous pouvez agrandir la fenêtre</p>
```

Votre message :

Vous pouvez agrandir la fenêtre

Effet de la classe `help-block`

Formulaire horizontal

Pour réaliser un formulaire horizontal, nous allons utiliser tout ce que nous venons de voir en exploitant la classe `form-horizontal` et une grille pour positionner les étiquettes et les contrôles :

Code : HTML

```
<form class="form-horizontal col-lg-6">
  <div class="form-group">
    <legend>Légende</legend>
  </div>
  <div class="row">
    <div class="form-group">
      <label for="text" class="col-lg-2">Text : </label>
      <div class="col-lg-10">
        <input type="text" class="form-control" id="text">
      </div>
    </div>
  </div>
  <div class="row">
    <div class="form-group">
      <label for="textarea" class="col-lg-2">Textarea : </label>
      <div class="col-lg-10">
        <input type="textarea" class="form-control" id="textarea">
      </div>
    </div>
  </div>
  <div class="row">
    <div class="form-group">
      <label for="select" class="col-lg-2">Select: </label>
      <div class="col-lg-10">
        <select id="select" class="form-control" >
          <option>Option 1</option>
          <option>Option 2</option>
          <option>Option 3</option>
        </select>
      </div>
    </div>
  </div>
  <div class="form-group">
```

```
<button class="pull-right btn btn-default">Envoyer</button>
</div>
</form>
```

Ce qui nous donne la figure suivante.

Légende

Text :

Textarea : Un formulaire

Select: ▼

horizontal

Les étiquettes disposent de 2 colonnes et les champs de saisie de 10 colonnes. On peut voir aussi l'utilisation de la classe `pull-right` pour envoyer le bouton vers la droite. Voici à la figure suivante une visualisation du découpage.

Légende

row	Text :	<input type="text"/>
row	Textarea :	<input type="text"/>
row	Select :	<input type="text" value="Option 1"/> ▼

↑ col-lg-2 ↑ col-lg-10

Décryptage du formulaire horizontal

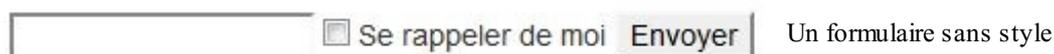
Formulaire en ligne

Il arrive qu'on ait besoin d'avoir un formulaire disposé sur une ligne, par exemple dans une barre de navigation. Faisons un essai avec un formulaire élémentaire :

Code : HTML

```
<form>
  <input type="text">
  <input type="checkbox"> Se rappeler de moi
  <button>Envoyer</button>
</form>
```

En l'état, on obtient le résultat visible à la figure suivante.



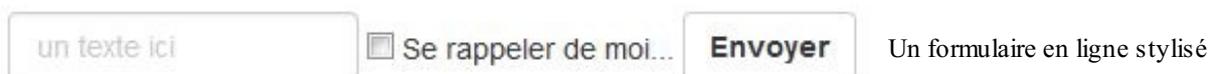
Un formulaire sans style

Bon ce n'est pas très joli, alors arrangeons ça avec tout ce que nous avons déjà vu et en précisant la classe `form-inline` :

Code : HTML

```
<form class="form-inline">
  <div class="form-group">
    <label class="sr-only" for="text">Texte</label>
    <input type="text" class="form-control" id="text"
placeholder="un texte ici">
  </div>
  <div class="form-group">
    <div class="checkbox">
      <label>
        <input type="checkbox"> Se rappeler de moi...
      </label>
    </div>
  </div>
  <button type="submit" class="btn btn-default">Envoyer</button>
</form>
```

On obtient alors la figure suivante.



Un formulaire en ligne stylisé

C'est beaucoup mieux ! Remarquez l'utilisation d'une étiquette invisible pour la zone de texte. Il est conseillé d'avoir toujours une étiquette pour chaque contrôle pour les lecteurs d'écran (accessibilité), la classe **sr-only** permet de rendre cette étiquette invisible.

Les classes `input-group`, `input-group-addon` et `input-group-btn` à l'œuvre

Accoler une information à un contrôle

Les classes `input-group` et `input-group-addon` servent à accoler une information à un contrôle de formulaire :

Code : HTML

```
<div class="input-group col-lg-3">
  <span class="input-group-addon">€</span>
  <input type="text" class="form-control" value="100">
</div>
<div class="input-group col-lg-3">
  <input type="text" class="form-control" style="text-align:right"
value="100,50">
  <span class="input-group-addon">€</span>
</div>
```

Ce qui nous donne la figure suivante.

€	100	100,50	€	Une information
---	-----	--------	---	-----------------

accolée à un contrôle

On peut aussi les utiliser conjointement pour accoler devant et derrière :

Code : HTML

```
<div class="input-group col-lg-2">
  <span class="input-group-addon">€</span>
  <input type="text" class="form-control" style="text-align:right"
value="100">
  <span class="input-group-addon">.00</span>
</div>
```

Ce qui nous donne la figure suivante.

€	100	.00	Deux informations accolées à un contrôle
---	-----	-----	--

Accoler un bouton à un contrôle

On peut aussi accoler un bouton avec la classe `input-group-btn` :

Code : HTML

```
<div class="input-group col-lg-3">
  <input type="text" class="form-control" style="text-align:right"
value="10 000">
  <span class="input-group-btn">
    <button class="btn btn-default" type="button">Valider</button>
  </span>
</div>
```

Ce qui nous donne la figure suivante.

10 000	Valider	Un bouton accolé à un contrôle
--------	---------	--------------------------------

Et rien n'empêche d'en ajouter plusieurs, comme pour la figure suivante.

Code : HTML

```
<div class="input-group col-lg-4">
  <span class="input-group-btn">
    <button class="btn btn-default" type="button">Vérifier</button>
  </span>
  <input type="text" class="form-control" style="text-align:right"
value="10 000">
  <span class="input-group-btn">
    <button class="btn btn-default" type="button">Valider</button>
  </span>
</div>
```

```
<button class="btn btn-default" type="button">Annuler</button>
</span>
</div>
```

Plusieurs boutons accolés à un contrôle

Ça fonctionne correctement, mais la documentation officielle précise que cela n'est pas pris en charge...

Accoler une case à cocher ou un bouton radio à un contrôle

Code : HTML

```
<div class="input-group col-lg-3">
  <div class="form-group">
    <div class="input-group">
      <span class="input-group-addon">
        <input type="checkbox">
      </span>
      <input type="text" class="form-control">
    </div>
  </div>
  <div class="form-group">
    <div class="input-group">
      <span class="input-group-addon">
        <input type="radio">
      </span>
      <input type="text" class="form-control">
    </div>
  </div>
</div>
```

Ce qui nous donne la figure suivante.

Bouton radio et case à cocher accolés à un contrôle

Dimension des contrôles

Pour ajuster un formulaire, il est parfois utile de jouer sur la taille des contrôles. Bootstrap propose les classes `input-lg` et `input-sm` pour le faire dans un sens ou dans l'autre :

Code : HTML

```
<div class="col-lg-3">
  <input type="text" class="form-control input-sm" value="Petit">
  <input type="text" class="form-control" value="Normal">
  <input type="text" class="form-control input-lg" value="Grand">
</div>
```

Ce qui nous donne la figure suivante.

Contrôles de dimensions différentes

Notez qu'on joue ici uniquement sur la hauteur du contrôle, la largeur étant définie par la grille.

Stylisation selon l'état

Bootstrap embarque des classes de stylisation des zones de saisie selon l'état. Voici un exemple :

Code : HTML

```
<form class="col-lg-5">
  <fieldset>
    <legend>Validation, erreur et succès</legend>
    <div class="form-group has-error">
      <label class="control-label" for="idError">Erreur</label>
      <input type="text" class="form-control" id="idError">
      <span class="help-block">Corrigez l'erreur s'il vous
plait</span>
    </div>
    <div class="form-group has-warning">
      <label class="control-label"
for="idWarning">Avertissement</label>
      <input type="text" class="form-control" id="idWarning">
      <span class="help-block">Il y a un problème dans la
saisie</span>
    </div>
    <div class="form-group has-success">
      <label class="control-label" for="idSuccess">Réussite</label>
      <input type="text" class="form-control" id="idSuccess">
      <span class="help-block">Saisie correcte</span>
    </div>
  </fieldset>
</form>
```

Ce qui nous donne la figure suivante.

Validation, erreur et succès

Erreur

Corrigez l'erreur s'il vous plait

Avertissement

Il y a un problème dans la saisie

Réussite

Saisie correcte

Stylisation selon l'état

La version 3.1 de Bootstrap a prévu la possibilité d'ajouter une icône pour améliorer l'effet visuel. Nous verrons comment Bootstrap permet d'utiliser des icônes dans le chapitre suivant. Je vous mets juste l'exemple précédent complété sans commenter l'ajout des icônes, vous pourrez y revenir plus tard quand vous aurez avancé dans ce cours :

Code : HTML

```
<form class="col-lg-5">
  <fieldset>
    <legend>Validation, erreur et succès</legend>
    <div class="form-group has-error has-feedback">
      <label class="control-label" for="idError">Erreur</label>
      <input type="text" class="form-control" id="idError">
    <span class="glyphicon glyphicon-remove form-control-feedback"></span>
      <span class="help-block">Corrigez l'erreur s'il vous
plait</span>
    </div>
    <div class="form-group has-warning has-feedback">
      <label class="control-label"
for="idWarning">Avertissement</label>
      <input type="text" class="form-control" id="idWarning">
    <span class="glyphicon glyphicon-warning-sign form-control-feedback"></span>
      <span class="help-block">Il y a un problème dans la
saisie</span>
    </div>
    <div class="form-group has-success has-feedback">
      <label class="control-label" for="idSuccess">Réussite</label>
      <input type="text" class="form-control" id="idSuccess">
    <span class="glyphicon glyphicon-ok form-control-feedback"></span>
      <span class="help-block">Saisie correcte</span>
    </div>
  </fieldset>
</form>
```

Remarquez l'ajout des classes `has-feedback` et `form-control-feedback` pour que l'icône soit bien intégrée dans le contrôle. Ce qui nous donne la figure suivante.

Validation, erreur et succès

Erreur

Corrigez l'erreur s'il vous plait

Avertissement

Il y a un problème dans la saisie

Réussite

Saisie correcte

Stylisation selon l'état avec des

icônes

Dans ces exemples il y a peu d'actions possibles car, pour le moment, je n'ai pas prévu d'interactivité. Alors un petit exemple : nous voulons faire un formulaire comme celui de la figure suivante.

Formulaire à obtenir

Et nous voulons tester la saisie pour vérifier que le texte fait au moins 4 caractères. Si ce n'est pas le cas, on change le style de la zone de texte et on fait apparaître un message d'avertissement, comme à la figure suivante.

Message à afficher

Tant qu'à faire, on veut aussi que le message d'avertissement disparaisse au bout de 4 secondes et que le tout se fasse avec fluidité. En combinant Bootstrap et jQuery c'est un jeu d'enfant :

Code : HTML

```
<div class="col-lg-4">
  <form class="form-inline well">
    <div class="form-group">
      <label class="sr-only" for="text">Saisie</label>
      <input id="text" type="text" class="form-control">
```

```
placeholder="Texte ici">
  </div>
  <button type="submit" class="btn btn-primary pull-
right">Envoyer</button>
  <div class="alert alert-block alert-danger"
style="display:none">
    <h4>Erreur !</h4>
    Vous devez entrer au moins 4 caractères !
  </div>
</form>
</div>
<script src="assets/js/jquery.js"></script>
<script>
  $(function() {
    $("form").on("submit", function() {
      if($("input").val().length < 4) {
        $("div.form-group").addClass("has-error");
        $("div.alert").show("slow").delay(4000).hide("slow");
        return false;
      }
    });
  });
</script>
```

Tester !

Je ne commente pas le code jQuery ajouté parce que ce n'est pas l'objet de ce cours, mais je conseille vraiment de connaître ce framework Javascript puissant et élégant pour tirer pleinement parti de toutes les possibilités de Bootstrap.

Un générateur de formulaire

Le site [bootsnipp](#) propose un [générateur de formulaire](#) pour Bootstrap (voir à la figure suivante).

Un générateur de formulaire pour Bootstrap

On construit le formulaire avec de simples glisser-déposer. On renseigne ensuite les paramètres avec un formulaire qui apparaît en fenêtre modale. À la fin, on n'a plus qu'à copier le code HTML. On ne peut réaliser que des formulaires horizontaux, mais une simple adaptation du code permet de réaliser facilement d'autres sortes de formulaires.

Boutons, icônes et images

La page d'exemple

Voici la nouvelle version de la page d'exemple :

Code : HTML

```
<!DOCTYPE HTML>
<html>

  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      body { background-color:#DDD; }
      [class*="col"] { margin-bottom: 20px; }
      .form-inline { margin-top: 20px}
      img { width: 100%; }
      .well {
        background-color:#CCC;
        padding: 20px;
      }
    </style>
  </head>

  <body>
```

```

<div class="container">
  <header class="row">
    <div class="col-sm-12 col-lg-7">
      <h1>Mon amour pour les tigres</h1>
    </div>
    <div class="col-sm-12 col-lg-5">
      <form class="form-inline pull-right">
        <input type="text" style="width:150px" class="input-sm
form-control" placeholder="Recherche">
        <button type="submit" class="btn btn-primary btn-
xs"><span class="glyphicon glyphicon-eye-open"></span>
Chercher</button>
      </form>
    </div>
  </header>
  <section class="row">
    <div class="col-lg-12">
      <p>
        Je suis passionné par les <strong>tigres</strong> depuis
très longtemps. Ce site a été construit en <em> hommage à ces
merveilleux félins...</em><br>
        Je fais partie de la <abbr title="Société des Amoureux
des Tigres">SAT</abbr> qui a pour but de faire connaître ces
splendides animaux.
      </p>
      <p>Voici ce qu'en dit le wikipedia :</p>
      <blockquote>
        Le Tigre (Panthera tigris) est un mammifère carnivore
de la famille des félidés (Felidae) du genre Panthera.
Aisément reconnaissable à sa fourrure rousse rayée de noir, il est
le plus grand félin sauvage et l'un des plus grands carnivores du
monde.
L'espèce est divisée en neuf sous-espèces possédant des différences
mineures en termes de taille ou de comportement.
Superprédateur, il chasse principalement les cerfs et les sangliers,
bien qu'il puisse s'attaquer à des proies de taille plus importante
comme les buffles.
Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La
structure sociale des tigres en fait un animal solitaire ;
le mâle possède un territoire qui englobe les domaines de plusieurs
femelles et ne participe pas à l'éducation des petits.<br>
        <small class="pull-right">Wikipedia</small><br>
      </blockquote>
    </div>
  </section>
  <section class="row">
    <div class="col-xs-4 col-sm-3 col-md-2"></div>
    <div class="col-xs-4 col-sm-3 col-md-2"></div>

```

```

</section>
<div class="row">
  <section class="col-sm-4">
    <h5>Voici les sous-espèces des tigres :</h5>
    <br>
    <ul>
      <li>Tigre de Sibérie</li>
      <li>Tigre de Chine méridionale</li>
      <li>Tigre de Bali</li>
      <li>Tigre de d'Indochine</li>
      <li>Tigre de Malaisie</li>
      <li>Tigre de Java</li>
      <li>Tigre de Sumatra</li>
      <li>Tigre du Bengale</li>
      <li>Tigre de la Caspienne</li>
    </ul>
  </section>
  <section class="col-sm-8">
    
  </section>
</div>
<div class="row">
  <section class="col-sm-12">
    <table class="table table-bordered table-striped table-
condensed">
      <caption>
        <h4>Les menaces pour les tigres</h4>
      </caption>
      <thead>
        <tr>
          <th>Lieu</th>
          <th>Menace</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Grand Mekong</td>
          <td>Demande croissante de certaines parties de
l'animal pour la médecine chinoise traditionnelle et fragmentation
des habitats
du fait du développement non durable d'infrastructures</td>
        </tr>
        <tr>
          <td>Île de Sumatra</td>
          <td>Production d'huile de palme et de pâtes à
papiers</td>
        </tr>
        <tr>
          <td>Indonésie et Malaisie</td>
          <td>Pâte à papier, l'huile de palme et le
caoutchouc</td>
        </tr>
        <tr>
          <td>États-Unis</td>
          <td>Les tigres captifs représentent un danger pour
les tigres sauvages</td>
        </tr>
        <tr>
          <td>Europe</td>
          <td>Gros appétit pour l'huile de palme</td>
        </tr>
        <tr>
          <td>Népal</td>
          <td>Commerce illégal de produits dérivés de
tigres</td>
        </tr>
      </tbody>
    </table>
  </section>

```

```

</div>
<div class="row">
  <section class="col-sm-8">
    <form class="well">
      <legend>Si vous voulez me laisser un message</legend>
      <h4>Comment m'avez-vous trouvé ?</h4>
      <fieldset>
        <label for="ami" class="radio">
          <input type="radio" name="origine" value="ami"
id="ami">
          Par un ami
        </label>
        <label for="web" class="radio">
          <input type="radio" name="origine" value="web"
id="web">
          Sur le web
        </label>
        <label for="hasard" class="radio">
          <input type="radio" name="origine" value="hasard"
id="hasard">
          Par hasard
        </label>
        <label for="autre" class="radio">
          <input type="radio" name="origine" value="autre"
id="autre">
          Autre...
        </label>
        <label for="textarea">Votre message :</label>
        <textarea id="textarea" rows="4" class="form-
control"></textarea>
        <p class="help-block">Vous pouvez agrandir la
fenêtre</p>
        <button class="btn btn-primary" type="submit"><span
class="glyphicon glyphicon-ok-sign"></span> Envoyer</button>
      </fieldset>
    </form>
  </section>
  <section class="col-sm-4">
    <address>
      <p>Vous pouvez me contacter à cette adresse :</p>
      <strong>Tigrou Alfred</strong><br>
      Allée des fauves<br>
      28645 Félines-sur-Loire<br>
    </address>
  </section>
</div>
</div>
</body>
</html>

```

Tester !

Qu'y a-t-il de nouveau ? Déjà un formulaire de recherche dans la partie supérieure droite, avec une zone de texte et un bouton muni d'une icône (voir figure suivante).



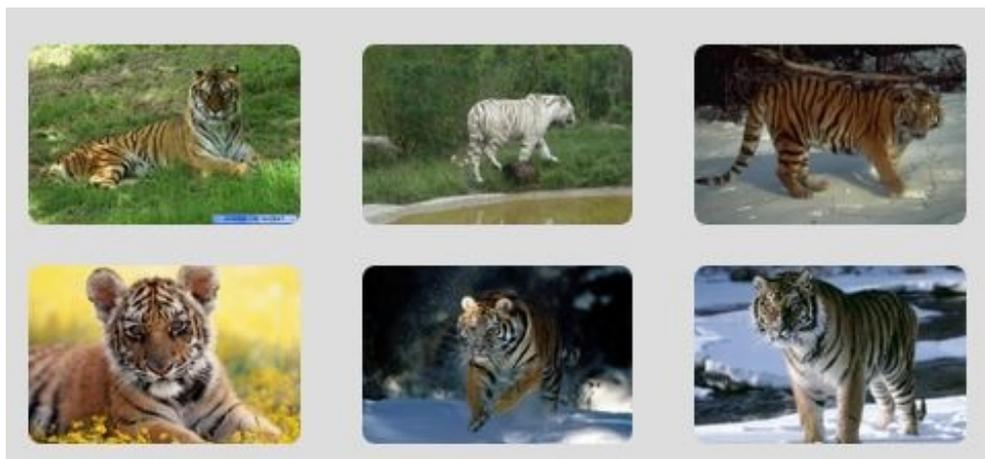
Le formulaire de recherche

Remarquez le traitement du *responsive* pour ce formulaire. Pour un écran large, il est calé en haut à droite avec une classe `col-lg-5` et un `pull-right`. Quand on passe au format moyen, il se retrouve avec la classe `col-sm-12` de telle manière qu'il vienne se disposer sous le titre, toujours calé à droite (voir figure suivante).

Mon amour pour les tigres

Effet de la classe pull-right

Les vignettes des images ont désormais leurs coins arrondis, comme le montre la figure suivante.



Des images aux coins arrondis

Le bouton du formulaire de contact a hérité d'une icône (voir figure suivante).

Votre message :

Vous pouvez agrandir la fenêtre

Une icône dans le bouton

Nous allons passer en revue ces nouveautés et détailler les possibilités de Bootstrap pour les boutons, les icônes et les images.

Des jolis boutons

Pour créer un bouton avec Bootstrap il suffit d'ajouter les classes `btn` et `btn-default` à une balise `<button>`, `<a>` ou `<input>`:

Code : HTML

```
<button type="button" class="btn btn-default">Bouton de
base</button>
<a class="btn btn-default" href="#">Bouton avec lien</a>
<input class="btn btn-default" type="button" value="Bouton avec
input">
```

Ce qui nous donne la figure suivante.



Le rendu est le même, tout dépend du contexte d'utilisation.

Les différents types de boutons

L'aspect par défaut est un gris avec le texte blanc. Il existe d'autres possibilité plus colorées :

Code : HTML

```
<button type="button" class="btn btn-primary">Bouton
standard</button>
<button type="button" class="btn btn-success">Réussite</button>
<button type="button" class="btn btn-info">Information</button>
<button type="button" class="btn btn-warning">Avertissement</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Lien</button>
```

Ce qui nous donne les boutons visibles à la figure suivante.



À utiliser selon les situations et à faire varier selon l'humeur !

Dimension des boutons

On peut aussi ajuster la dimension du bouton :

Code : HTML

```
<button type="button" class="btn btn-primary btn-lg">Bouton
gros</button>
<button type="button" class="btn btn-primary">Bouton
standard</button>
<button type="button" class="btn btn-primary btn-sm">Bouton
petit</button>
<button type="button" class="btn btn-primary btn-xs">Bouton très
petit</button>
```

Ce qui nous donne la figure suivante.



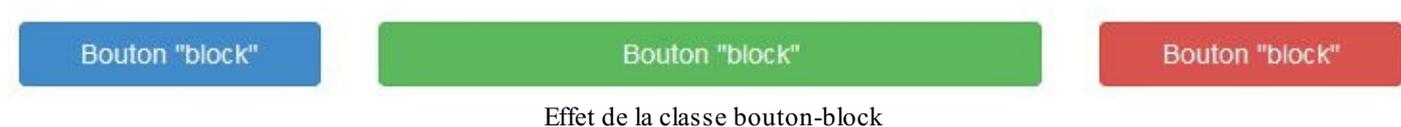
Boutons "block"

La classe `btn-block` règle la dimension du bouton sur la dimension de son contenant. Voici un exemple :

Code : HTML

```
<div class="row">
  <div class="col-sm-3">
    <button type="button" class="btn btn-primary btn-block">Bouton
"block"</button>
  </div>
  <div class="col-sm-6">
    <button type="button" class="btn btn-success btn-block">Bouton
"block"</button>
  </div>
  <div class="col-sm-3">
    <button type="button" class="btn btn-danger btn-block">Bouton
"block"</button>
  </div>
</div>
```

Ce qui nous donne la figure suivante.



Grouper des boutons

Il est possible de grouper des boutons horizontalement avec la classe `btn-group`. Ce groupement peut s'effectuer aussi verticalement avec la classe `btn-group-vertical`. Voici un exemple avec les deux groupements :

Code : HTML

```
<div class="row">
  <div class="col-lg-2">
    <div class="btn-group">
      <a class="btn btn-danger" href="#">1</a>
      <a class="btn btn-info" href="#">2</a>
      <a class="btn btn-warning" href="#">3</a>
    </div>
  </div>
  <div class="col-lg-1">
    <div class="btn-group-vertical">
      <a class="btn btn-danger" href="#">4</a>
      <a class="btn btn-info" href="#">5</a>
      <a class="btn btn-warning" href="#">6</a>
    </div>
  </div>
</div>
```

Ce qui nous donne la figure suivante.



On peut dimensionner le groupement :

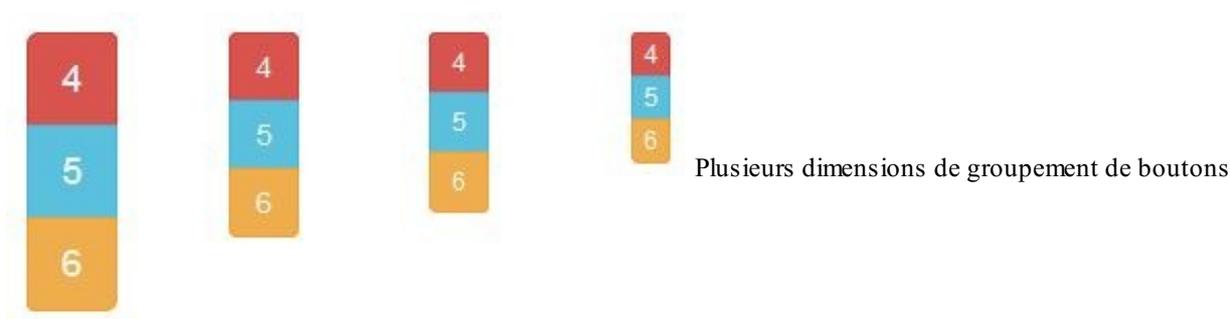
Code : HTML

```

<div class="row">
  <div class="col-lg-1">
    <div class="btn-group-vertical btn-group-lg">
      <a class="btn btn-danger" href="#">4</a>
      <a class="btn btn-info" href="#">5</a>
      <a class="btn btn-warning" href="#">6</a>
    </div>
  </div>
  <div class="col-lg-1">
    <div class="btn-group-vertical">
      <a class="btn btn-danger" href="#">4</a>
      <a class="btn btn-info" href="#">5</a>
      <a class="btn btn-warning" href="#">6</a>
    </div>
  </div>
  <div class="col-lg-1">
    <div class="btn-group-vertical btn-group-sm">
      <a class="btn btn-danger" href="#">4</a>
      <a class="btn btn-info" href="#">5</a>
      <a class="btn btn-warning" href="#">6</a>
    </div>
  </div>
  <div class="col-lg-1">
    <div class="btn-group-vertical btn-group-xs">
      <a class="btn btn-danger" href="#">4</a>
      <a class="btn btn-info" href="#">5</a>
      <a class="btn btn-warning" href="#">6</a>
    </div>
  </div>
</div>

```

Ce qui nous donne la figure suivante.



Boutons justifiés

Si vous voulez que vos boutons se répartissent harmonieusement et uniformément dans un contenant, utilisez la classe `btn-group-justified` en plus de `btn-group` :

Code : CSS

```
<div class="col-lg-6">
  <div class="btn-group btn-group-justified">
    <a class="btn btn-danger" href="#">1</a>
    <a class="btn btn-info" href="#">2</a>
    <a class="btn btn-warning" href="#">3</a>
  </div>
</div>
```

Ce qui nous donne la figure suivante.



Cet exemple utilise des balises <a>. On peut réaliser la même chose avec des balises <button> à condition d'utiliser la classe btn-group pour chaque bouton :

Code : CSS

```
<div class="col-lg-6">
  <div class="btn-group btn-group-justified">
    <div class="btn-group">
      <button type="button" class="btn btn-danger">1</button>
    </div>
    <div class="btn-group">
      <button type="button" class="btn btn-info">2</button>
    </div>
    <div class="btn-group">
      <button type="button" class="btn btn-warning">3</button>
    </div>
  </div>
</div>
```

Des générateurs de bouton

Le site [bootsnipp](#) propose un générateur de bouton simple à utiliser (voir figure suivante).

Button Builder for Bootstrap 3

love it? Share:

Output:



Button text :

Mon bouton

Color :

Default Primary Info Success Warning Danger

Size :

Mini Small Default Large

Full width button

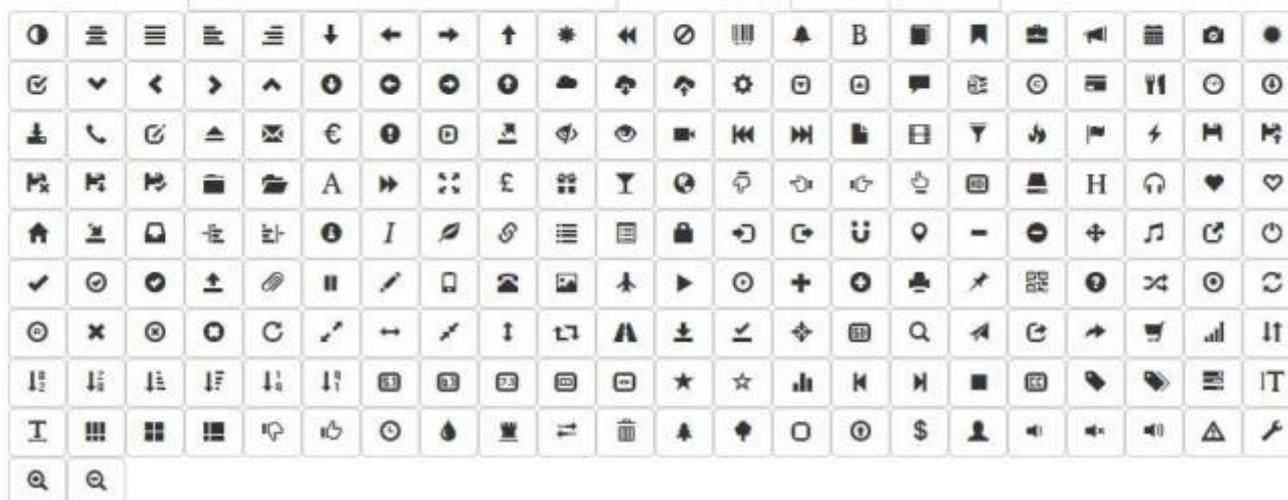
Icon:

Search icons:

Search icon

Icon position:

← Left Right →

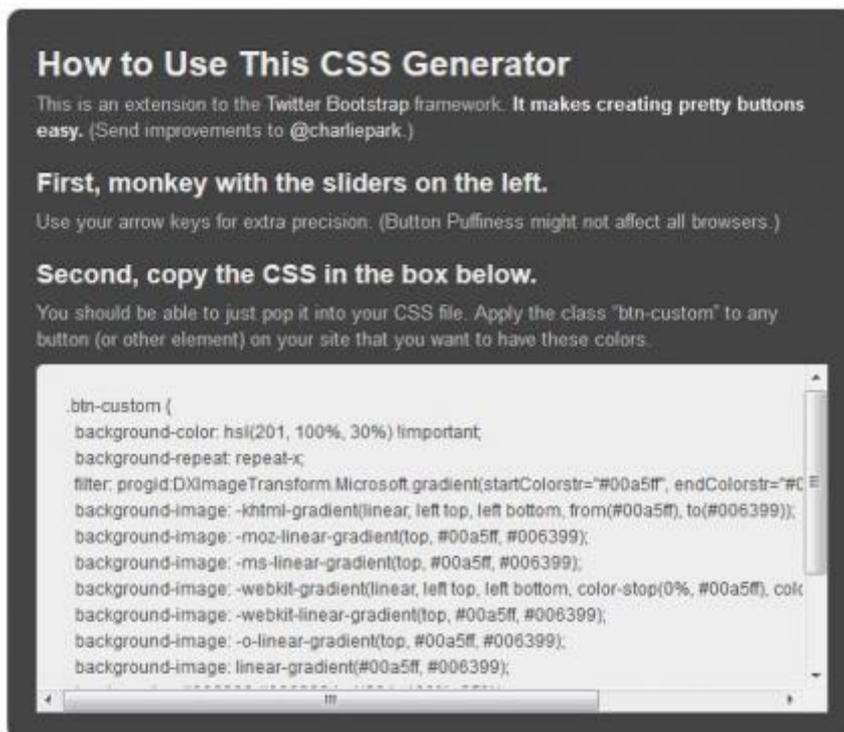


Un générateur de bouton

Vous entrez le texte du bouton, la couleur, la taille, vous choisissez l'icône et son positionnement. Vous visualisez directement le résultat et vous disposez du code HTML correspondant.

Si vous voulez des styles différents pour vos boutons et sortir des couleurs standards que Bootstrap propose, vous pouvez créer vos propres styles. Si vous n'êtes pas très inspiré, il existe un générateur simple et pratique sur le site http://charliepark.org/bootstrap_buttons/ (voir figure suivante).

Beautiful Buttons for Twitter Bootstrappers



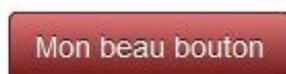
Un autre générateur de bouton

Il suffit de récupérer le code et de l'insérer dans votre feuille de style. L'intégration dans un bouton (ou autre) est simple, comme à la figure suivante.

Code : HTML

```
<button type="button" class="btn btn-custom">Mon beau bouton</button>
```

Il est ainsi facile et rapide de créer son propre style de bouton (voir figure suivante).



Un bouton généré

De jolies icônes

Les icônes de Glyphicons

Bootstrap propose 200 icônes de **Glyphicons**. L'intégration d'une icône est très facile parce qu'il suffit d'utiliser la balise ``. Pour le formulaire de contact de la page d'exemple, le bouton est amélioré en le complétant par une icône (voir figure suivante).

Code : HTML

```
<button class="btn btn-primary" type="submit"><span class="glyphicon glyphicon-ok-sign"></span> Envoyer</button>
```

Ce qui nous donne la figure suivante.



Icône dans un bouton

La syntaxe est toute simple. On utilise une balise `` avec deux classes. La principale est `glyphicon` suivi de la classe qui correspond à l'icône à afficher `glyphicon-*`. On peut ainsi créer d'élégantes barres de boutons :

Code : HTML

```
<div class="container">
  <div class="row" >
    <div class="col-lg-3">
      <div class="btn-group">
        <a class="btn btn-danger" href="#"><span class="glyphicon glyphicon-fast-backward"></span></a>
        <a class="btn btn-info" href="#"><span class="glyphicon glyphicon-backward"></span></a>
        <a class="btn btn-warning" href="#"><span class="glyphicon glyphicon-play"></span></a>
        <a class="btn btn-info" href="#"><span class="glyphicon glyphicon-forward"></span></a>
        <a class="btn btn-danger" href="#"><span class="glyphicon glyphicon-fast-forward"></span></a>
      </div>
    </div>
    <div class="col-lg-1">
      <div class="btn-group-vertical">
        <a class="btn btn-danger" href="#"><span class="glyphicon glyphicon-pencil"></span></a>
        <a class="btn btn-info" href="#"><span class="glyphicon glyphicon-search"></span></a>
        <a class="btn btn-warning" href="#"><span class="glyphicon glyphicon-print"></span></a>
        <a class="btn btn-success" href="#"><span class="glyphicon glyphicon-picture"></span></a>
      </div>
    </div>
  </div>
</div>
```

Ce qui nous donne la figure suivante.



Barres de boutons avec des icônes



Et si on veut une icône colorée ?

Comme il s'agit de polices, il suffit d'utiliser du style :

Code : HTML

```
<button class="btn btn-primary" type="submit"><span class="glyphicon glyphicon-ok-sign" style="color:#4f4;"></span> Envoyer</button>
```

Ce qui nous donne la figure suivante.



Une icône colorée

Évidemment on peut mettre ces icônes ailleurs que sur des boutons... Partout où on peut écrire en fait !

Rien n'empêche évidemment de sauter des lignes dans un bouton. On peut ainsi créer des boutons en séparant bien l'icône du texte :

Code : HTML

```
<button class="btn btn-primary btn-lg"><span class="glyphicon glyphicon-user"></span><br>Utilisateurs</button>
<button class="btn btn-warning btn-lg"><span class="glyphicon glyphicon-comment"></span><br>Commentaires</button>
<button class="btn btn-success btn-lg"><span class="glyphicon glyphicon-calendar"></span><br>Evénements</button>
<button class="btn btn-danger btn-lg"><span class="glyphicon glyphicon-shopping-cart"></span><br>Boutique</button>
<button class="btn btn-info btn-lg"><span class="glyphicon glyphicon-bullhorn"></span><br>Messages</button>
```



Séparer l'icône du texte

Des boutons ronds ?

Il est facile, en ajoutant un peu de style, de créer des boutons ronds qui conviendront à merveille aux icônes. Voici un exemple de style ajouté (voir aussi figure suivante) :

Code : CSS

```
.btn-lg {
  width: 50px;
  height: 50px;
  border-radius: 25px;
}
```

Et le code HTML :

Code : HTML

```
<button class="btn btn-primary btn-lg"><span class="glyphicon glyphicon-user"></span></button>
<button class="btn btn-warning btn-lg"><span class="glyphicon glyphicon-comment"></span></button>
<button class="btn btn-success btn-lg"><span class="glyphicon
```

```

glyphicon-calendar"></span></button>
<button class="btn btn-danger btn-lg"><span class="glyphicon
glyphicon-shopping-cart"></button>
<button class="btn btn-info btn-lg"><span class="glyphicon
glyphicon-bullhorn"></span></button>

```



Des boutons ronds

Les icônes de Font Awesome

Le site [Font Awesome](#) propose une collection de 369 icônes toutes prêtes pour Bootstrap. Évidemment elles ne sont pas intégrées de base dans Bootstrap et il faut déclarer le fichier CSS pour pouvoir les utiliser. Le site propose [plusieurs possibilités](#) pour le faire, la plus simple étant d'utiliser le CDN :

Code : HTML

```

<link href="http://netdna.bootstrapcdn.com/font-
awesome/4.0.3/css/font-awesome.css" rel="stylesheet">

```

Une fois le fichier CSS déclaré, vous pouvez utiliser les icônes. La syntaxe est celle qui était utilisée pour Bootstrap 2 avec la balise `<i>`. Le site propose [une galerie d'exemples](#). On peut insérer une icône directement dans un texte (voir figure suivante) :

Code : HTML

```

Il faut déverrouiller <i class="fa fa-unlock"></i> pour sortir <i
class="fa fa-arrow-right"></i>

```

Il faut déverrouiller  pour sortir  Icônes dans un texte

On peut aussi dimensionner les icônes selon le contexte, comme à la figure suivante :

Code : HTML

```

<p><i class="fa fa-print fa-lg"></i> Taille normale</p>
<p><i class="fa fa-print fa-2x"></i> Taille double</p>
<p><i class="fa fa-print fa-3x"></i> Taille triple</p>
<p><i class="fa fa-print fa-4x"></i> Taille quadruple</p>
<p><i class="fa fa-print fa-5x"></i> Taille quintuple</p>

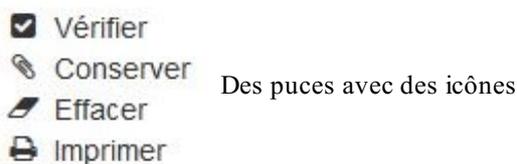
```



On peut aussi très facilement créer une liste dont les éléments sont bien identifiés avec une icône (voir figure suivante) :

Code : HTML

```
<ul class="fa-ul">
  <li><i class="fa-li fa fa-check-square"></i>Vérifier</li>
  <li><i class="fa-li fa fa-paperclip"></i>Conserver</li>
  <li><i class="fa-li fa fa-eraser"></i>Effacer</li>
  <li><i class="fa-li fa fa-print"></i>Imprimer</li>
</ul>
```



Il existe bien d'autres possibilités comme la rotation des icônes, leur animation, leur empilement, que je vous laisse découvrir en consultant [la page des exemples](#) du site.

Mise en forme des images

Bootstrap donne la possibilité de modifier l'apparence des bordures des images. Il y a pour cela 3 classes utilisables :

Classe	Effet
img-rounded	Image à coins arrondis
img-circle	Image circulaire
img-thumbnail	Image avec liseré blanc

Voici un exemple illustratif de ces trois effets :

Code : HTML

```
<div class="container">
  <div class="col-lg-2">
    
  </div>
```

```
<div class="col-lg-2">
  
</div>
<div class="col-lg-2">
  
</div>
</div>
```

Ce qui nous donne la figure suivante.



Mise en forme des

images

Notez que vous pouvez rendre les images adaptatives facilement avec la classe `img-responsive`. Cette classe possède 3 règles :

Code : CSS

```
.img-responsive {
  display: block;
  max-width: 100%;
  height: auto;
}
```

Le but est de faire en sorte que l'image occupe toute la place disponible de son contenant.

En résumé :

- Bootstrap permet une mise en forme élégante des listes, descriptions et tableaux.
- Bootstrap permet de réaliser des formulaires classiques, horizontaux ou en ligne. Il permet aussi d'enrichir les contrôles en jouant sur leur dimension ou leur style.
- Bootstrap permet de créer des boutons colorés, seuls ou groupés, de diverses dimensions.
- Bootstrap comprend une collection d'icônes vectorielles faciles à intégrer comme des caractères.
- Bootstrap prévoit la mise en forme des images avec des coins arrondis ou avec un effet de diapositive.

Les composants intégrés

Bootstrap propose des composants bien pratiques à mettre en œuvre : barres de navigation, effets typographiques, panneaux, thumbnails... et tout ça sans écrire une seule ligne de code Javascript. Nous verrons aussi alertes et barres de progression qui demandent un peu de Javascript pour être utilisées...

Allez c'est parti pour une visite guidée !

La navigation

La page d'exemple

Voici le code ajouté à la page d'exemple pour l'équiper d'une barre de navigation :

Code : HTML

```
<!DOCTYPE HTML>
<html>

  <head>

    ...   en-tête   ...

  </head>

  <body>
    <div class="container">

      <nav class="navbar navbar-inverse">
        <ul class="nav navbar-nav">
          <li class="active"> <a href="#">Accueil</a> </li>
          <li> <a href="#">Liens</a> </li>
          <li> <a href="#">Témoignages</a> </li>
          <li> <a href="#">Références</a> </li>
        </ul>
        <form class="navbar-form pull-right">
          <input type="text" style="width:150px" class="input-small"
placeholder="Recherche">
          <button type="submit" class="btn btn-primary btn-xs"><span
class="glyphicon glyphicon-eye-open"></span> Chercher</button>
        </form>
      </nav>

      ...   contenu   ...

    </body>
  </html>
```

Ce qui nous donne la figure suivante.

[Accueil](#) [Liens](#) [Témoignages](#) [Références](#)

Mon amour pour les tigres

Je suis passionné par les **tigres** depuis très longtemps. Ce site a été construit en *hommage à ces merveilleux félins...*

Je fais partie de la SAT qui a pour but de faire connaître ces splendides animaux.

Voici ce qu'en dit le wikipedia :

Le Tigre (*Panthera tigris*) est un mammifère carnivore de la famille des félidés (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands carnivores du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement. Superprédateur, il chasse principalement les cerfs et les sangliers, bien qu'il puisse s'attaquer à des proies de taille plus importante comme les buffles. Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle possède un territoire qui englobe les domaines de plusieurs femelles et ne participe pas à l'éducation des petits.

— Wikipedia



La page d'exemple

[Tester](#)

Une barre de navigation

Principe de base

Bootstrap propose de superbes classes pour créer une barre de navigation. À la base, une barre nécessite juste les classes `navbar` et `navbar-default`:

Code : HTML

```
<nav class="navbar navbar-default">
  Je suis une barre
</nav>
```

Ce code donne la figure suivante.

Je suis une barre

Une simple barre

La classe `navbar` est toute simple :

Code : CSS

```
.navbar {
  position: relative;
  min-height: 50px;
  margin-bottom: 20px;
  border: 1px solid transparent;
}
```

Une hauteur fixée à 50 pixels, une petite marge basse pour bien distinguer la barre du reste de la page, une bordure fine et une position relative. La classe `navbar-default` fixe les couleurs du fond et de la bordure :

Code : CSS

```
.navbar-default {
  background-color: #f8f8f8;
  border-color: #e7e7e7;
}
```

Pourquoi 2 classes ? Nous verrons plus loin qu'il est possible d'avoir une barre « inversée », avec une couleur de fond différente.

Mais une barre vide ne sert pas à grand chose, en général on la garnit d'items. Voici un exemple :

Code : HTML

```
<div class="navbar navbar-default">
  <ul class="nav navbar-nav">
    <li class="active"> <a href="#">Accueil</a> </li>
    <li> <a href="#">Liens</a> </li>
    <li> <a href="#">Témoignages</a> </li>
    <li class="disabled"> <a href="#">Références</a> </li>
  </ul>
</div>
```

Les classes `nav` et `navbar-nav` doivent être utilisées avec une balise ``. Tous les items de navigation apparaissent alors avec des balises ``. L'item actif est repéré avec la classe `active`. Un item désactivé est défini avec la classe `disabled`. Voici à la figure suivante un aperçu de la barre.

Accueil

Liens

Témoignages

Références

Effet des classes `navbar` et `navbar-default`

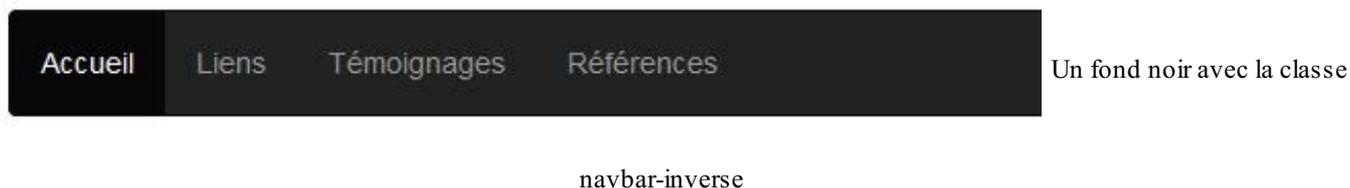
Barre avec un fond noir

Si vous préférez une barre avec un fond noir, il suffit de remplacer la classe `navbar-default` par la classe `navbar-inverse` :

Code : HTML

```
<div class="navbar navbar-inverse">
  <ul class="nav navbar-nav">
    <li class="active"> <a href="#">Accueil</a> </li>
    <li> <a href="#">Liens</a> </li>
    <li> <a href="#">Témoignages</a> </li>
    <li> <a href="#">Références</a> </li>
  </ul>
</div>
```

On obtient alors le même résultat que la figure suivante.



Un titre dans la barre

On peut faire apparaître un titre avec des caractères plus visibles avec la classe `navbar-brand` :

Code : HTML

```
<div class="navbar navbar-default">
  <div class="navbar-header">
    <a class="navbar-brand" href="#">Mon beau site</a>
  </div>
  <ul class="nav navbar-nav">
    <li class="active"><a href="#">Accueil</a></li>
    <li><a href="#">Liens</a></li>
    <li><a href="#">Témoignages</a></li>
    <li><a href="#">Références</a></li>
    <li><a href="#">Bibliographie</a></li>
  </ul>
</div>
```

Ce qui nous donne la figure suivante.



Le titre doit être inclus dans une classe `navbar-header` pour avoir un affichage correct sur les mobiles.

Fixer la barre ?

A priori, on veut une barre de navigation dans la partie supérieure d'un site, mais en général sous un bandeau. Avec le code tel qu'on l'a vu jusque-là, la barre se situe dans le flux. Par exemple...

Code : HTML

```

<p>Un paragraphe</p>
<div class="navbar navbar-default">
  <div class="navbar-header">
    <a class="navbar-brand" href="#">Mon beau site</a>
  </div>
  <ul class="nav navbar-nav">
    <li class="active"><a href="#">Accueil</a></li>
    <li><a href="#">Liens</a></li>
    <li><a href="#">Témoignages</a></li>
    <li><a href="#">Références</a></li>
    <li><a href="#">Bibliographie</a></li>
  </ul>
</div>
<p>Un autre paragraphe</p>

```

... donnera la figure suivante.

Un paragraphe



La

Un autre paragraphe

barre est dans le flux

On peut fixer la barre en haut de la page du navigateur avec la classe `navbar-fixed-top` :

Code : HTML

```

<p>Un paragraphe</p>
<div class="navbar navbar-default navbar-fixed-top">
  <div class="navbar-header">
    <a class="navbar-brand" href="#">Mon beau site</a>
  </div>
  <ul class="nav navbar-nav">
    <li class="active"><a href="#">Accueil</a></li>
    <li><a href="#">Liens</a></li>
    <li><a href="#">Témoignages</a></li>
    <li><a href="#">Références</a></li>
    <li><a href="#">Bibliographie</a></li>
  </ul>
</div>
<p>Un autre paragraphe</p>
<p>Encore un autre paragraphe</p>

```

Ce qui nous donne la figure suivante.



Barre fixée

Encore un autre paragraphe

en haut de la page

La barre prend alors toute la largeur de la fenêtre. Les éléments de la page se déplacent sous la barre fixe en cas de scrolling.



Mais où sont passés les 2 premiers paragraphes ?

Il sont sous la barre ! Il suffit de regarder la classe pour comprendre :

Code : CSS

```
.navbar-fixed-top, .navbar-fixed-bottom {  
  position: fixed;  
  right: 0;  
  left: 0;  
  z-index: 1030;  
}
```

La barre est située en positionnement fixe en haut de la page du navigateur avec un z-index important... Donc si on adopte un positionnement fixe en haut de la page du navigateur, il faut prendre la précaution de ménager suffisamment d'espace pour que les éléments ne soient pas masqués par la barre :

Code : CSS

```
<link href="assets/css/bootstrap.css" rel="stylesheet">  
<style type="text/css">  
  body { padding-top: 70px; }  
</style>
```

Ce qui nous donne la figure suivante.



Un paragraphe

Barre avec

Un autre paragraphe

Encore un autre paragraphe

espace réservé

Maintenant on retrouve nos paragraphes bien visibles !

Il existe aussi la classe `navbar-fixed-bottom` pour fixer la barre en bas de la page, mais les cas d'utilisation sont plus anecdotiques. Le principe demeure le même.

Une barre statique

Une barre fixe située en haut reste collée au bord de la fenêtre, et quand vous faites défiler avec l'ascenseur, le contenu passe sous la barre de façon logique. Il se peut que vous désiriez que la barre suive le mouvement et ne soit plus collée au bord de la fenêtre, mais au-dessus du contenu. Dans ce cas, évidemment, la barre disparaît quand vous faites défiler. Il faut alors utiliser la classe `navbar-static-top` :

Code : HTML

```
<div class="navbar navbar-default navbar-static-top">
```

Un formulaire de recherche dans la barre

Pour intégrer un formulaire il suffit d'ajouter une balise `<form>` avec la classe `navbar-form`. On a 2 classes pour envoyer ce formulaire sur l'un des côtés :

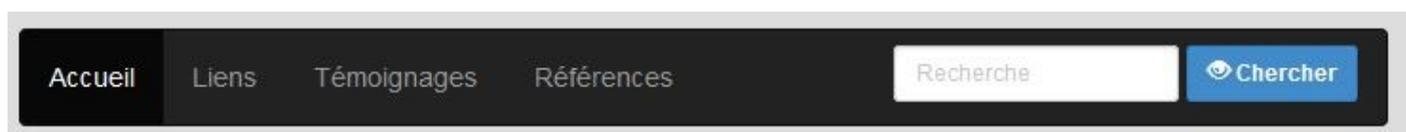
- `pull-right` pour le fixer à droite
- `pull-left` pour le fixer à gauche

Voici l'intégration pour notre page d'exemple :

Code : HTML

```
<nav class="navbar navbar-inverse">
  <ul class="nav navbar-nav">
    <li class="active"> <a href="#">Accueil</a> </li>
    <li> <a href="#">Liens</a> </li>
    <li> <a href="#">Témoignages</a> </li>
    <li> <a href="#">Références</a> </li>
  </ul>
  <form class="navbar-form pull-right">
    <input type="text" style="width:150px" class="input-sm form-control"
    placeholder="Recherche">
    <button type="submit" class="btn btn-primary btn-sm"><span
    class="glyphicon glyphicon-eye-open"></span> Chercher</button>
  </form>
</nav>
```

Ce qui nous donne la figure suivante.



Un formulaire dans la barre

Notez que j'ai fixé la largeur de la zone de saisie de texte à 150 px. Ce n'est pas indispensable, mais j'avais envie d'augmenter un peu la valeur par défaut.

Bouton et texte dans une barre

Il existe la classe `navbar-btn` pour intégrer un bouton dans une barre, mais qui ne doit être utilisée uniquement si le bouton n'appartient pas à un formulaire. Il y a aussi la classe `navbar-text` pour insérer du texte. Voici un exemple :

Code : HTML

```
<nav class="navbar navbar-default">
  <div class="navbar-header">
    <a class="navbar-brand" href="#">Accueil</a>
  </div>
  <p class="navbar-text">Connexion au site :</p>
  <button type="button" class="btn btn-default navbar-
  btn">Login</button>
</nav>
```

Ce qui nous donne la figure suivante.

Accueil

Connexion au site :

Login

Texte et bouton dans la barre

La pagination

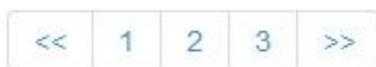
Un exemple simple

Bootstrap propose la classe `pagination` pour réaliser... des paginations. Le principe est tout simple, voici un exemple de base :

Code : HTML

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

On obtient alors le résultat visible à la figure suivante.



Pagination simple

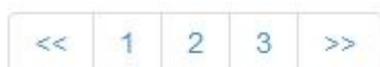
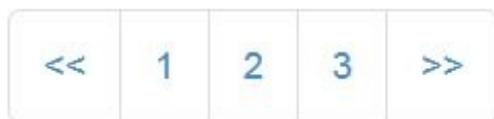
La dimension

Il y a 3 possibilités de dimension pour la pagination :

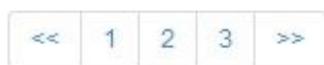
Code : HTML

```
<div class="col-lg-3">
  <ul class="pagination pagination-large">
    <li><a href="#"><<</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">>></a></li>
  </ul>
  <ul class="pagination">
    <li><a href="#"><<</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">>></a></li>
  </ul>
  <ul class="pagination pagination-small">
    <li><a href="#"><<</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">>></a></li>
  </ul>
</div>
```

Ce qui nous donne le rendu visible à la figure suivante.



Plusieurs dimensions pour la pagination



Un cas pratique

On va utiliser cette possibilité de pagination sur un exemple pratique à partir des photos de tigres que j'utilise dans ma page de démonstration. Le but est de faire défiler les images 3 par 3. Pour définir la page actuelle on utilise la classe active.

Code : HTML

```
<!DOCTYPE HTML>
<html>
  <head>
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      body { background-color:#DDD; }
      img { width: 100%; }
      a:focus { outline:0; }
    </style>
  </head>

  <body>

    <br>
    <div class="container">
      <div id="content" class="row">
        <div class="col-lg-2"></div>
        <div class="col-lg-2"></div>
</div>
<div id="page_navigation"> </div>
</div>

<script src="assets/js/jquery.js"></script>
<script>

    var show_per_page = 3;
    var current_page = 0;

    function set_display(first, last) {
        $('#content').children().css('display', 'none');
        $('#content').children().slice(first, last).css('display',
'block');
    }

    function previous(){
        if($('.active').prev('.page_link').length)
go_to_page(current_page - 1);
    }

    function next(){
        if($('.active').next('.page_link').length)
go_to_page(current_page + 1);
    }

    function go_to_page(page_num){
        current_page = page_num;
        start_from = current_page * show_per_page;
        end_on = start_from + show_per_page;
        set_display(start_from, end_on);
        $('.active').removeClass('active');
        $('#id' + page_num).addClass('active');
    }

    $(document).ready(function() {

        var number_of_pages =
Math.ceil($('#content').children().size() / show_per_page);

        var nav = '<ul class="pagination"><li><a
href="javascript:previous();"><<</a>';

        var i = -1;
        while(number_of_pages > ++i){
            nav += '<li class="page_link'
            if(!i) nav += ' active';
            nav += '" id="id' + i +'>';
            nav += '<a href="javascript:go_to_page(' + i + ')">'+ (i + 1)
+'</a>';
        }
        nav += '<li><a href="javascript:next();">>></a></ul>';

        $('#page_navigation').html(nav);
        set_display(0, show_per_page);

    });

</script>
</body>
</html>

```

Ce qui nous donne la figure suivante.



action

Tester

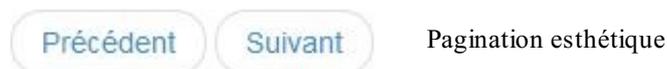
Pagination esthétique

Une autre possibilité de pagination simple consiste à utiliser la classe pager :

Code : HTML

```
<ul class="pager">
  <li><a href="#">Précédent</a></li>
  <li><a href="#">Suivant</a></li>
</ul>
```

Ce qui nous donne la figure suivante.



Par défaut, les boutons sont centrés. Pour les disposer de part et d'autre, il faut utiliser les classes previous et next:

Code : HTML

```
<ul class="pager col-lg-4">
  <li class="previous"><a href="#"><span class="glyphicon glyphicon-backward"></span> Précédent</a></li>
  <li class="next"><a href="#">Suivant <span class="glyphicon glyphicon-forward"></span></a></li>
</ul>
```

On obtient alors le résultat visible à la figure suivante.



Aide à la navigation

L'aide à la navigation suit le même principe que la pagination, avec la classe breadcrumb dans la balise et utilisation de la classe active pour définir l'élément actif :

Code : HTML

```
<ul class="breadcrumb">
  <li><a href="#">Accueil</a></li>
  <li><a href="#">Témoignages</a></li>
  <li class="active">Chasseurs</li>
</ul>
```

Ce qui nous donne la figure suivante.

Accueil / **Témoignages** / Chasseurs

Effet de la classe breadcrumb

Effets typographiques et thumbnails

La page d'exemple

Pour la nouvelle version de la page d'exemple, on va avoir une amélioration de la présentation des photos avec le composant thumbnails et des badges dans la liste des tigres, comme à la figure suivante.



Voici les sous-espèces des tigres :

- Tigre de Sibérie 230
- Tigre de Chine méridionale 120
- Tigre de Bali 540
- Tigre de d'Indochine 230
- Tigre de Malaisie 240
- Tigre de Java 1200
- Tigre de Sumatra 710
- Tigre du Bengale 20
- Tigre de la Caspienne 40



On va aussi prévoir un libellé dans le formulaire et améliorer la présentation de l'adresse (voir figure suivante).



Si vous voulez me laisser un message

Comment m'avez-vous trouvé ?

Par un ami
 Sur le web
 Par hasard
 Autre...

Votre message :

Vous pouvez agrandir la fenêtre

Vous pouvez me contacter à cette adresse :

Tigrou Alfred
Allée des fauves
28845 Félines-sur-Loire

Tester !

Jumbotron

Jumbotron contenu

La classe `jumbotron` permet de rendre le texte très visible. Prenons un exemple :

Code : HTML

```
<h1>Coucou</h1>
<p>et bonjour</p>
<div class="jumbotron">
  <h1>Coucou</h1>
  <p>et bonjour</p>
</div>
```

Ce qui donne la figure suivante.

Coucou

et bonjour

Coucou

et bonjour

La classe jumbotron

On voit bien la différence entre les éléments « normaux » et ceux qui bénéficient de la classe `jumbotron`.

Voyons cette classe :

Code : CSS

```
.jumbotron {  
  padding: 30px;  
  margin-bottom: 30px;  
  color: inherit;  
  background-color: #eee;  
}
```

Un peu de gris en fond, un réglage de hauteur de ligne et quelques marges.

Jumbotron pleine largeur

Vous avez certainement rencontré des sites créés avec Bootstrap comportant des panneaux occupant toute la largeur de la page. On réalise cela facilement avec la classe `jumbotron`, mais il faut organiser différemment le code. Il ne faut pas le mettre dans un conteneur, en revanche il faut en prévoir un à l'intérieur. Voici un exemple :

Code : HTML

```
<div class="jumbotron">  
  <div class="container">  
    <h1>Bonjour je suis le Jumbotron</h1>  
    <p>J'aime bien prendre mes aises et occuper tout l'espace pour  
    que l'on me remarque facilement !</p>  
    <p><a class="btn btn-info btn-lg" role="button">En savoir plus  
<span class="glyphicon glyphicon-glyphicon-hand-  
right"></span></a></p>  
  </div>  
</div>
```

Ce qui donne la figure suivante.



Un jumbotron

en plein largeur

Des libellés colorés

Pour créer des repères visuels textuels attractifs, il est possible de créer de beaux libellés :

Code : HTML

```
<p><span class="label label-default">Valeur d'inhibition</span>  
600</p>
```

```
<p><span class="label label-info">Valeur de signalement</span>
4000</p>
<p><span class="label label-success">Valeur de fonctionnement</span>
8000</p>
<p><span class="label label-warning">Valeur dangereuse</span>
11000</p>
<p><span class="label label-danger">Valeur critique</span> 12000</p>
```

Ce qui donne la figure suivante.



Ces libellés s'adaptent en dimension :

Code : HTML

```
<h1><span class="label label-success">H1</span></h1>
<h2><span class="label label-success">H2</span></h2>
<h3><span class="label label-success">H3</span></h3>
<h4><span class="label label-success">H4</span></h4>
<h5><span class="label label-success">H5</span></h5>
<h6><span class="label label-success">H6</span></h6>
```

On obtient ainsi un rendu identique à celui de la figure suivante.



J'ai utilisé un libellé dans le formulaire de contact de la page d'exemple :

Code : HTML

```
<h4><span class="label">Si vous voulez me laisser un
```

```
message</span></h4>
<h4>Comment m'avez-vous trouvé ?</h4>
```

On obtient ainsi la figure suivante.



Un libellé dans un formulaire

Des badges

Les badges sont un peu comme les libellés, mais pour des informations plus courtes, idéalement des nombres :

Code : HTML

```
<a href="#">Messages reçus <span class="badge">42</span></a>
```

Ce qui donne la figure suivante.

Messages reçus 42 Un badge simple

Comme le badge se trouve dans une balise ``, il est facile de l'intégrer par exemple à un bouton et de le colorer :

Code : HTML

```
<button class="btn btn-danger">Envoi <span class="badge text-success">42</span></button>
```

Ce qui donne la figure suivante.

Envoi 42 Stylisation du badge

J'ai ajouté des badges sur la page d'exemple pour représenter le nombre de tigres :

Code : HTML

```
<ul>
  <li>Tigre de Sibérie <span class="badge">230</span></li>
  <li>Tigre de Chine méridionale <span class="badge">120</span></li>
```

```

<li>Tigre de Bali <span class="badge">540</span></li>
<li>Tigre de d'Indochine <span class="badge">230</span></li>
<li>Tigre de Malaisie <span class="badge">240</span></li>
<li>Tigre de Java <span class="badge">1200</span></li>
<li>Tigre de Sumatra <span class="badge">710</span></li>
<li>Tigre du Bengale <span class="badge">20</span></li>
<li>Tigre de la Caspienne <span class="badge">40</span></li>
</ul>

```

Ce qui donne la figure suivante.



Badges dans une liste

Thumbnails

Si vous avez des images, des vidéos ou tous autres éléments à afficher dans une grille, Bootstrap propose le composant Thumbnails très facile à mettre en œuvre. Je reviens sur ma page d'exemple qui comporte de petites images de tigres. Jusque-là, j'ai positionné les images en utilisant tout simplement la grille :

Code : HTML

```

<section class="row">
  <div class="col-xs-4 col-sm-3 col-md-2"></div>
  <div class="col-xs-4 col-sm-3 col-md-2"></div>
</section>

```

Ce qui donne une organisation bien régulière, comme le montre la figure suivante.



Disposition des images avec la grille

Voyons ce que ça donne avec le composant `Thumbnails`. Le principe est simple, il faut utiliser la classe `thumbnail`, par exemple dans une balise `<a>`. Voilà ce que ça donne pour la page d'exemple en supposant que les vignettes servent de liens :

Code : HTML

```
<section class="row">
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
  <div class="col-xs-4 col-sm-3 col-md-2 ">
    <a href="#" class="thumbnail">
      
    </a>
  </div>
</section>
```

```

</div>
<div class="col-xs-4 col-sm-3 col-md-2 ">
  <a href="#" class="thumbnail">
    
  </a>
</div>
<div class="col-xs-4 col-sm-3 col-md-2 ">
  <a href="#" class="thumbnail">
    
  </a>
</div>
<div class="col-xs-4 col-sm-3 col-md-2 ">
  <a href="#" class="thumbnail">
    
  </a>
</div>
</section>

```

On obtient ainsi le même résultat qu'à la figure suivante.



Effet de la classe thumbnail

Remarquez à la figure suivante la stylisation quand on passe le curseur de la souris.



Effet du passage du curseur de la souris

Pour des raisons d'homogénéité de la présentation, je prévois aussi de traiter la photo plus grande située au-dessous :

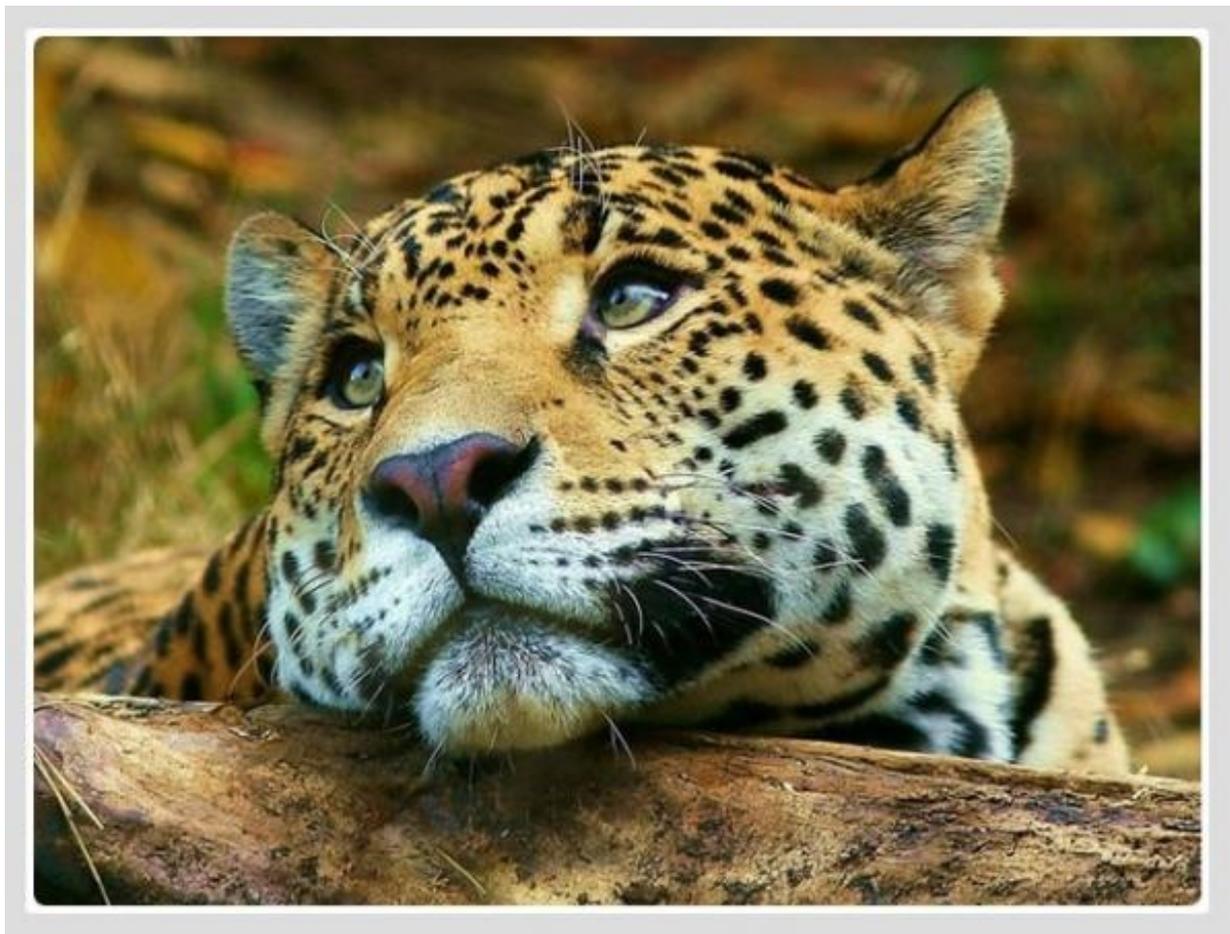
Code : HTML

```

<div class="thumbnail">
  
</div>

```

Ce qui donne la figure suivante.



Classe

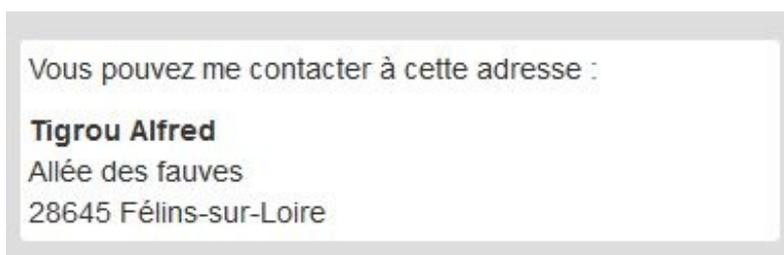
thumbnail appliquée à une grande image

Nous ne sommes évidemment pas limités à afficher des images dans un thumbnail. Puisque vous pouvez mettre cette classe dans un `<div>` ou autre, vous pouvez garnir le thumbnail de ce que vous voulez. Par exemple j'ai stylisé l'adresse sur la page d'exemple :

Code : HTML

```
<address class="thumbnail">
  <p>Vous pouvez me contacter à cette adresse :</p>
  <strong>Tigrou Alfred</strong><br>
  Allée des fauves<br>
  28645 Félin-sur-Loire<br>
</address>
```

Ce qui donne la figure suivante.



Classe thumbnail appliquée à une adresse

Listes groupées, panneaux et médias

La page d'exemple

Nous allons encore améliorer l'aspect de la page d'exemple avec une présentation de vidéos, comme à la figure suivante.



On va aussi prévoir un panneau pour la liste des tigres et également pour le tableau (voir figure suivante).

Les sous-espèces des tigres :

Tigre de Sibérie	120
Tigre de Chine méridionale	540
Tigre de Bali	250
Tigre de d'Indochine	240
Tigre de Malaisie	1200
Tigre de Java	710
Tigre de Sumatra	20
Tigre du Bengale	40
Tigre de la Caspienne	200



Les menaces pour les tigres

Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc
États-Unis	Les tigres captifs représentent un danger pour les tigres sauvages
Europe	Gros appétit pour l'huile de palme
Népal	Commerce illégal de produits dérivés de tigres

Tester !

Les listes groupées

Les listes groupées proposent une mise en forme des listes non ordonnées. Voici un premier exemple avec la visualisation d'une liste sans et avec formatage pour comparer :

Code : HTML

```
<div class="container">
  <div class="col-lg-2">
    <ul>
      <li>Item 1</li>
      <li>Item 2</li>
      <li>Item 3</li>
      <li>Item 4</li>
    </ul>
  </div>
  <div class="col-lg-2">
    <ul class="list-group">
      <li class="list-group-item">Item 1</li>
      <li class="list-group-item">Item 2</li>
      <li class="list-group-item">Item 3</li>
      <li class="list-group-item">Item 4</li>
    </ul>
  </div>
</div>
```

Ce qui donne la figure suivante.

- Item 1
- Item 2
- Item 3
- Item 4

Item 1
Item 2
Item 3
Item 4

Liste simple et liste groupée

Liste avec badges

Il est possible d'ajouter des badges tout simplement en intégrant un ``, par défaut affiché à droite :

Code : HTML

```
<ul class="list-group">
  <li class="list-group-item">
    <span class="badge">22</span>
    Item 1
  </li>
  <li class="list-group-item">
    <span class="badge">12</span>
    Item 2
  </li>
  <li class="list-group-item">
    <span class="badge">33</span>
    Item 3
  </li>
  <li class="list-group-item">
    <span class="badge">87</span>
    Item 4
  </li>
</ul>
```

Ce qui donne la figure suivante.

Item 1	22
Item 2	12
Item 3	33
Item 4	87

Liste avec des badges

Listes avec liens

Les éléments de la liste peuvent aussi être des liens. Il suffit de prévoir des balises `<a>`, l'item actif est repéré avec la classe `active` :

Code : HTML

```
<ul class="list-group">
  <a href="#" class="list-group-item active">
    <span class="glyphicon glyphicon-chevron-right pull-right"></span>
    Item 1
  </a>
  <a href="#" class="list-group-item">
    <span class="glyphicon glyphicon-chevron-right pull-right"></span>
    Item 2
  </a>
  <a href="#" class="list-group-item">
    <span class="glyphicon glyphicon-chevron-right pull-right"></span>
    Item 3
  </a>
  <a href="#" class="list-group-item">
    <span class="glyphicon glyphicon-chevron-right pull-right"></span>
    Item 4
  </a>
</ul>
```

Ce qui donne la figure suivante.



Liste avec des liens

J'ai prévu des chevrons pour appuyer la sémantique des liens, ils sont poussés sur la droite avec la classe `pull-right`.

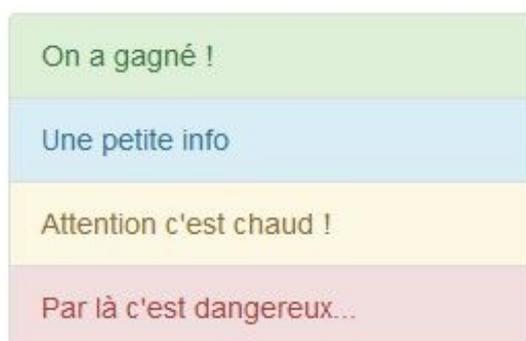
Un peu de couleur

De nouvelles classes sont apparues avec la version 3.1 de Bootstrap qui permettent d'avoir un peu de couleur dans les listes groupées. Vous trouverez un exemple illustratif à la figure suivante.

Code : HTML

```
<div class="col-lg-3">
  <ul class="list-group">
    <li class="list-group-item list-group-item-success">On a gagné
  !</li>
    <li class="list-group-item list-group-item-info">Une petite
  info</li>
    <li class="list-group-item list-group-item-warning">Attention
  c'est chaud !</li>
    <li class="list-group-item list-group-item-danger">Par là c'est
  dangereux...</li>
  </ul>
</div>
```

Et voici le résultat visuel :



Un peu de couleur dans les listes

Contenus divers

La liste est suffisamment versatile pour contenir des éléments variés. On peut avoir une en-tête avec la classe `list-group-item-heading`, et du texte avec la classe `list-group-item-text`, mais on peut intégrer par exemple une image comme je le fais ici :

Code : HTML

```
<ul class="list-group">
  <a href="#" class="list-group-item active">
    <h4 class="list-group-item-heading">Tigre 1</h4>
```

```

<p class="list-group-item-text pull-right">Mon joli tigre 1</p>

</a>
<a href="#" class="list-group-item">
  <h4 class="list-group-item-heading">Tigre 2</h4>
  <p class="list-group-item-text pull-right">Mon joli tigre 2</p>
  
</a>
</ul>

```

Ce qui donne la figure suivante.



Liste avec des images

Utilisation de liste sur la page d'exemple

J'ai utilisé une liste groupée sur la page d'exemple pour les sous-espèces de tigres (avec badges, le tout intégré dans un panneau dont nous allons bientôt parler) :

Code : HTML

```

<div class="list-group">
  <a href="#" class="list-group-item">
    Tigre de Sibérie
    <span class="badge">120</span>
  </a>
  <a href="#" class="list-group-item">
    Tigre de Chine méridionale
    <span class="badge">540</span>
  </a>
  <a href="#" class="list-group-item">
    Tigre de Bali
    <span class="badge">230</span>
  </a>
  <a href="#" class="list-group-item">
    Tigre de d'Indochine
    <span class="badge">240</span>
  </a>
  <a href="#" class="list-group-item">
    Tigre de Malaisie
    <span class="badge">1200</span>
  </a>
  <a href="#" class="list-group-item">

```

```

    Tigre de Java
    <span class="badge">710</span>
  </a>
  <a href="#" class="list-group-item">
    Tigre de Sumatra
    <span class="badge">20</span>
  </a>
  <a href="#" class="list-group-item">
    Tigre du Bengale
    <span class="badge">40</span>
  </a>
  <a href="#" class="list-group-item">
    Tigre de la Caspienne
    <span class="badge">200</span>
  </a>
</div>
</div>

```

Ce qui donne la figure suivante.

Les sous-espèces des tigres :	
Tigre de Sibérie	120
Tigre de Chine méridionale	540
Tigre de Bali	230
Tigre de d'Indochine	240
Tigre de Malaisie	1200
Tigre de Java	710
Tigre de Sumatra	20
Tigre du Bengale	40
Tigre de la Caspienne	200

Liste avec badges de la page d'exemple

Les panneaux

Les panneaux permettent d'isoler visuellement une partie de la page avec une bordure esthétique, titre, contenu et pied de panneau. Voici un exemple élémentaire :

Code : HTML

```

<div class="panel panel-default">
  <div class="panel-heading">
    <h3 class="panel-title">Titre</h3>
  </div>
  <div class="panel-body">Contenu</div>
  <div class="panel-footer">Pied de panneau</div>

```

```
</div>
```

Ce qui donne la figure suivante.

Titre
Contenu
Pied de panneau

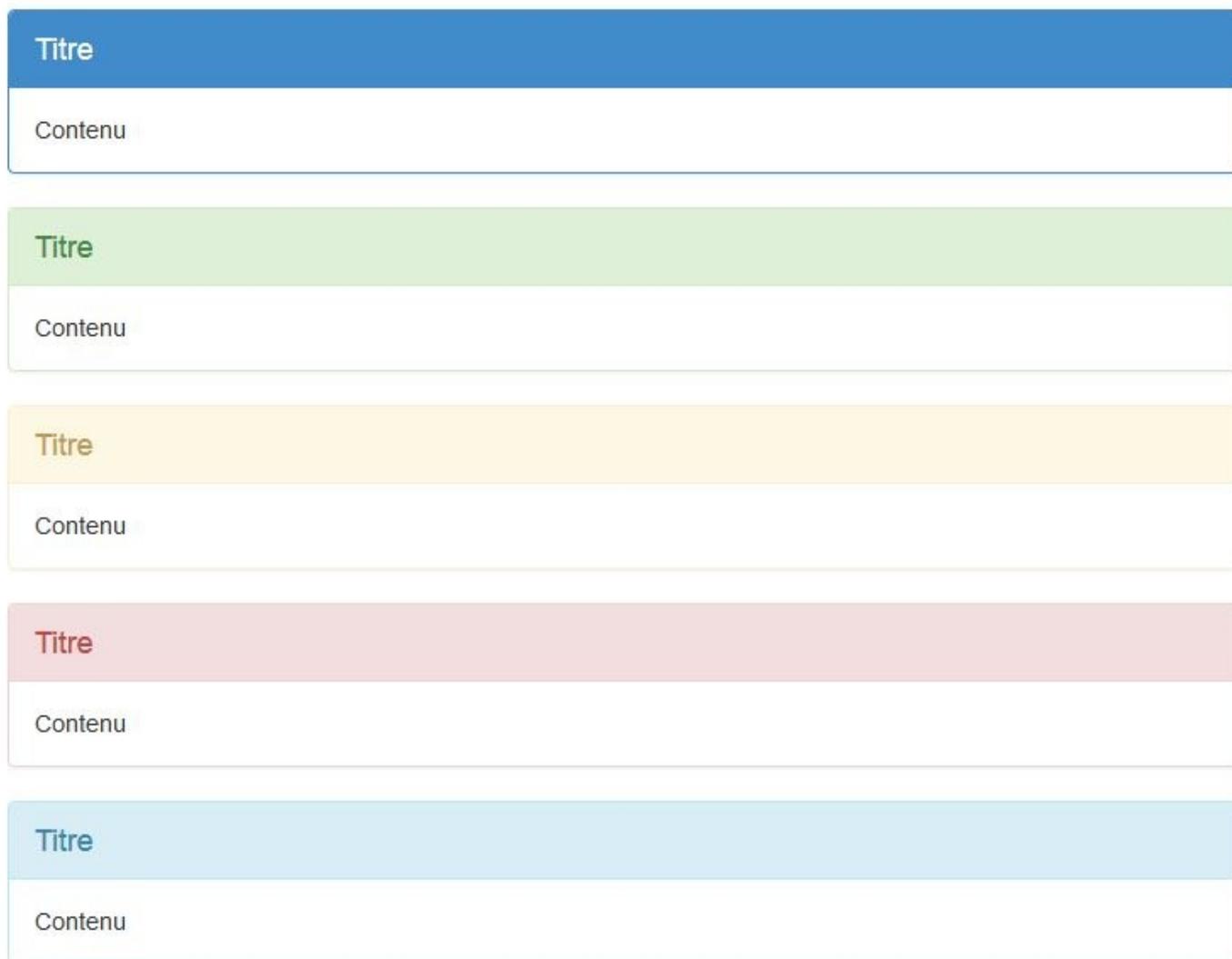
Un simple panneau

Il y a aussi des classes pour créer un sens visuel pour ces panneaux :

Code : HTML

```
<div class="panel panel-primary">  
  <div class="panel-heading">  
    <h3 class="panel-title">Titre</h3>  
  </div>  
  <div class="panel-body">Contenu</div>  
</div>  
<div class="panel panel-success">  
  <div class="panel-heading">  
    <h3 class="panel-title">Titre</h3>  
  </div>  
  <div class="panel-body">Contenu</div>  
</div>  
<div class="panel panel-warning">  
  <div class="panel-heading">  
    <h3 class="panel-title">Titre</h3>  
  </div>  
  <div class="panel-body">Contenu</div>  
</div>  
<div class="panel panel-danger">  
  <div class="panel-heading">  
    <h3 class="panel-title">Titre</h3>  
  </div>  
  <div class="panel-body">Contenu</div>  
</div>  
<div class="panel panel-info">  
  <div class="panel-heading">  
    <h3 class="panel-title">Titre</h3>  
  </div>  
  <div class="panel-body">Contenu</div>  
</div>
```

Ce qui donne la figure suivante.



Des panneaux stylisés

Utilisation avec une liste groupée

J'ai intégré une liste groupée dans un panneau au niveau de la page d'exemple :

Code : HTML

```
<div class="panel panel-info">
  <div class="panel-heading">
    <h3 class="panel-title">Les sous-espèces des tigres :</h3>
  </div>
  <div class="list-group">
    <a href="#" class="list-group-item">
      ...
    </a>
    <a href="#" class="list-group-item">
      ...
    </a>
    ...
  </div>
</div>
```

Ce qui donne la figure suivante.

Les sous-espèces des tigres :	
Tigre de Sibérie	120
Tigre de Chine méridionale	540
Tigre de Bali	230
Tigre de d'Indochine	240
Tigre de Malaisie	1200
Tigre de Java	710
Tigre de Sumatra	20
Tigre du Bengale	40
Tigre de la Caspienne	200

Liste groupée dans un panneau

Utilisation avec un tableau

J'ai intégré un tableau (sans bordure) dans un panneau au niveau de la page d'exemple :

Code : HTML

```
<div class="panel panel-primary">
  <table class="table table-striped table-condensed">
    <div class="panel-heading">
      <h3 class="panel-title">Les menaces pour les tigres</h3>
    </div>
    <thead>
      <tr>
        <th>Lieu</th>
        <th>Menace</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Grand Mekong</td>
        <td>Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures</td>
      </tr>
      <tr>
        <td>Île de Sumatra</td>
        <td>Production d'huile de palme et de pâtes à papiers</td>
      </tr>
      <tr>
        <td>Indonésie et Malaisie</td>
        <td>Pâte à papier, l'huile de palme et le caoutchouc</td>
      </tr>
      <tr>
        <td>États-Unis</td>
        <td>Les tigres captifs représentent un danger pour les
```

```

tigres sauvages</td>
</tr>
<tr>
  <td>Europe</td>
  <td>Gros appétit pour l'huile de palme</td>
</tr>
<tr>
  <td>Népal</td>
  <td>Commerce illégal de produits dérivés de tigres</td>
</tr>
</tbody>
</table>
</div>

```

Ce qui donne la figure suivante.

Les menaces pour les tigres	
Lieu	Menace
Grand Mekong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et fragmentation des habitats du fait du développement non durable d'infrastructures
Île de Sumatra	Production d'huile de palme et de pâtes à papiers
Indonésie et Malaisie	Pâte à papier, l'huile de palme et le caoutchouc
États-Unis	Les tigres captifs représentent un danger pour les tigres sauvages
Europe	Gros appétit pour l'huile de palme
Népal	Commerce illégal de produits dérivés de tigres

Tableau dans un panneau

Les médias

Simple insertion

L'objet média permet d'insérer un média (audio, image, vidéo...) à gauche ou à droite d'un texte. C'est ce que j'ai prévu dans la page d'exemple avec une vidéo de tigre :

Code : HTML

```

<div class="media col-lg-12">
  <div class="pull-right">
    <iframe width="420" height="320"
    src="http://www.youtube.com/embed/VmnIeLmjuHA" frameborder="0"
    allowfullscreen></iframe>
  </div>
  <div class="media-body pull-right">
    <h4 class="media-heading">Un tigre domestique dans son canapé
  </h4>
  Admirez ce noble animal...
  </div>
</div>

```

Ce qui donne la figure suivante.



Utilisation de la classe media

La classe `media` englobe l'ensemble. On utilise ensuite au choix `pull-right` ou `pull-left` pour le média. Le corps de composant est défini par la classe `media-body` et on peut mettre un titre avec `media-heading`.

Liste de médias

Une autre possibilité fort intéressante consiste à créer une liste de médias avec la classe `media-list`. Prenons par exemple le cas d'un forum, on peut imaginer ce type de scénario :

Code : HTML

```
<div class="container">
  <ul class="media-list col-lg-7">
    <li class="media thumbnail">
      <a class="pull-left" href="#">
        
      </a>
      <div class="media-body">
        <h4 class="media-heading">Suggestion</h4>
        <p>Je pense souhaitable d'interdire la chasse au tigre pour
préservier l'espèce !</p>
      <div class="media thumbnail">
        <a class="pull-left" href="#">
          
        </a>
        <div class="media-body">
          <h4 class="media-heading">Animaux dangereux</h4>
          <p>Tu délirés complètement, ce sont des animaux trop
dangereux pour les laisser vivre.</p>
        <div class="media thumbnail">
          <a class="pull-left" href="#">
            
          </a>
          <div class="media-body">
            <h4 class="media-heading">Quel délire !</h4>
            <p>C'est toi qui délirés ! Il faut préserver la bio-
diversité. Les tigres ne sont pas vraiment dangereux, c'est l'homme
qui l'est !</p>
          </div>
        </div>
      </div>
    </li>
  </ul>
  <div class="media">
    <a class="pull-left" href="#">
      
    </a>
  </div>
</div>
```

```

        </a>
        <div class="media-body">
          <h4 class="media-heading">Quel courage !</h4>
          <p>J'aimerais te voir face à un tigre toi !</p>
        </div>
      </div>
    </div>
    <div class="media thumbnail">
      <a class="pull-left" href="#">
        
      </a>
      <div class="media-body">
        <h4 class="media-heading">Un peu de calme</h4>
        <p>Je suis favorable à la préservation de l'espèce mais
dans des sites protégés.</p>
        <div class="media thumbnail">
          <a class="pull-left" href="#">
            
          </a>
          <div class="media-body">
            <h4 class="media-heading">Tu as vu ta tête !</h4>
            <p>Avec la tête que tu as tu ferais même peur à un
tigre toi !</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</li>
</ul>
</div>

```

Ce qui donne la figure suivante.



Suggestion

Je pense souhaitable d'interdire la chasse au tigre pour préserver l'espèce !



Animaux dangereux

Tu délires complètement, ce sont des animaux trop dangereux pour les laisser vivre.



Quel délire !

C'est toi qui délires ! Il faut préserver la bio-diversité. Les tigres ne sont pas vraiment dangereux, c'est l'homme qui l'est !



Quel courage !

J'aimerais te voir face à un tigre toi !



Un peu de calme

Je suis favorable à la préservation de l'espèce mais dans des sites protégés.



Tu as vu ta tête !

Avec la tête que tu as tu ferais même peur à un tigre toi !

Une liste de médias

[Tester](#)

Remarquez l'utilisation de la classe `thumbnail` pour bien isoler les commentaires et rendre le tout lisible.

Alertes et barres de progression

Nous allons voir maintenant 2 composants qui, bien que ne nécessitant pas de Javascript pour leur présentation, en ont besoin pour leur mise en œuvre.

Alertes

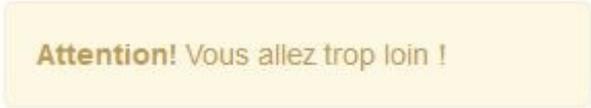
Utilisation de base

Les alertes permettent de présenter un message visuellement significatif. Il suffit d'utiliser les classes `alert` et `alert-warning`:

Code : HTML

```
<div class="alert alert-warning col-lg-3">
  <strong>Attention!</strong> Vous allez trop loin !
</div>
```

Ce qui donne la figure suivante.



Attention! Vous allez trop loin !

Une simple alerte

Les 3 variantes

Trois autres classes permettent de changer l'aspect de l'alerte selon le résultat désiré :

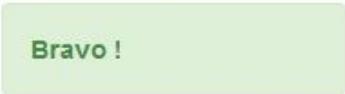
Code : HTML

```
<div class="alert col-lg-2 alert-info">
  <strong>Infos !</strong>
</div>
<div class="alert col-lg-offset-1 col-lg-2 alert-success">
  <strong>Bravo !</strong>
</div>
<div class="alert col-lg-offset-1 col-lg-2 alert-danger">
  <strong>Danger !</strong>
</div>
```

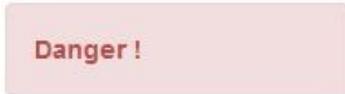
Ce qui donne la figure suivante.



Infos !



Bravo !



Danger !

Des alertes colorées

Un bloc d'information

Si le texte doit être conséquent, il suffit d'écrire le code que l'on désire :

Code : HTML

```
<div class="alert span5 alert-info">
  <h3>Information importante!</h3>
  L'utilisation de la classe <strong>alert</strong> n'est pas sans
  danger pour les yeux fatigués des pauvres internautes,
  et je vous demande instamment de l'utiliser avec parcimonie...
</div>
```

Ce qui donne la figure suivante.

Information importante!

L'utilisation de la classe **alert** n'est pas sans danger pour les yeux fatigués des pauvres internautes et je vous demande instamment de l'utiliser avec parcimonie...

Une alerte plus fournie

Un lien dans une alerte

La classe `alert-link` permet d'insérer un lien esthétique dans une alerte :

Code : HTML

```
<div class="alert alert-info col-lg-3">
  Un <a href="#" class="alert-link">joli lien</a> dans une
  alerte...
</div>
```

Ce qui donne la figure suivante.



Un bouton de fermeture

Une fois qu'on a lu une alerte, on peut avoir envie de la fermer. Il suffit de prévoir un bouton et d'ajouter la classe `alert-dismissible` :

Code : HTML

```
<div class="alert alert-warning alert-dismissible col-lg-4">
<button type="button" class="close" data-
dismiss="alert">&times;</button>
  <strong>Attention!</strong> Vous allez vraiment trop loin !
</div>
```

Ce qui donne la figure suivante.



Utilisation dynamique

Mais évidemment ces alertes ne trouvent leur sens que dans un contexte dynamique et nous devons pouvoir les faire apparaître et disparaître quand le besoin se présente. Bootstrap prévoit un petit plugin qui ne me paraît pas vraiment utile, étant donné qu'on peut facilement gérer les alertes avec quelques lignes de Javascript dopées par jQuery :

Code : HTML

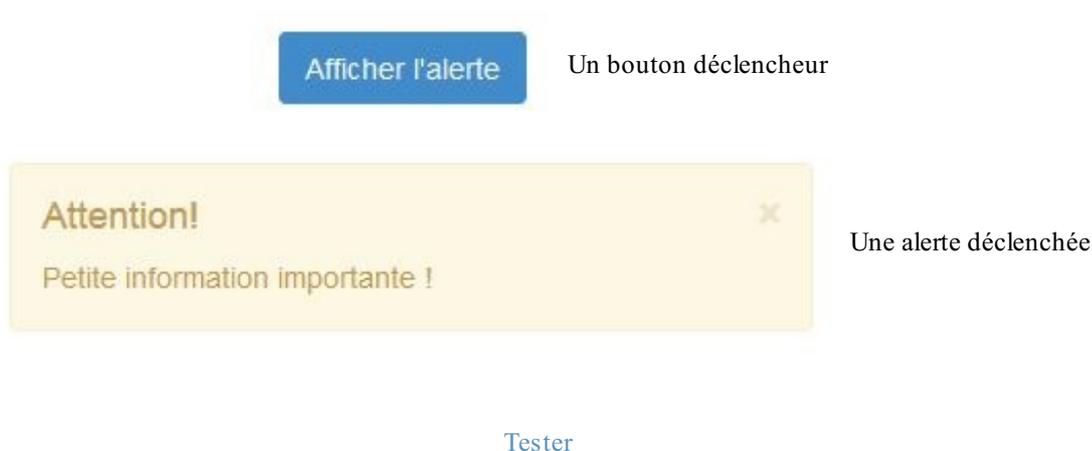
```
<div class="container">
  <div class="alert alert-warning alert-dismissible col-lg-4"
  style="display: none">
    <button type="button" class="close">x</button>
    <h4>Attention!</h4> Petite information importante !
  </div>
  <div class="col-lg-3">
    <input type="button" class="btn btn-primary" id="afficher"
    value="Afficher l'alerte">
  </div>
```

```

</div>
<script src="assets/js/jquery.js"></script>
<script>
  $(function () {
    $("#afficher").click(function () {
      $("#afficher").hide();
      $(".alert").show("slow");
    });
    $(".close").click(function () {
      $(".alert").hide("slow");
      $("#afficher").show();
    });
  });
</script>

```

Ce qui donne la figure suivante.



Un bouton adapté par la classe `close` a été prévu pour la fermeture de l'alerte. On pourrait également mettre un bouton normal.

Barres de progression

Les barres de progression constituent une façon élégante de faire patienter l'utilisateur pendant une longue tâche. Elles ont aussi pour vocation de le rassurer et de lui indiquer approximativement le temps qu'il lui reste à attendre. Les classes `progress` et `progress-bar` permettent de réaliser facilement ce genre d'effet. Pour les exemples, je prévois un peu de code javascript pour simuler un processus temporel. Voici la barre de base :

Code : HTML

```

<div class="container">
  <div class="progress">
    <div class="progress-bar"></div>
  </div>
  <input type="button" class="btn btn-primary" id="animer"
  value="Animer">
</div>
<script src="assets/js/jquery.js"></script>
<script>
  function timer(n) {
    $(".progress-bar").css("width", n + "%");
    if(n < 100) {
      setTimeout(function () {
        timer(n + 10);
      }, 200);
    }
  }
  $(function () {
    $("#animer").click(function () {
      timer(0);
    });
  });
</script>

```

```

        timer(0);
    });
});
</script>

```

Ce qui donne la figure suivante.



Barre de progression simple

[Tester](#)

Le code correspondant à la barre est surligné, le reste est juste l'intendance pour l'animer pour l'exemple. On se contente de faire évoluer la propriété `width` de la barre.

Il existe une version rayée de la barre avec la classe `progress-striped` :

Code : HTML

```

<div class="progress progress-striped">
  <div class="progress-bar"></div>
</div>

```

Ce qui donne la figure suivante.



Barre de progression rayée

[Tester](#)

Il existe aussi une version rayée et élégamment animée avec la classe `active` :

Code : HTML

```

<div class="container">
  <div class="progress progress-striped active">
    <div class="progress-bar"></div>
  </div>
  <div id="pourcentage" class="pull-right"></div>
  <input type="button" class="btn btn-primary" id="animer"
value="Animer">
</div>
<script src="assets/js/jquery.js"></script>
<script>
  function timer(n) {
    $(".progress-bar").css("width", n + "%");
    $("#pourcentage").text(n + "%");
  }
  timer(0);
</script>

```

```

    if(n < 100) {
      setTimeout(function() {
        timer(n + 10);
      }, 200);
    }
  }
  $(function () {
    $("#animer").click(function() {
      timer(0);
    });
  });
</script>

```

Ce qui donne la figure suivante, même si sur l'image vous ne voyez pas l'animation...



Barre de progression rayée et animée

[Tester](#)

Pour changer un peu, j'ai prévu l'affichage du pourcentage. Remarquez l'utilisation de la classe `pull-right` pour envoyer cet affichage sur la droite.

Il est également prévu des couleurs différentes pour enrichir la sémantique de ces barres, que ce soit en version unie ou rayée :

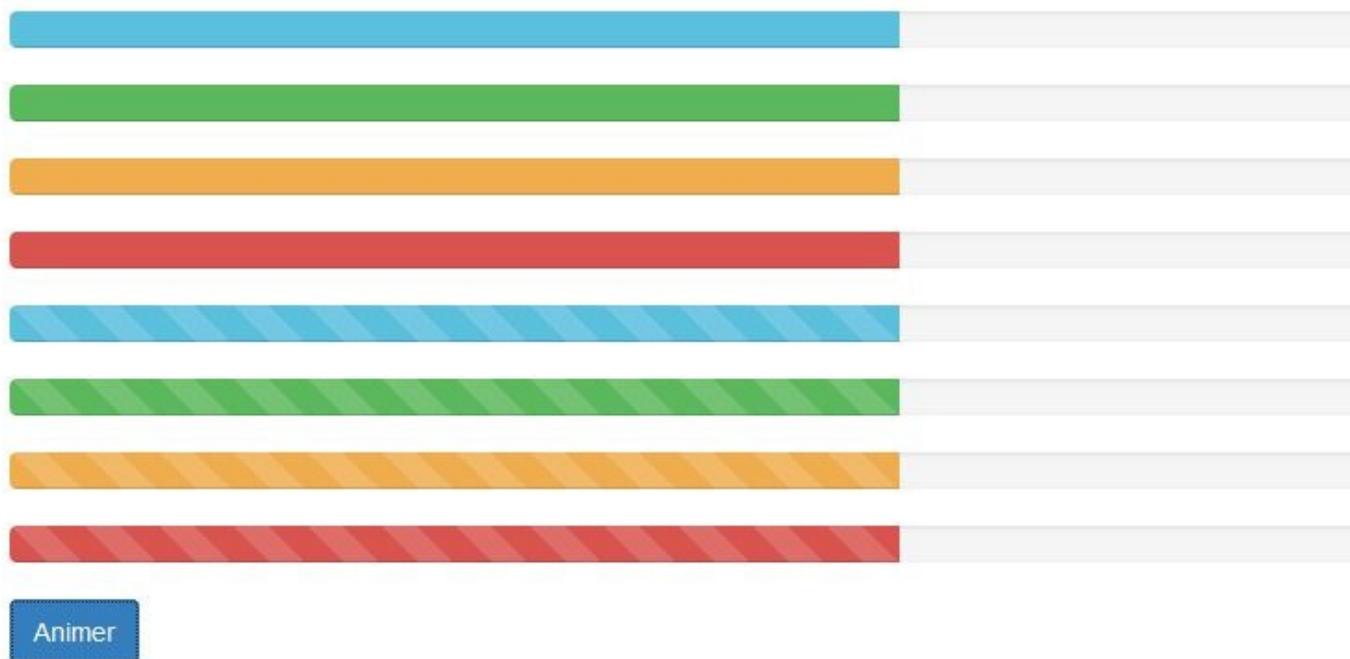
Code : HTML

```

<div class="container">
  <div class="progress">
    <div class="progress-bar progress-bar-info"></div>
  </div>
  <div class="progress">
    <div class="progress-bar progress-bar-success"></div>
  </div>
  <div class="progress">
    <div class="progress-bar progress-bar-warning"></div>
  </div>
  <div class="progress">
    <div class="progress-bar progress-bar-danger"></div>
  </div>
  <div class="progress progress-striped">
    <div class="progress-bar progress-bar-info"></div>
  </div>
  <div class="progress progress-striped">
    <div class="progress-bar progress-bar-success"></div>
  </div>
  <div class="progress progress-striped">
    <div class="progress-bar progress-bar-warning"></div>
  </div>
  <div class="progress progress-striped">
    <div class="progress-bar progress-bar-danger"></div>
  </div>
  <input type="button" class="btn btn-primary" id="animer"
value="Animer">
</div>

```

Ce qui donne la figure suivante.



Des barres colorées

[Tester](#)

Une dernière possibilité, enfin, consiste à empiler des barres :

Code : HTML

```
<div class="container">
  <div class="progress">
    <div class="progress-bar progress-bar-success"></div>
    <div class="progress-bar progress-bar-warning"></div>
    <div class="progress-bar progress-bar-danger"></div>
  </div>
  <input type="button" class="btn btn-primary" id="animer"
value="Animer">
</div>
<script src="assets/js/jquery.js"></script>
<script>
  function timer(n) {
    $(".progress-bar").css("width", n + "%");
    if(n < 33.2) {
      setTimeout(function() {
        timer(n + 3.33);
      }, 200);
    }
  }
  $(function () {
    $("#animer").click(function() {
      timer(0);
    });
  });
</script>
```

Ce qui donne la figure suivante.



Des barres empilées

[Tester](#)

Dans cette version empilée, on conserve la possibilité d'avoir des rayures, animées ou pas...

En résumé :

- Bootstrap permet de créer une barre de navigation fixe ou mobile qui peut comporter un titre, des liens, des boutons et des formulaires.
- Bootstrap possède un composant pour la navigation entre pages web.
- Pour des effets visuels Bootstrap propose son Jumbotron, des libellés, des badges et des thumbnails pour les images.
- Les listes groupées peuvent comporter des badges, des liens, des images et être colorées.
- Les panneaux permettent d'isoler visuellement une partie de la page et peuvent comporter un titre, ils peuvent intégrer facilement une liste groupée ou un tableau.
- Le composant `media` permet de proposer des images, des musiques, des vidéos.
- Bootstrap permet l'affichage de bandeaux d'alerte colorés et de barres de progression.

Partie 3 : Mise en forme avancée

Dans cette partie, nous verrons comment animer un peu un site web avec jQuery, le célèbre framework javascript, et nous étudierons l'intérêt de LESS dans la construction des feuilles de style en cascade avec un cas pratique d'application pour modifier Bootstrap.

Les plugins jQuery

jQuery est un framework Javascript devenu incontournable au fil des années. Les créateurs de Bootstrap l'ont choisi pour offrir des plugins esthétiques et faciles à mettre en œuvre, dans la plupart des cas sans même mettre les doigts dans du code Javascript ! Toutefois, si vous voulez obtenir le meilleur de ces plugins, il vous faudra un peu manipuler jQuery.

J'ai traité les plugins les plus importants en délaissant Affix qui est à la fois simple d'utilisation et d'un usage limité. Une page d'exemple servira à la fois de collecteur d'exemples pratiques et de fil conducteur.

Généralités

La librairie jQuery



Javascript, c'est quoi ?

- C'est un langage de script qui permet de rendre les pages web interactives.
- C'est un langage qui fonctionne côté client et ne demande aucune ressource au niveau du serveur.
- C'est un langage interprété directement par le navigateur.



jQuery, c'est quoi ?

jQuery est une librairie Javascript, on dit aussi un framework, c'est à dire un ensemble de fonctionnalités bien pratiques pour simplifier la mise en œuvre de Javascript. On peut ainsi facilement :

- Modifier des éléments d'une page web : texte, style, animation...
- Gérer des événements tels : clic de souris, action sur une touche du clavier...
- Créer des effets graphiques...
- Utiliser Ajax...

Installation de jQuery

Pour installer jQuery il faut faire un appel à la bibliothèque. Pour cela vous avez deux solutions :

1. Vous allez chercher la dernière version sur le site <http://jquery.com/download/> et vous copiez ce fichier dans un répertoire de votre site.
2. Vous référencez un CDN : jQuery, cdnjs, Google CDN, Microsoft CDN...

Si vous choisissez de télécharger la librairie, vous trouvez 2 lignes de versions : la 1 et la 2. Ces 2 versions comportent les mêmes fonctionnalités, la seule différence est que la seconde ne prend pas en charge Internet Explorer dans ses versions 6 à 8. Vous avez donc le choix de la version, les 2 étant parfaitement adaptées à Bootstrap 3,

Vous trouvez aussi 2 possibilités au niveau du téléchargement :

- Uncompressed : c'est une version lisible et commentée pour comprendre le code. Vous l'utilisez pendant le développement de vos sites si vous avez besoin de savoir ce qui se passe dans la bibliothèque.
- Minified : c'est une version compressée pour être la plus "légère" possible. C'est cette version que vous devez privilégier pour la mettre sur votre site en "production"

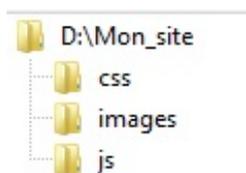
Quand vous cliquez sur le lien, vous obtenez directement le code de la figure suivante dans votre navigateur.

```
/*! jQuery v2.1.0 | (c) 2005, 2014 jQuery Foundation, Inc. | jquery.o
!function(a,b){"object"===typeof module&&"object"===typeof module.expor
b(a):b(a)}("undefined"!==typeof window?window:this,function(a,b){var
{}},m=a.document,n="2.1.0",o=function(a,b){return new o.fn.init(a,b)},
{jquery:n,constructor:o,selector:"",length:0,toArray:function(){retur
b=o.merge(this.constructor(),a);return b.prevObject=this,b.context=th
a.call(b,c,b))}),slice:function(){return this.pushStack(d.apply(this
return this.pushStack(c>=0&&b>c?[this[c]]:[])),end:function(){return
a,b,c,d,e,f,g=arguments[0]||{},h=1,i=arguments.length,j=!1;for("boole
```

Faites un "enregistrer sous" en donnant le nom `jquery.js` pour la version complète et `jquery.min.js` la version compressée.

Placer jQuery

Il est bon d'organiser un peu les fichiers d'un site en répertoires du genre (voir figure suivante).



Organisation des dossiers d'un site

Si vous organisez ainsi l'architecture de votre site, déposez les fichiers jQuery dans le répertoire `js`.

Référencer jQuery

Si vous avez placé la librairie sur votre site, il suffit de référencer le fichier dans le code HTML, de préférence à la fin pour ne pas péjorer le chargement des éléments visuels :

Code : HTML

```
<script src="js/jquery.js"></script>
```

Si vous passez par un CDN, c'est la même chose, à part l'adresse qui change. Par exemple :

Code : HTML

```
<script src="http://code.jquery.com/jquery-2.1.0.min.js"></script>
```

Utiliser jQuery

La principale fonction à connaître est `jQuery()`, mais on utilise le symbole `$` pour faire plus court. On veut utiliser jQuery seulement lorsque le DOM de la page est totalement chargé pour éviter les erreurs d'exécution, ainsi on encapsule tout le code dans un bloc :

Code : JavaScript

```
$(function() {  
    // ...  
});
```

Si vous connaissez les sélecteurs CSS, vous savez déjà utiliser une grande partie de jQuery. En effet, cette librairie les utilise directement ! Par exemple :

Code : JavaScript

```
var a = $("div");
```

Ici la variable "a" référence un objet jQuery qui sélectionne toutes les balises `div` du document.

jQuery propose aussi des méthodes, par exemple pour ajouter une classe à un élément :

Code : JavaScript

```
$("#img").addClass( "ma_classe" );
```

Cette action ajoute la classe `ma_classe` à toutes les balises `img` du document. Il est possible de chaîner les méthodes. Par exemple :

Code : JavaScript

```
$('#el').css('background-color', 'red').fadeIn('slow');
```

Ici on change la propriété `background-color` de l'élément avec l'identifiant `#el`, avec la valeur `red`, et on le fait apparaître progressivement.

Une grande force de jQuery réside aussi dans sa gestion facilitée des événements, par exemple :

Code : JavaScript

```
$("#action").click(function() {  
    ... mon code ...  
});
```

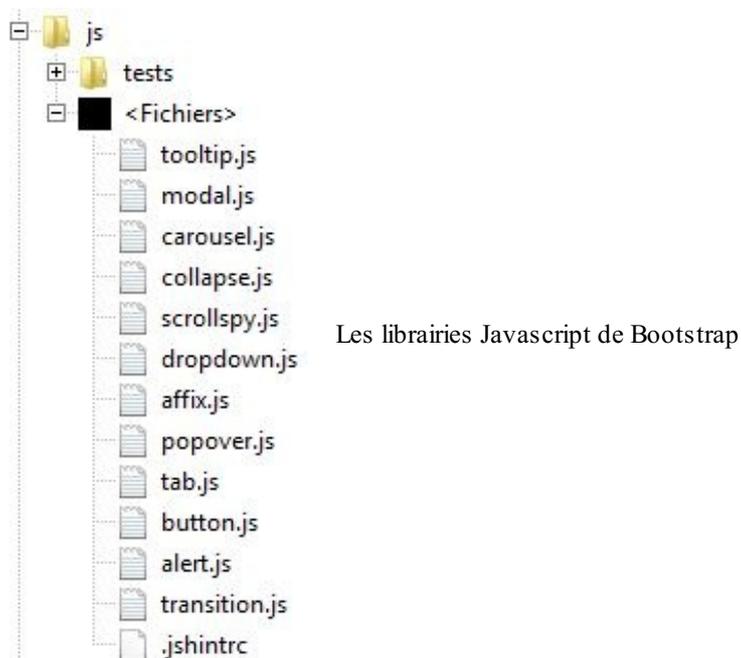
Ici, un clic sur l'élément d'identifiant `#action` va provoquer l'exécution du code.

Les bibliothèques Javascript de Bootstrap

Référencement

Bootstrap propose un certain nombre de plugins pour améliorer les pages web, ils sont tous fondés sur jQuery (voir figure suivante). Il y a deux façons de faire appel à ces plugins :

- soit vous référencez juste ce qui vous est utile en prenant la librairie correspondante dans le pack de Bootstrap (que vous récupérez en cliquant sur "Download source" sur la page "Getting Started" comme à la figure suivante).
- soit vous référencez la bibliothèque complète (que vous récupérez en cliquant sur "Download Bootstrap" sur la page "Getting Started").



Dans la première hypothèse, il faut faire attention aux dépendances. Elles ne sont pas nombreuses mais il faut les connaître :

- popover doit être utilisé avec tooltip
- collapse doit être utilisé avec transition

Pour référencer les plugins dans l'hypothèse de l'utilisation individuelle, par exemple pour tab :

Code : HTML

```
<script src="js/jquery.js"></script>
<script src="js/tab.js "></script>
```

On déclare d'abord jQuery, puis ensuite le plugin. Dans l'hypothèse de la librairie globale :

Code : HTML

```
<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js "></script>
```

Vous pouvez utiliser la version commentée pendant la phase de développement :

Code : HTML

```
<script src="js/jquery.js"></script>
<script src="js/bootstrap.js "></script>
```

Mise en œuvre

Il y a 2 façons d'utiliser un plugin de Bootstrap :

- à partir des attributs data
- à partir d'objets jQuery

Prenons un exemple avec l'effet accordéon que nous verrons en détail ultérieurement :

Code : HTML

```
<a href="#item" data-toggle="collapse">Changer</a>
<div id="item" class="collapse">
  Contenu
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
```

Ne vous inquiétez pas pour la syntaxe globale que nous verrons plus tard. Ce plugin est destiné à faire apparaître ou disparaître un contenu lors d'une action de l'utilisateur, ici un simple clic. Le contenu concerné dans cet exemple est le mot Contenu. Le plugin est déclenché par l'attribut `data-toggle="collapse"`. Dans cette hypothèse d'utilisation, vous n'avez pas à manipuler de Javascript.

On peut obtenir exactement le même résultat en utilisant jQuery :

Code : HTML

```
<a>Changer</a>
<div id="item" class="collapse">
  Contenu
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
  $(function() {
    $('a').click(function() {
      $('#item').collapse('toggle');
    });
  });
</script>
```

Cette fois je n'ai pas utilisé l'attribut mais un objet jQuery avec le sélecteur 'a' pour pointer le lien et la méthode `collapse` avec le paramètre `toggle`. Nous verrons plusieurs cas d'utilisation de cette façon de procéder qui est plus laborieuse que la précédente, mais qui s'avère incontournable selon les actions que nous voulons effectuer.

Les événements

La plupart des plugins exposent des événements. En poursuivant l'exemple précédent, on peut faire apparaître un message lorsque le contenu est visible :

Code : HTML

```
<a>Changer</a>
<div id="item" class="collapse">
  Contenu
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
  $(function() {
    $('a').click(function() {
      $('#item').collapse('toggle');
    });
    $('#item').on('shown.bs.collapse', function () {
      alert('On me voit !');
    });
  });
</script>
```

```
</script>
```

Nous verrons aussi plusieurs exemples d'événements dans les plugins de Bootstrap.

La page d'exemple

Pour éviter de surcharger de code les chapitres sur les plugins, voici la page d'exemple complétée avec des cas d'utilisation des plugins concernés :

Code : HTML

```
<!DOCTYPE HTML>
<html>

  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      body {
        background-color:#DDD;
        padding-top: 10px;
      }
      [class*="col-"] { margin-bottom: 20px; }
      img { width: 100%; }
      .well {
        background-color:#CCC;
        padding: 20px;
      }
      a:active, a:focus { outline:none; }
    </style>
  </head>

  <body>
    <div class="container">
      <nav class="navbar navbar-inverse" role="navigation">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="#">Les Tigres</a>
        </div>
        <div class="collapse navbar-collapse">
          <ul class="nav navbar-nav">
            <li><a href="#">Accueil</a> </li>
            <li class="dropdown">
              <a class="dropdown-toggle" data-toggle="dropdown"
href="#">Témoignages <b class="caret"></b></a>
              <ul class="dropdown-menu">
                <li><a href="#">Dompteurs</a></li>
                <li><a href="#">Zoos</a></li>
                <li><a href="#">Chasseurs</a></li>
                <li class="divider"></li>
                <li><a href="#">Autres témoignages</a></li>
              </ul>
            </li>
            <li><a href="#">Liens</a> </li>
            <li><a href="#">Références</a> </li>
          </ul>
          <form class="navbar-form navbar-right">
            <div class="input-group">
              <input type="text" style="width:150px" class="input-sm
form-control" placeholder="Recherche">
              <span class="input-group-btn">
```

```

        <span class="input-group-btn" >
          <button type="submit" class="btn btn-primary btn-
sm"><span class="glyphicon glyphicon-eye-open"></span>
Chercher</button>
        </span>
      </div>
    </form>
  </div>
</nav>
<header class="row">
  <div class="col-lg-12">
    <h1>Mon amour pour les tigres</h1>
  </div>
</header>
<section class="row">
  <div class="col-lg-12">
    <p>
      Je suis passionné par les <strong>tigres</strong> depuis
      très longtemps. Ce site a été construit en <em>hommage à ces
      merveilleux félins...</em>
    <br>
      Je fais partie de la <abbr title="Société des Amoureux
      des Tigres">SAT</abbr> qui a pour but de faire connaître ces
      splendides animaux.
    </p>
    <p>Voici ce qu'en dit le wikipedia :</p>
    <blockquote>
      Le Tigre (Panthera tigris) est un <a data-
toggle="tooltip" href="#" title="Classe de vertébrés">mammifère</a>
      carnivore de la famille
      des félidés (Felidae) du genre Panthera. Aisément reconnaissable à
      sa fourrure rousse rayée de noir, il est le plus grand félin sauvage
      et l'un
      des plus grands <a data-toggle="tooltip"
href="#" title='Synonyme de "carnassier"'>carnivores </a>du monde.
      L'espèce est divisée en neuf sous-espèces possédant des différences
      mineures
      en termes de taille ou de comportement. Superprédateur, il chasse
      principalement les cerfs et les sangliers, bien qu'il puisse
      s'attaquer à des
      proies de taille plus importante comme les buffles. Jusqu'au XIXe
      siècle, le Tigre était réputé mangeur d'homme. La structure sociale
      des tigres
      en fait un animal solitaire ; le mâle possède un territoire qui
      englobe les domaines de plusieurs femelles et ne participe pas à
      l'éducation des petits.<br>
      <small class="pull-right">Wikipedia</small><br>
    </blockquote>
  </div>
</section>
<section class="row">
  <div class="media col-lg-12">
    <div class="pull-right">
      <iframe width="420" height="320"
src="http://www.youtube.com/embed/VmnIeLmjuHA" frameborder="0"
allowfullscreen></iframe>
    </div>
    <div class="media-body pull-right">
      <div class="btn-group" data-toggle="buttons">
        <label class="btn btn-primary">
          <input type="radio"> Vidéo 1
        </label>
        <label class="btn btn-primary">
          <input type="radio"> Vidéo 2
        </label>
        <label class="btn btn-primary">
          <input type="radio"> Vidéo 3
        </label>
      </div>
    </div>
  </div>
</section>

```

```
    </div>
  </section>
  <section class="row">
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
    <div class="col-xs-4 col-sm-3 col-md-2 ">
      <a href="#" class="thumbnail">
        
      </a>
    </div>
  </div>
```

```

    <a href="#" class="thumbnail">
      
    </a>
  </div>
</section>
<div class="row">
  <section class="col-sm-4">
    <div class="panel panel-info">
      <div class="panel-heading">
        <h3 class="panel-title">Les sous-espèces des tigres
:</h3>
      </div>
      <div class="list-group">
        <a href="#infos" class="list-group-item" data-
toggle="modal">
          Tigre de Sibérie
          <span class="badge">120</span>
        </a>
        <div class="modal fade" id="infos" role="dialog" aria-
labelledby="modalTitre" aria-hidden="true">
          <div class="modal-dialog">
            <div class="modal-content">
              <div class="modal-header">
                <button type="button" class="close" data-
dismiss="modal" aria-hidden="true">x</button>
                <h4 id="modalTitre" class="modal-title">Plus
d'informations sur le tigre de Sibérie</h4>
              </div>
              <div class="modal-body">
                <blockquote>
                  Les tigres de Sibérie pèsent de 180 à 350 kg
pour les mâles et de 100 à 200 kg pour les femelles. Le plus gros
tigre de Sibérie,
                  tué en 1950, atteignait les 384 kg. Le Tigre de Sibérie est le
troisième plus gros prédateur terrestre derrière l'ours kodiak et
l'ours polaire.
                  La longueur totale du corps avec la queue est comprise entre 2,7 et
3,8 m pour les mâles et entre 2,4 et 2,75 m pour les femelles. La
longueur
                  du crâne est de 341 à 383 mm pour les mâles et 279 à 318 mm pour les
femelles.<hr>
                  <small class="pull-right">Wikipedia</small>
                </blockquote>
              </div>
            </div>
          </div>
        </div>
      </div>
      <a href="#" class="list-group-item">
        Tigre de Chine méridionale
        <span class="badge">540</span>
      </a>
      <a href="#" class="list-group-item">
        Tigre de Bali
        <span class="badge">230</span>
      </a>
      <a href="#" class="list-group-item">
        Tigre de d'Indochine
        <span class="badge">240</span>
      </a>
      <a href="#" class="list-group-item">
        Tigre de Malaisie
        <span class="badge">1200</span>
      </a>
      <a href="#" class="list-group-item">
        Tigre de Java
        <span class="badge">710</span>
      </a>
      <a href="#" class="list-group-item">
        Tigre de Sumatra

```

```

        <span class="badge">20</span>
      </a>
      <a href="#" class="list-group-item">
        Tigre du Bengale
        <span class="badge">40</span>
      </a>
      <a href="#" class="list-group-item">
        Tigre de la Caspienne
        <span class="badge">200</span>
      </a>
    </div>
  </div>
</section>
<section class="col-sm-8">
  <div id="carousel" class="carousel slide">
    <ol class="carousel-indicators">
      <li data-target="#carousel" data-slide-to="0"
class="active"></li>
      <li data-target="#carousel" data-slide-to="1"></li>
      <li data-target="#carousel" data-slide-to="2"></li>
      <li data-target="#carousel" data-slide-to="3"></li>
    </ol>
    <div class="carousel-inner">
      <div class="item active"> </div>
      <div class="item"> </div>
      <div class="item"> </div>
      <div class="item"> </div>
    </div>
  </div>
</section>
</div>
<div class="row">
  <section class="col-sm-12">
    <div class="panel panel-primary">
      <table class="table table-striped table-condensed">
        <div class="panel-heading">
          <h3 class="panel-title">Les menaces pour les
tigres</h3>
        </div>
        <thead>
          <tr>
            <th>Lieu</th>
            <th>Menace</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Grand Mekong</td>
            <td>Demande croissante de certaines parties de
l'animal pour la médecine chinoise traditionnelle et fragmentation
des habitats du
fait du développement non durable d'infrastructures</td>
          </tr>
          <tr>
            <td>Île de Sumatra</td>
            <td>Production d'huile de palme et de pâtes à
papiers</td>
          </tr>
          <tr>
            <td>Indonésie et Malaisie</td>
            <td>Pâte à papier, l'huile de palme et le
caoutchouc</td>
          </tr>
          <tr>
            <td>États-Unis</td>
            <td>Les tigres captifs représentent un danger pour

```

```

les tigres sauvages</td>
    </tr>
    <tr>
        <td>Europe</td>
        <td>Gros appétit pour l'huile de palme</td>
    </tr>
    <tr>
        <td>Népal</td>
        <td>Commerce illégal de produits dérivés de
tigres</td>
    </tr>
</tbody>
</table>
</div>
</section>
</div>
<div class="row">
    <section class="col-sm-8">
        <ul class="nav nav-pills">
            <li class="active"><a href="#contact" data-
toggle="tab">Contact</a></li>
            <li><a href="#informations" data-
toggle="tab">Informations pratiques</a></li>
        </ul>
        <div class="tab-content">
            <div class="tab-pane active fade in" id="contact">
                <form class="well">
                    <h4><span class="label label-default">Si vous voulez
me laisser un message</span></h4>
                    <h4>Comment m'avez-vous trouvé ?</h4>
                    <fieldset>
                        <label for="ami" class="radio">
                            <input type="radio" name="origine" value="ami"
id="ami">
                                Par un ami
                            </label>
                        <label for="web" class="radio">
                            <input type="radio" name="origine" value="web"
id="web">
                                Sur le web
                            </label>
                        <label for="hasard" class="radio">
                            <input type="radio" name="origine"
value="hasard" id="hasard">
                                Par hasard
                            </label>
                        <label for="autre" class="radio">
                            <input type="radio" name="origine" value="autre"
id="autre">
                                Autre...
                            </label>
                        <label for="textarea">Votre message :</label>
                        <textarea id="textarea" rows="4" class="form-
control"></textarea>
                        <p class="help-block">Vous pouvez agrandir la
fenêtre</p>
                        <button class="btn btn-primary"
type="submit"><span class="glyphicon glyphicon-ok-sign"></span>
Envoyer</button>
                    </fieldset>
                </form>
            </div>
            <div class="tab-pane fade" id="informations">
                <div class="well">
                    <h2>Informations pratiques</h2>
                    <p>Si vous êtes intéressés par la protection des
tigres et si vous désirez prendre part aux actions menées...
blablaba...</p>
                </div>
            </div>
        </div>
    </section>
</div>

```

```

    <div class="tab-content">
  </section>
  <section class="col-sm-4">
    <address class="thumbnail">
      <p>Vous pouvez me contacter à cette adresse :</p>
      <strong>Tigrou Alfred</strong><br>
      Allée des fauves<br>
      28645 Félines-sur-Loire<br>
    </address>
  </section>
</div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
  $(function () {
    $('.carousel').carousel();
    $('blockquote a').tooltip();
    $('.btn-group .btn:nth-child(1)').click(function () {
      $('iframe').attr('src',
'http://www.youtube.com/embed/VmnIeLmjuHA');
    })
    $('.btn-group .btn:nth-child(2)').click(function () {
      $('iframe').attr('src',
'http://www.youtube.com/embed/HQU2ZFmMbJg');
    })
    $('.btn-group .btn:nth-child(3)').click(function () {
      $('iframe').attr('src',
'http://www.youtube.com/embed/qDBKzdNpiy0');
    })
  });
</script>
</body>
</html>

```

Tester

Je détaillerai dans chaque chapitre la mise en œuvre des différents plugins utilisés dans cette page.

Menu déroulant

Menu déroulant dans la barre de navigation

Le plugin jQuery Dropdown permet de faire de jolis menus déroulants. Il suffit de modifier une barre de navigation pour ajouter un menu déroulant (voir figure suivante) :

Code : HTML

```

<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
  </head>
  <body>
    <div class="container">
      <nav class="navbar navbar-default">
        <ul class="nav navbar-nav">
          <li><a href="#">Accueil</a> </li>
          <li><a href="#">Liens</a> </li>
          <li class="dropdown">
            <a data-toggle="dropdown" href="#">Témoignages<b
class="caret"></b></a>
            <ul class="dropdown-menu">
              <li><a href="#">Dompteurs</a></li>
              <li><a href="#">Zoos</a></li>
              <li><a href="#">Chasseurs</a></li>

```

```

<li class="divider"></li>
<li><a href="#">Autres témoignages</a></li>
</ul>
</li>
      <li> <a href="#">Références</a> </li>
    </ul>
  </nav>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>
</html>

```



Menu déroulant dans une barre de navigation

Il a fallu ajouter quelques classes mais le résultat est très réussi !

Voyons ça de plus près : pour ajouter la fonctionnalité de dropdown, il faut utiliser la classe `dropdown` et ajouter `data-toggle="dropdown"` dans la balise concernée pour que le menu fonctionne. Ici, je l'ai fait pour la balise `<a>` qui contient le libellé du lien du menu. Il faut aussi utiliser la classe `dropdown-menu` dans la balise `` qui doit englober le menu déroulant.

C'est ce qui a été réalisé dans [la page d'exemple](#).

Menu déroulant dans un bouton

On peut créer un menu déroulant activé par un bouton. Voici un code de base :

Code : HTML

```

<div class="btn-group">
  <button class="btn btn-default dropdown-toggle" data-
toggle="dropdown"> Action</button>
  <ul class="dropdown-menu">
    <li><a href="#">Dompteurs</a></li>
    <li><a href="#">Zoos</a></li>
    <li><a href="#">Chasseurs</a></li>
    <li class="divider"></li>
    <li><a href="#">Autres témoignages</a></li>
  </ul>
</div>

```

Vous remarquez l'utilisation de la classe `btn-group` nécessaire au fonctionnement du plugin, ainsi que la classe `dropdown-toggle`, mais pour le reste c'est pratiquement identique au code qu'on a vu pour la barre de navigation. Vous trouverez le résultat à la figure suivante.



Menu déroulant dans un bouton



C'est pas mal, mais comme ça, on ne comprend pas vraiment que c'est un bouton spécial, on peut avoir un repère graphique ?

On peut, en ajoutant une classe :

Code : HTML

```
<button class="btn btn-default dropdown-toggle" data-toggle="dropdown">Action <span class="caret"></span></button>
```

J'ai ajouté la classe `caret` pour le bouton et maintenant celui-ci est bien identifié, comme à la figure suivante.



Menu déroulant dans un bouton avec repère



Oui c'est bien, mais parfois j'ai besoin que le menu s'affiche vers le haut

On peut aussi le faire, en ajoutant la classe `dropup` (voir figure suivante) :

Code : HTML

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



Menu se déroulant vers le haut

Des boutons colorés et des icônes

On peut aussi ajouter des icônes et avoir de la couleur, comme pour tout bouton (voir figure suivante) :

Code : HTML

```
<div class="btn-group">
  <button class="btn btn-primary dropdown-toggle" data-
toggle="dropdown">Action <span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#"><span class="glyphicon glyphicon-user"></span>
Dompteurs</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
picture"></span> Zoos</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
screenshot"></span> Chasseurs</a></li>
    <li class="divider"></li>
    <li><a href="#"><span class="glyphicon glyphicon-list-
alt"></span> Autres témoignages</a></li>
  </ul>
</div>
```



Des icônes dans un menu déroulant

Séparer les fonctions

On peut faire en sorte que le bouton ait 2 usages : en tant que bouton pour une action et en tant que déclencheur de menu déroulant. Il est alors séparé en 2 parties : à gauche pour l'action, à droite pour le menu. Pour réaliser cela, on crée un groupe de 2 boutons, un avec le texte, l'autre avec le signe, comme à la figure suivante :

Code : HTML

```
<div class="btn-group">
  <button class="btn btn-info">Action</button>
  <button class="btn btn-info dropdown-toggle" data-
```

```
toggle="dropdown"><span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#"><span class="glyphicon glyphicon-user"></span>
Dompteurs</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
picture"></span> Zoos</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
screenshot"></span> Chasseurs</a></li>
    <li class="divider"></li>
    <li><a href="#"><span class="glyphicon glyphicon-list-
alt"></span> Autres témoignages</a></li>
  </ul>
</div>
```



Deux fonctions pour un bouton

En-têtes

Pour ajouter des en-têtes pour subdiviser le menu, il faut utiliser la classe `dropdown-header` (voir figure suivante) :

Code : HTML

```
<div class="btn-group">
  <button class="btn btn-info">Action</button>
  <button class="btn btn-info dropdown-toggle" data-
toggle="dropdown"><span class="caret"></span></button>
  <ul class="dropdown-menu">
<li class="dropdown-header"> Personnel</li>
    <li><a href="#"><span class="glyphicon glyphicon-user"></span>
Dompteurs</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
screenshot"></span> Chasseurs</a></li>
    <li class="divider"></li>
<li class="dropdown-header"> Lieux</li>
    <li><a href="#"><span class="glyphicon glyphicon-book"></span>
Bibliothèques</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
cutlery"></span> Restaurants</a></li>
  </ul>
</div>
```



Des en-têtes dans un menu déroulant

Alignement à droite

Par défaut, le menu est aligné sur le côté gauche du bouton. On peut utiliser la classe `pull-right` pour l'aligner sur le côté droit (voir figure suivante) :

Code : HTML

```
<div class="btn-group">
  <button class="btn btn-warning">Action</button>
  <button class="btn btn-warning dropdown-toggle" data-
toggle="dropdown"><span class="caret"></span></button>
  <ul class="dropdown-menu pull-right">
    <li><a href="#"><span class="glyphicon glyphicon-user"></span>
Dompteurs</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
picture"></span> Zoos</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
screenshot"></span> Chasseurs</a></li>
    <li class="divider"></li>
    <li><a href="#"><span class="glyphicon glyphicon-list-
alt"></span> Autres témoignages</a></li>
  </ul>
</div>
```



Menu déroulant aligné à droite

Désactiver une option

Pour rendre une option du menu inactive, il faut utiliser la classe `disabled` :

Code : HTML

```

<div class="btn-group">
  <button class="btn btn-danger">Action</button>
  <button class="btn btn-danger dropdown-toggle" data-
toggle="dropdown"><span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#"><span class="glyphicon glyphicon-book"></span>
Lire</a></li>
    <li class="disabled"><a href="#"><span class="glyphicon glyphicon-
print"></span> Imprimer</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-pencil"></span>
Ecrire</a></li>
  </ul>
</div>

```



Option désactivée dans un menu déroulant

Des sous-menus

La version 3 de Bootstrap, principalement orientée vers l'informatique nomade, n'a pas reconduit la possibilité de créer des sous-menus. Il est toutefois possible avec quelques styles de rétablir cette possibilité (astuce trouvée sur le site [bootply](#)). Prévoyez ces styles pour votre page :

Code : CSS

```

.dropdown-submenu{position:relative;}
.dropdown-submenu>.dropdown-menu{top:0;left:100%;margin-
top:-6px;margin-left:-1px;
-webkit-border-radius:0 6px 6px 6px;-moz-border-radius:0 6px 6px
6px;border-radius:0 6px 6px 6px;}
.dropdown-submenu:hover>.dropdown-menu{display:block;}
.dropdown-submenu>a:after{display:block;content:"
";float:right;width:0;height:0;
border-color:transparent;border-style:solid;border-width:5px 0 5px
5px;border-left-color:#cccccc;margin-top:5px;margin-right:-10px;}
.dropdown-submenu:hover>a:after{border-left-color:#ffffff;}
.dropdown-submenu.pull-left{float:none;}.dropdown-submenu.pull-
left>.dropdown-menu{left:-100%;margin-left:10px;
-webkit-border-radius:6px 0 6px 6px;-moz-border-radius:6px 0 6px
6px;border-radius:6px 0 6px 6px;}

```

Un exemple de mise en œuvre à la figure suivante :

Code : HTML

```

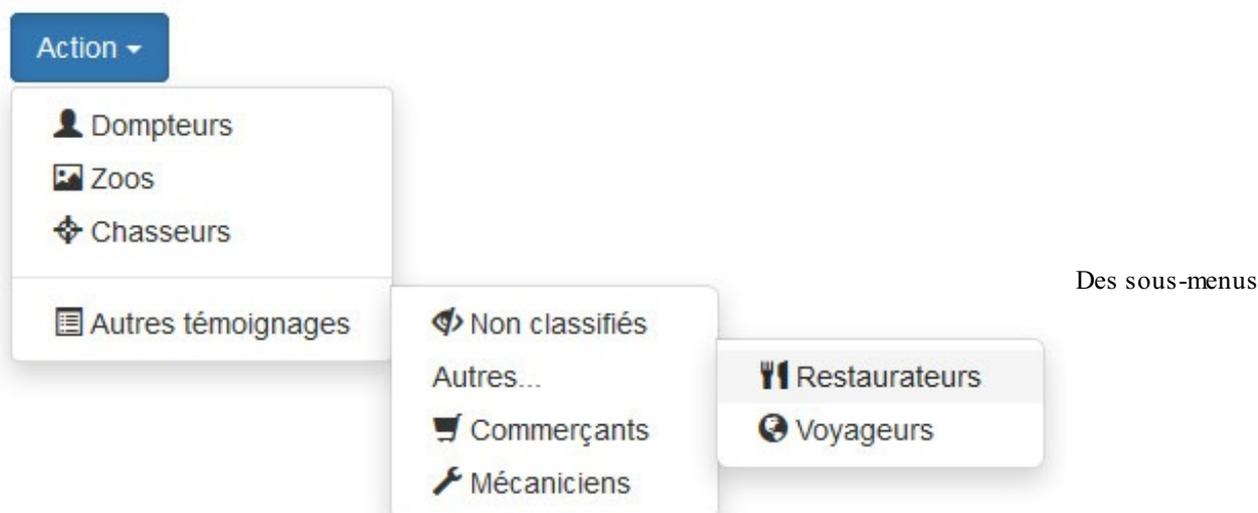
<div class="btn-group">
  <button class="btn btn-primary dropdown-toggle" data-
toggle="dropdown">Action <span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#"><span class="glyphicon glyphicon-user"></span>
Dompteurs</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-
picture"></span> Zoos</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-

```

```

screenshot"></span> Chasseurs</a></li>
  <li class="divider"></li>
  <li class="dropdown-submenu">
    <a tabindex="-1" href="#"><span class="glyphicon glyphicon-
list-alt"></span> Autres témoignages</a>
    <ul class="dropdown-menu">
      <li><a tabindex="-1" href="#"><span class="glyphicon
glyphicon-eye-close"></span> Non classifiés</a></li>
      <li class="dropdown-submenu">
        <a href="#">Autres...</a>
        <ul class="dropdown-menu">
          <li><a href="#"><span class=" glyphicon glyphicon-
cutlery"></span> Restaurateurs</a></li>
          <li><a href="#"><span class="glyphicon glyphicon-
globe"></span> Voyageurs</a></li>
        </ul>
      </li>
      <li><a href="#"><span class="glyphicon glyphicon-shopping-
cart"></span> Commerçants</a></li>
      <li><a href="#"><span class="glyphicon glyphicon-
wrench"></span> Mécaniciens</a></li>
    </ul>
  </li>
</ul>
</div>

```



Activation par Javascript

Déclenchement du menu

Tant qu'on veut une mise en œuvre du menu déroulant avec un clic, la propriété `data-toggle` est suffisante. Mais il peut se présenter des cas de figure particuliers. Imaginez que vous vouliez déclencher le menu déroulant non plus par un clic, mais par le passage du curseur de la souris au-dessus d'un élément particulier. Dans ce cas, vous devez utiliser Javascript pour faire fonctionner votre menu. Reprenons par exemple le cas d'un menu déclenché par un bouton, comme nous l'avons vu précédemment, mais en supprimant la propriété `data-toggle` pour éviter de déclencher automatiquement le plugin :

Code : HTML

```

<div class="btn-group">
  <button class="btn btn-info dropdown-toggle">Action</button>
  <ul class="dropdown-menu">
    <li><a href="#">Dompteurs</a></li>
    <li><a href="#">Zoos</a></li>
    <li><a href="#">Chasseurs</a></li>
  </ul>

```

```

    <li class="divider"></li>
    <li><a href="#">Autres témoignages</a></li>
  </ul>
</div>

```

En l'état le menu ne fonctionne plus. Maintenant nous voulons qu'il se déclenche au survol. Il faut ajouter un peu de Javascript pour gérer l'événement correspondant et utiliser la méthode `dropdown` du plugin :

Code : HTML

```

<script>
  $('<code>.dropdown-toggle</code>').mouseover(function() {
    $(this).dropdown('toggle');
  });
</script>

```

Maintenant, le menu s'ouvre au survol. Mais on rencontre un petit inconvénient : on n'arrive plus à le refermer ! Pourtant, normalement, un "toggle" est censé changer à chaque fois l'état. On va arranger cela. Le plugin ajoute la classe `open` dans la `<div>` qui contient la classe `btn-group` lorsque le menu est ouvert. Il faut donc s'arranger pour supprimer cette classe au survol si elle est présente :

Code : HTML

```

<script>
  $('<code>.dropdown-toggle</code>').mouseover(function() {
    if( $('<code>.btn-group</code>').hasClass('open') ) $('<code>.btn-group</code>').removeClass('open');
    else $(this).dropdown('toggle');
  });
</script>

```

La syntaxe que j'ai utilisée n'est pas la plus concise, mais elle a le mérite d'être lisible. Maintenant le menu se referme au survol s'il était ouvert. En revanche, il ne se referme pas lorsque l'on clique sur une option. Il faudrait donc prévoir de le faire lors du traitement de l'option sélectionnée.

Les événements

Vous disposez également de 4 événements pour ce composant :

Événement	Description
<code>show.bs.dropdown</code>	Se déclenche dès l'appel à la méthode <code>show</code>
<code>shown.bs.dropdown</code>	Se déclenche lorsque le menu devient visible
<code>hide.bs.dropdown</code>	Se déclenche dès l'appel à la méthode <code>hide</code>
<code>hidden.bs.dropdown</code>	Se déclenche lorsque le menu est masqué

L'utilité de ces événements ne saute pas spontanément aux yeux. Si vous en avez néanmoins besoin la syntaxe est celle-ci :

Code : JavaScript

```

$('<code>#mon_dropdown</code>').on('shown.bs.dropdown', function () {
  // Je fais un truc quand le menu est devenu visible
})

```

Fenêtre modale

Un premier exemple simple

La mise en place d'une fenêtre modale nécessite l'utilisation de quelques classes. Voici un premier exemple :

Code : HTML

```
<button data-toggle="modal" href="#infos" class="btn btn-
primary">Informations</button>
<div class="modal" id="infos">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-
dismiss="modal">x</button>
        <h4 class="modal-title">Plus d'informations</h4>
      </div>
      <div class="modal-body">
        Le Tigre (Panthera tigris) est un mammifère carnivore de la
famille des félidés...
      </div>
    </div>
  </div>
</div>
```

Le principe est simple : on crée un `<div>` englobant avec la classe `modal`, qui se charge de configurer et positionner la fenêtre. On crée un second `<div>` englobant avec la classe `modal-dialog`. On peut ensuite organiser la fenêtre avec une classe englobante `modal-content`, un en-tête grâce à la classe `modal-header` et un corps avec `modal-body`. On prévoit également une action pour fermer la fenêtre (sinon elle va toujours rester ouverte !) avec un bouton et la classe `close`, et la propriété `data-dismiss="modal"`. Il ne reste plus qu'à créer un bouton pour ouvrir la fenêtre en faisant correspondre bouton et fenêtre avec un identifiant, et en utilisant `data-toggle="modal"` pour que le plugin fonctionne. Au départ, on n'a donc que le bouton de la figure suivante à l'écran.



Bouton de commande

Et quand on clique, on obtient la fenêtre modale et le fond devient gris, comme à la figure suivante.



Fenêtre

modale sur fond gris

[Tester](#)

On a une petite croix pour fermer la fenêtre.

Un pied de page

On peut améliorer un peu l'esthétique en créant un pied de page avec la classe `modal-footer` et un bouton pour sortir, comme à la figure suivante.

Code : HTML

```

<button data-toggle="modal" href="#infos" class="btn btn-
primary">Informations</button>
<div class="modal" id="infos">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-
dismiss="modal">x</button>
        <h4 class="modal-title">Plus d'informations</h4>
      </div>
      <div class="modal-body">
        Le Tigre (Panthera tigris) est un mammifère carnivore de la
famille des félidés...
      </div>
      <div class="modal-footer">
        <button class="btn btn-info" data-dismiss="modal">Fermer</button>
      </div>
    </div>
  </div>
</div>

```



Pied de

page dans une fenêtre modale

Une transition

Il est possible également d'obtenir un effet de transition en ajoutant la classe `fade`, comme à la figure suivante :

Code : HTML

```
<div class="modal fade" id="infos">
```



Effet de transition pour une fenêtre modale

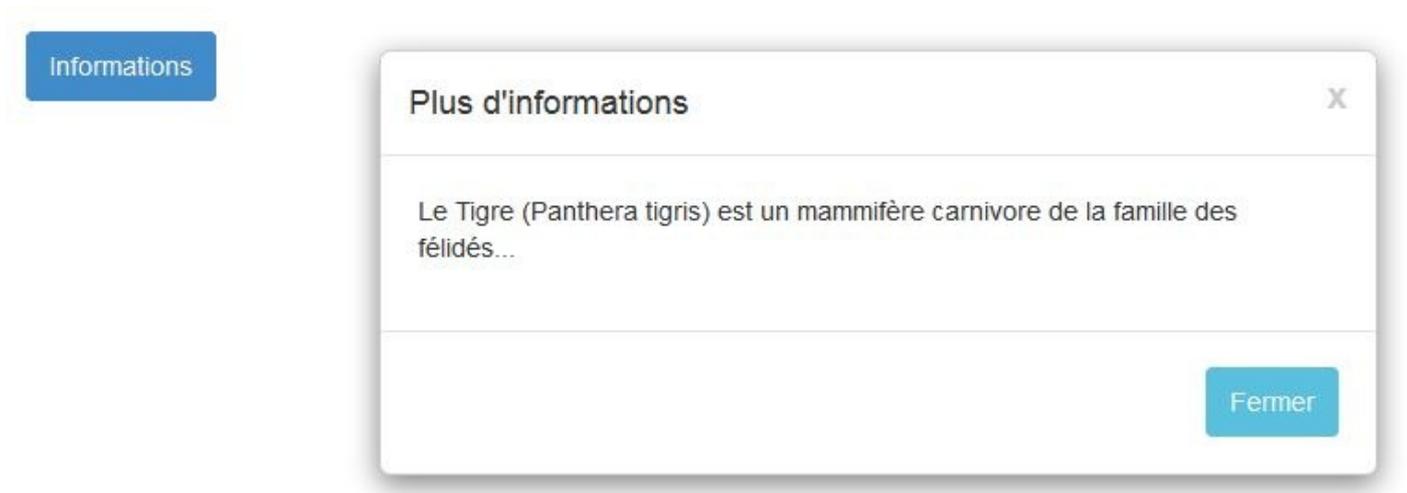
[Tester](#)

Garder le fond de l'écran

Si vous désirez conserver l'aspect normal de la fenêtre de fond, il faut renseigner l'option `backdrop` qui est vraie par défaut (voir figure suivante) :

Code : HTML

```
<button data-toggle="modal" data-backdrop="false" href="#infos"
class="btn btn-primary">Informations</button>
```



Fenêtre modale avec fond d'écran conservé

[Tester](#)

Injecter une page HTML

Il est possible d'injecter une page HTML dans le corps de la fenêtre modale, mais il faut que cette page ait été pensée pour cela. Vous en verrez un exemple à la figure suivante :

Code : HTML

```
<!DOCTYPE HTML>
```

```

<:DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      body { padding-top: 20px; }
    </style>
  </head>
  <body>
    <div class="container">
      <button data-toggle="modal" href="remote.html" data-target="#infos"
class="btn btn-primary">
        informations
      </button>
      <div class="modal fade" id="infos">
        <div class="modal-dialog">
          <div class="modal-content"></div>
        </div>
      </div>
    </div>
    <script src="assets/js/jquery.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>
  </body>
</html>

```

Le bouton comporte l'adresse de la page à charger avec l'attribut `href`. On se contente de prévoir le contenant avec la classe `modal-content`. Quand on clique sur le bouton, jQuery charge la page ciblée et l'injecte dans la `<div>` qui comporte la classe `modal-content`.



Cette mise en œuvre a changé avec la version 3.1 de Bootstrap. Auparavant on se contentait d'une `div` avec la classe `modal`.

Voici le code de la page à charger :

Code : HTML

```

<div class="modal-header">
  <button type="button" class="close" data-
dismiss="modal">&times;</button>
  <h1>Mon amour pour les tigres</h1>
</div>
<div class="modal-body">
  <p>
    Je suis passionné par les <strong>tigres</strong> depuis très
longtemps. Ce site a été construit en <em>
hommage à ces merveilleux félins...</em><br>
    Je fais partie de la <abbr title="Société des Amoureux des
Tigres">SAT</abbr> qui a pour but de faire
connaître ces splendides animaux.
  </p>
  <p>Voici ce qu'en dit le wikipedia :</p>
  <blockquote>
    Le Tigre (Panthera tigris) est un mammifère carnivore de la
famille des félidés (Felidae) du genre Panthera.
    Aisément reconnaissable à sa fourrure rousse rayée de noir, il est
le plus grand félin sauvage et l'un des plus
grands carnivores du monde. L'espèce est divisée en neuf sous-
espèces possédant des différences mineures en termes
de taille ou de comportement. Superprédateur, il chasse
principalement les cerfs et les sangliers, bien qu'il
puisse s'attaquer à des proies de taille plus importante comme les
buffles. Jusqu'au XIXe siècle, le Tigre était
réputé mangeur d'homme. La structure sociale des tigres en fait un

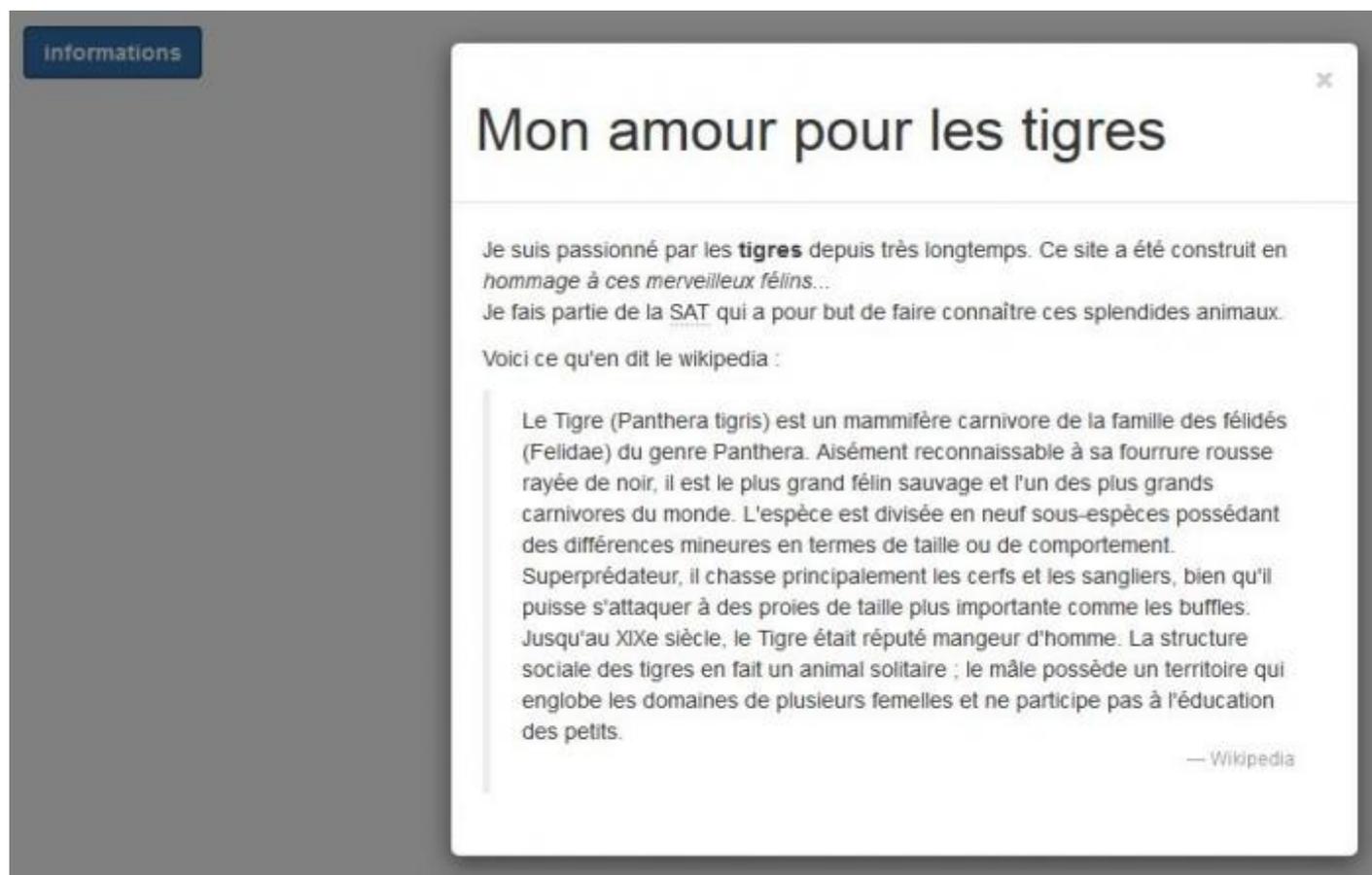
```

```

animal solitaire ; le mâle possède un territoire
qui englobe les domaines de plusieurs femelles et ne participe pas
à l'éducation des petits.<br>
<small class="pull-right">Wikipedia</small><br>
</blockquote>
</div>

```

Il faut prévoir les classes pour faire bien apparaître la fenêtre modale. Voici le résultat :



Une page HTML dans une fenêtre modale

[Tester](#)

Mon exemple tel qu'il est traité ne présente pas un grand intérêt, autant prévoir le code directement dans la page de base. En revanche, cela peut devenir intéressant dans le cas où plusieurs contenus sont disponibles. Vous en verrez un exemple à la figure suivante :

Code : HTML

```

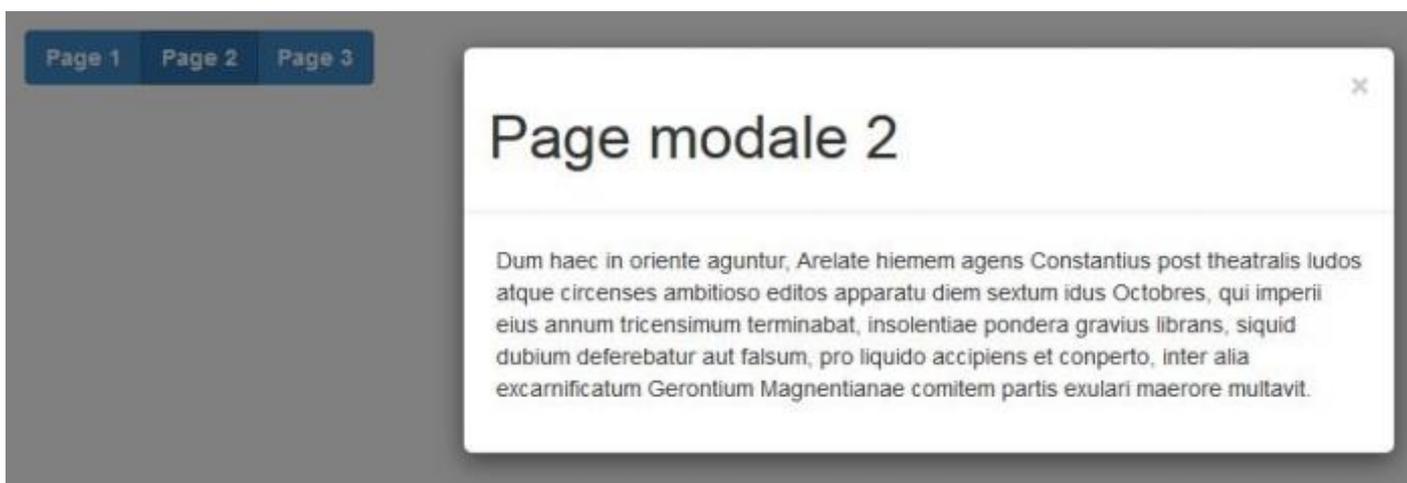
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <style type="text/css">
      body { padding-top: 20px; }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="btn-group">
        <button id="page1" type="button" class="btn btn-
primary">Page 1</button>
        <button id="page2" type="button" class="btn btn-

```

```

primary">Page 2</button>
  <button id="page3" type="button" class="btn btn-
primary">Page 3</button>
</div>
  <div class="modal fade" id="infos">
    <div class="modal-dialog">
      <div class="modal-content"></div>
    </div>
  </div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
  $("body").on("hidden.bs.modal", ".modal", function () {
    $(this).removeData("bs.modal");
  });
  $("#page1").click(function () {
    $("#infos").modal({ remote: "remote1.html" }, "show");
  });
  $("#page2").click(function () {
    $("#infos").modal({ remote: "remote2.html" }, "show");
  });
  $("#page3").click(function () {
    $("#infos").modal({ remote: "remote3.html" }, "show");
  });
</script>
</body>
</html>

```



Fenêtres modales commandées par des boutons

[Tester](#)

Notez l'utilisation de l'événement `hidden.bs.modal` qui survient à la fermeture de la fenêtre modale. Il est nécessaire de réinitialiser la fenêtre sinon on ouvre toujours la même...

Largeur de la fenêtre modale

La version 3.1 de Bootstrap a ajouté la possibilité de choisir la dimension d'une fenêtre modale avec 2 classes : `modal-lg` et `modal-sm`. La première permet d'obtenir une fenêtre large, alors que la seconde fait l'inverse en créant une fenêtre étroite. Ces classes doivent aller avec la classe `modal-dialog`. Vous en trouverez un exemple à la figure suivante :

Code : HTML

```

<div class="container">
  <button class="btn btn-primary" data-toggle="modal" data-
target="#f1">Grande fenêtre</button>
  <div class="modal fade" id="f1">

```

```

<div class="modal-dialog modal-lg">
  <div class="modal-content">
    <div class="modal-body">
      Je suis une grande fenêtre !
      <button type="button" class="close" data-
dismiss="modal">x</button>
    </div>
  </div>
</div>
</div>
</div>
<button class="btn btn-primary" data-toggle="modal" data-
target="#f2">Petite fenêtre</button>
<div class="modal fade" id="f2">
  <div class="modal-dialog modal-sm">
    <div class="modal-content">
      <div class="modal-body">
        Je suis une petite fenêtre !
        <button type="button" class="close" data-
dismiss="modal">x</button>
      </div>
    </div>
  </div>
</div>
</div>
</div>

```



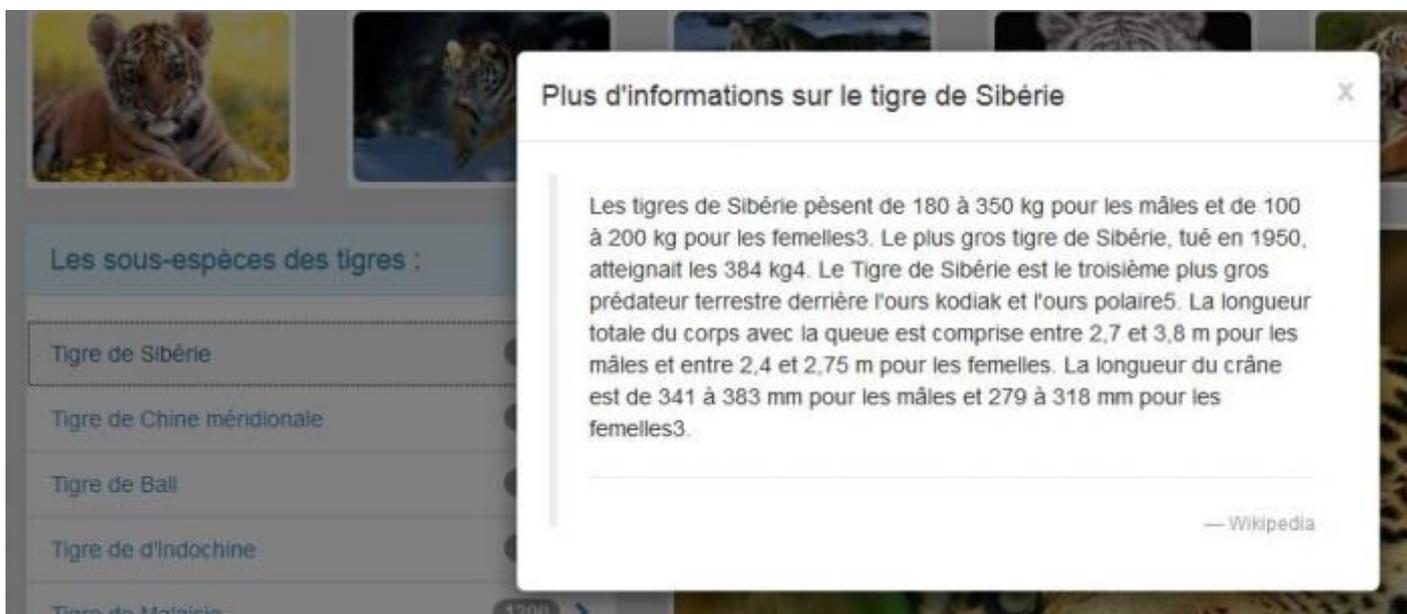
Petite fenêtre avec la classe modal-sm



Grande fenêtre avec la classe modal-lg

La page d'exemple et l'accessibilité

Une fenêtre modale a été ajoutée à la page d'exemple. Pour ne pas trop alourdir cette page, j'ai juste prévu une fenêtre qui s'ouvre quand on clique sur le lien "Tigre de Sibérie", comme à la figure suivante.



Fenêtre modale dans la page d'exemple

Tester

Voici le code correspondant :

Code : HTML

```
<a href="#infos" class="list-group-item" data-toggle="modal">
  Tigre de Sibérie
  <span class="badge">120</span>
</a>
<div class="modal fade" id="infos" role="dialog" aria-
labelledby="modalTitre" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"
aria-hidden="true">x</button>
        <h4 id="modalTitre" class="modal-title">Plus d'informations
sur le tigre de Sibérie</h4>
      </div>
      <div class="modal-body">
        <blockquote>
          Les tigres de Sibérie pèsent de 180 à 350 kg pour les
mâles et de 100 à 200 kg pour les femelles.
          Le plus gros tigre de Sibérie, tué en 1950, atteignait les 384 kg.
          Le Tigre de Sibérie est le troisième plus
          gros prédateur terrestre derrière l'ours kodiak et l'ours polaire.
          La longueur totale du corps avec la queue
          est comprise entre 2,7 et 3,8 m pour les mâles et entre 2,4 et 2,75
          m pour les femelles. La longueur du crâne
          est de 341 à 383 mm pour les mâles et 279 à 318 mm pour les
          femelles.<hr>
          <small class="pull-right">Wikipedia</small>
        </blockquote>
      </div>
    </div>
  </div>
</div>
```

J'ai intégré le code ARIA pour l'accessibilité. Le rôle a été défini comme `dialog` ([définition du W3C](#)), c'est ce qui est préconisé par les auteurs de Bootstrap. Si on lit la définition, on parle de fenêtre qui interrompt le flux normal et qui requiert une entrée de l'utilisateur (ce qui n'est pas le cas ici). J'avoue ne m'être jamais vraiment inquiété de cet aspect d'accessibilité et la lecture de la préconisation W3C est quelque peu laborieuse. Il me semble toutefois judicieux de prendre en compte les besoins des personnes présentant un déficit sensoriel. En parcourant les rôles, j'aurais plutôt eu tendance à utiliser `complementary` dans le cas présent.

La propriété `aria-labelledby` fait référence au titre de la fenêtre modale. Par exemple, un lecteur d'écran va lire le texte correspondant pour renseigner l'utilisateur sur le contenu de la fenêtre modale. Enfin la propriété `aria-hidden` indique de ne pas tenir compte de l'élément. Évidemment, ces ajouts d'accessibilité n'ont vraiment de sens que si toute la page est traitée de cette façon...

Activation avec Javascript

Déclenchement de la fenêtre modale

On a vu que l'on peut pratiquement tout faire avec des propriétés, mais vous avez aussi la possibilité d'utiliser Javascript pour utiliser ce plugin. La propriété `data-toggle="modal"` placée dans l'élément déclencheur active automatiquement le plugin. Reprenons un exemple vu ci-dessus en n'utilisant plus cette propriété :

Code : HTML

```
<button class="btn btn-primary">Informations</button>
```

```

<div class="modal" id="infos">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-
dismiss="modal">x</button>
        <h4 class="modal-title">Plus d'informations</h4>
      </div>
      <div class="modal-body">
        Le Tigre (Panthera tigris) est un mammifère carnivore de la
famille des félidés...
      </div>
      <div class="modal-footer">
        <button class="btn btn-info" data-
dismiss="modal">Fermer</button>
      </div>
    </div>
  </div>
</div>

```

Maintenant lorsqu'on clique sur le bouton, il ne se passe plus rien. Pour activer le plugin, il faut utiliser du Javascript :

Code : JavaScript

```

$('.btn').click(function() {
  $('.modal').modal('show');
});

```

On peut aussi modifier le comportement de la fenêtre avec des propriétés. Par exemple, la propriété booléenne `keyboard`, qui est normalement vraie, permet de fermer la fenêtre par action sur la touche d'échappement. On peut la renseigner en complétant notre exemple. Mais il faut aussi renseigner la propriété `show` qui est vraie par défaut, et ouvre la fenêtre dès le chargement de la page, ce que l'on ne désire évidemment pas. Voici le code correspondant :

Code : JavaScript

```

$('.modal').modal({
  keyboard: false,
  show: false
});

```

Les événements

Vous disposez également de 5 événements pour ce composant :

Événement	Description
<code>show.bs.modal</code>	Se déclenche dès l'appel à la méthode <code>show</code>
<code>shown.bs.modal</code>	Se déclenche lorsque la fenêtre devient visible
<code>hide.bs.modal</code>	Se déclenche dès l'appel à la méthode <code>hide</code>
<code>hidden.bs.modal</code>	Se déclenche lorsque la fenêtre est masquée
<code>loaded.bs.modal</code>	Se déclenche lorsque la fenêtre a chargé le contenu récupéré avec la propriété <code>remote</code>

Vous avez par exemple un formulaire sur la page modale et vous désirez que le premier contrôle soit actif, vous pouvez le réaliser facilement ainsi :

Code : JavaScript

```
$(".modal").on('shown.bs.modal', function() {
  $('input:first').focus();
});
```

Onglets

Un premier exemple simple

Le plugin **Tab** permet de créer facilement des onglets, comme à la figure suivante.

Code : HTML

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#accueil" data-
toggle="tab">Accueil</a></li>
  <li><a href="#livres" data-toggle="tab">Livres</a></li>
  <li><a href="#temoignages" data-
toggle="tab">Témoignages</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane active" id="accueil">Texte d'accueil</div>
  <div class="tab-pane" id="livres">Tous les livres</div>
  <div class="tab-pane" id="temoignages">Tous les
temoignages</div>
</div>
```



On utilise une liste et quelques classes, rien de bien compliqué. Les classes `nav` et `nav-tab` s'occupent de la mise en forme. Le plugin est activé par `data-toggle="tab"`. On pointe le bon contenu avec `href`.

Un autre aspect

Si cette apparence tabulaire ne vous convient pas, vous pouvez changer pour un autre aspect avec une modification minime, comme à la figure suivante.

Code : HTML

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



Transition progressive

Pour obtenir un effet progressif, il suffit d'ajouter la classe `fade` pour chacun des `tab-pane` :

Code : HTML

```

<ul class="nav nav-pills">
  <li class="active"><a href="#accueil" data-
toggle="tab">Accueil</a></li>
  <li><a href="#livres" data-toggle="tab">Livres</a></li>
  <li><a href="#temoignages" data-
toggle="tab">Témoignages</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane fade active in" id="accueil">Texte
d'accueil</div>
  <div class="tab-pane fade" id="livres">Tous les livres</div>
  <div class="tab-pane fade" id="temoignages">Tous les
témoignages</div>
</div>

```

Il faut ajouter aussi la classe `in` pour le `tab-pane` actif pour faire apparaître le texte (opacity à 1) au chargement de la page.

Empilage

Si ça ne vous convient encore pas (mais vous êtes difficile là 😊), vous pouvez empiler les liens (voir figure suivante).

Code : HTML

```

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

```



Onglets empilés

Justification

Il est aussi possible de faire en sorte que les onglets occupent tout l'espace disponible en se répartissant de façon homogène (voir figure suivante).

Code : HTML

```

<div class="col-lg-8">
<ul class="nav nav-pills nav-justified">
  <li class="active"><a href="#accueil" data-
toggle="tab">Accueil</a></li>
  <li><a href="#livres" data-toggle="tab">Livres</a></li>
  <li><a href="#temoignages" data-
toggle="tab">Témoignages</a></li>
</ul>
  <div class="tab-content">
    <div class="tab-pane active" id="accueil">Texte
d'accueil</div>
    <div class="tab-pane" id="livres">Tous les livres</div>
    <div class="tab-pane" id="temoignages">Tous les

```

```
témoignages</div>
</div>
</div>
```



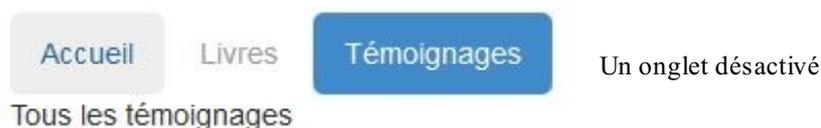
Onglets justifiés

Désactiver un lien

Vous pouvez marquer un lien comme inactif avec la classe `disabled`, comme à la figure suivante :

Code : HTML

```
<ul class="nav nav-pills">
  <li class="active"><a href="#accueil" data-
toggle="tab">Accueil</a></li>
  <li class="disabled"><a href="#livres" data-
toggle="tab">Livres</a></li>
  <li><a href="#temoignages" data-
toggle="tab">Témoignages</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane active" id="accueil">Texte d'accueil</div>
  <div class="tab-pane" id="livres">Tous les livres</div>
  <div class="tab-pane" id="temoignages">Tous les
témoignages</div>
</div>
```



Un onglet désactivé

Mais ce n'est pas parce qu'il apparaît comme inactif qu'il l'est vraiment ! Pour le rendre réellement inactif, il faut supprimer la propriété `data-toggle` :

Code : HTML

```
<li class="disabled"><a href="#livres">Livres</a></li>
```

Un menu déroulant dans un onglet

Pour avoir un menu déroulant sur un onglet, il faut utiliser le plugin `dropdown` que nous avons déjà vu (comme à la figure suivante) :

Code : HTML

```
<ul class="nav nav-tabs">
  <li><a href="#accueil" data-toggle="tab">Accueil</a></li>
  <li class="dropdown"><a class="dropdown-toggle" data-
toggle="dropdown" href="#">Livres <span class="caret"></span></a>
  <ul class="dropdown-menu">
```

```

    <li><a href="#policiers" data-toggle="tab">Policiers</a></li>
    <li><a href="#romans" data-toggle="tab">Romans</a></li>
    <li><a href="#contes" data-toggle="tab">Contes</a></li>
  </ul>
</li>
<li><a href="#temoignages" data-toggle="tab">Témoignages</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane active" id="accueil">Texte d'accueil</div>
  <div class="tab-pane" id="temoignages">Tous les
témoignages</div>
  <div class="tab-pane" id="policiers">Tous les livres
policiers</div>
  <div class="tab-pane" id="romans">Tous les romans</div>
  <div class="tab-pane" id="contes">Tous les contes</div>
</div>

```



Onglets latéraux

Vous pouvez adapter la disposition et mettre par exemple les onglets empilés à gauche, comme à la figure suivante :

Code : HTML

```

<ul class="nav nav-pills nav-stacked pull-left">
  <li><a href="#accueil" data-toggle="tab">Accueil</a></li>
  <li class="active"><a href="#livres" data-
toggle="tab">Livres</a></li>
  <li><a href="#temoignages" data-toggle="tab">Témoignages</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane" id="accueil">Texte d'accueil</div>
  <div class="tab-pane" id="temoignages">Tous les
témoignages</div>
  <div class="tab-pane active" id="livres">Tous les livres</div>
</div>

```



Ou à droite (voir figure suivante) :

Code : HTML

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

Texte d'accueil

Accueil

Livres

Onglets à droite

Témoignages

La page d'exemple

Notre page d'exemple est maintenant enrichie d'onglets dans la partie inférieure, comme à la figure suivante.

Onglets dans la page d'exemple

Tester

Voici la partie du code concernée :

Code : HTML

```
<section class="col-sm-8">
  <ul class="nav nav-pills">
    <li class="active"><a href="#contact" data-
toggle="tab">Contact</a></li>
    <li><a href="#informations" data-toggle="tab">Informations
pratiques</a></li>
  </ul>
  <div class="tab-content">
    <div class="tab-pane active fade in" id="contact">
      <form class="well">
        <h4><span class="label label-default">Si vous voulez me
laisser un message</span></h4>
        <h4>Comment m'avez-vous trouvé ?</h4>
        <fieldset>
          <label for="ami" class="radio">
            <input type="radio" name="origine" value="ami" id="ami">
Par un ami
```

```

        </label>
        <label for="web" class="radio">
          <input type="radio" name="origine" value="web" id="web">
            Sur le web
        </label>
        <label for="hasard" class="radio">
          <input type="radio" name="origine" value="hasard"
id="hasard">
            Par hasard
        </label>
        <label for="autre" class="radio">
          <input type="radio" name="origine" value="autre"
id="autre">
            Autre...
        </label>
        <label for="textarea">Votre message :</label>
        <textarea id="textarea" rows="4" class="form-
control"></textarea>
        <p class="help-block">Vous pouvez agrandir la
fenêtre</p>
        <button class="btn btn-primary" type="submit"><span
class="glyphicon glyphicon-ok-sign"></span> Envoyer</button>
      </fieldset>
    </form>
  </div>
  <div class="tab-pane fade" id="informations">
    <div class="well">
      <h2>Informations pratiques</h2>
      <p>Si vous êtes intéressés par la protection des tigres et
si vous désirez prendre part aux actions menées... blablabla...</p>
    </div>
  </div>
</div>
<div class="tab-content">
</section>

```

Utilisation de Javascript

Activation des onglets

Nous avons ci-dessus activé le plugin avec la propriété `data-toggle="tab"`. Il est aussi possible d'utiliser le plugin directement avec du Javascript. Voici un des exemples sans la propriété renseignée :

Code : HTML

```

<ul class="nav nav-pills nav-stacked pull-right">
  <li><a href="#accueil" data-toggle="tab">Accueil</a></li>
  <li class="active"><a href="#livres" data-
toggle="tab">Livres</a></li>
  <li><a href="#temoignages" data-toggle="tab">Témoignages</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane" id="accueil">Texte d'accueil</div>
  <div class="tab-pane" id="temoignages">Tous les
témoignages</div>
  <div class="tab-pane active" id="livres">Tous les livres</div>
</div>

```

Les onglets sont devenus inactifs. Pour que ça fonctionne il faut ajouter un peu de Javascript :

Code : JavaScript

```

$('a').click(function (e) {
  e.preventDefault();

```

```
$(this).tab('show');
});
```

Tous les onglets doivent être activés, c'est pour cette raison que j'ai choisi pour ma page le sélecteur 'a' qui est sans ambiguïté.

Les événements

Vous disposez également de 2 événements pour ce composant :

Événement	Description
show.bs.tab	Se déclenche dès l'appel à la méthode show
shown.bs.tab	Se déclenche lorsque l'onglet devient visible

D'autre part il est possible de connaître l'onglet actif avec `event.target` et l'onglet précédemment sélectionné avec `event.relatedTarget`. On va utiliser tout cela pour afficher le nom de l'onglet actuel et du précédent à chaque changement. On va donc ajouter le code HTML pour accueillir l'information :

Code : HTML

```
<ul class="nav nav-pills">
  <li class="active"><a href="#accueil">Accueil</a></li>
  <li><a href="#livres">Livres</a></li>
  <li><a href="#temoignages">Témoignages</a></li>
</ul>
<div class="tab-content">
  <div class="tab-pane active" id="accueil">Texte d'accueil</div>
  <div class="tab-pane" id="livres">Tous les livres</div>
  <div class="tab-pane" id="temoignages">Tous les
témoignages</div>
</div>
<hr>
<p id='actif'><strong>Onglet actif </strong>: <span></span></p>
<p id='precedent'><strong>Onglet précédent </strong>:
<span></span></p>
```

Il ne reste plus qu'à prévoir le code Javascript pour déclencher le plugin et mettre en place l'écoute de l'événement et agir en conséquence (voir figure suivant) :

Code : JavaScript

```
$(function() {
  $('a')
  .click(function (e) {
    e.preventDefault();
    $(this).tab('show');
  })
  .on('shown.bs.tab', function (e) {
    $('#actif span').html($(e.target).text());
    $('#precedent span').html($(e.relatedTarget).text());
  });
});
```



Boutons

Bouton bascule

On a parfois besoin de boutons à 2 états stabilisés : repos et appuyé. Le plugin permet de réaliser cela facilement. Il suffit de créer le bouton en prévoyant `data-toggle="button"` (comme à la figure suivante) :

Code : HTML

```
<a class="btn btn-success" data-toggle="button">Simple Bascule</a>
```



Boutons effet "checkbox"

On peut grouper des boutons avec la classe `btn-group` et les faire fonctionner comme des checkbox avec `data-toggle="buttons"`. Les boutons restent indépendants dans leur fonctionnement mais ont l'avantage d'être groupés. Pour réaliser cela, il faut utiliser des contrôles `input` de type checkbox (voir figure suivante) :

Code : HTML

```
<div class="btn-group" data-toggle="buttons">  
  <label class="btn btn-success">  
    <input type="checkbox">Un  
  </label>  
  <label class="btn btn-success">  
    <input type="checkbox">Deux  
  </label>  
  <label class="btn btn-success">  
    <input type="checkbox">Trois  
  </label>  
</div>
```



Boutons effet "radio"

On peut grouper des boutons avec la classe `btn-group` et les faire fonctionner comme des boutons radio avec `data-toggle="buttons"`. Les boutons sont maintenant liés dans leur fonctionnement, un seul peut être enfoncé. Pour réaliser cela, il faut utiliser des contrôles `input` de type radio (voir figure suivante) :

Code : HTML

```
<div class="btn-group" data-toggle="buttons">
  <label class="btn btn-success">
    <input type="radio">Un
  </label>
  <label class="btn btn-success">
    <input type="radio">Deux
  </label>
  <label class="btn btn-success">
    <input type="radio">Trois
  </label>
</div>
```



Boutons pour attente de processus

Parfois on clique sur un bouton pour réaliser un processus qui demande un certain temps, comme par exemple un envoi de fichier. Le plugin prévoit cette possibilité. Voici un premier exemple avec une temporisation artificielle juste pour marquer l'effet.

D'abord on crée le bouton :

Code : HTML

```
<button type="button" class="btn btn-primary" data-loading-
text="Chargement...">Cliquez !</button>
```

Ensuite, on crée le code pour gérer l'effet. La méthode `loading` provoque le changement du texte du bouton avec ce qui est indiqué dans `data-loading-text` et la méthode `reset` ramène le bouton à son état initial (voir figure suivante):

Code : JavaScript

```
$(function () {
  $('a').click(function () {
    with($ (this)) {
      button('loading');
      setTimeout(function () {
        button('reset');
      }, 4000);
    }
  })
});
```



Tester

Maintenant voyons un cas d'utilisation un peu plus réaliste que notre minuterie. Nous avons une image lourde à charger, et nous voulons que le bouton change d'état le temps de chargement de l'image. Voilà le bouton et la balise prête à recevoir l'image :

Code : HTML

```
<a class="btn btn-primary" data-loading-text="Chargement en cours...">Chargez l'image !</a>
<img id="mon_image">
```

Et voici le code Javascript pour gérer l'effet :

Code : JavaScript

```
$(function () {
  $('a').click(function () {
    with($(this)) {
      if(hasClass('btn-primary')) {
        removeClass('btn-primary').addClass('btn-danger').button('loading');
        var image = new Image();
        image.onload = function () {
          removeClass('btn-danger').addClass('btn-primary').button('reset');
          $("#mon_image").attr({ src:"images/legumes.jpg" });
        };
        image.src = "images/legumes.jpg";
      }
    }
  })
});
```

[Tester](#)

Avec cet exemple, vous ne verrez certainement l'effet qu'une fois dans votre navigateur, étant donné qu'ensuite l'image sera en cache.

La page d'exemple

La page d'exemple a été agrémentée de boutons groupés de type "radio" pour choisir une vidéo (voir la figure suivante).



Boutons avec effet option dans la page d'exemple

Tester

Code : HTML

```
<div class="btn-group" data-toggle="buttons">
  <label class="btn btn-primary">
    <input type="radio">Vidéo 1
  </label>
  <label class="btn btn-primary">
    <input type="radio">Vidéo 2
  </label>
  <label class="btn btn-primary">
    <input type="radio">Vidéo 3
  </label>
</div>
```

Carrousel

La page d'exemple

La page d'exemple comporte un carrousel (voir la figure suivante). Pour information, ce terme prend un *R* en anglais et deux en français.



Carrousel dans la page

d'exemple

Tester

Un premier exemple simple

Ce plugin permet de créer facilement un carrousel. La mise en place nécessite l'utilisation de quelques classes. Voici la structure de base HTML :

Code : HTML

```
<div class="carousel slide">
  <div class="carousel-inner">
    <div class="item active"> ... </div>
    <div class="item"> ... </div>
    <div class="item"> ... </div>
  </div>
</div>
```

Voici un premier exemple simple :

Code : HTML

```
<div class="container">
  <div class="col-lg-offset-2 col-lg-8">
    <div class="carousel slide">
      <div class="carousel-inner">
        <div class="item active"> </div>
        <div class="item"> </div>
        <div class="item"> </div>
      </div>
    </div>
  </div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
  $(function () {
    $('.carousel').carousel();
  });
</script>
```

Pour chaque item on place une balise `` pour référencer une image. On utilise la classe `active` pour désigner l'image qui doit apparaître en premier. J'ai utilisé une classe `col-lg-8` pour calibrer le carrousel et une classe `col-lg-offset-2` pour le centrer. Les 3 images sont aux mêmes dimensions. Il nous faut aussi initialiser le carrousel avec jQuery. Par défaut les images défilent au rythme de 5s (voir figure suivante) :



Un carrousel simple

[Tester](#)

Des titres pour les images

Il est possible de faire apparaître un titre pour chaque image dans la partie inférieure avec la classe `carousel-caption` (voir figure suivante) :

Code : HTML

```
<div class="col-lg-offset-2 col-lg-8">
  <div class="carousel slide">
    <div class="carousel-inner">
      <div class="item active"> 
```

```

<h1 class="carousel-caption">Une présentation</h1>
  </div>
  <div class="item"> 
<h1 class="carousel-caption">Une autre présentation</h1>
  </div>
  <div class="item"> 
<h1 class="carousel-caption">Et encore une autre !</h1>
  </div>
</div>
</div>
</div>
</div>

```



Des titres dans les images

Tester



Notez que le titre vient au-dessus de l'image et qu'il faut donc adapter le style aux couleurs présentes sur l'image pour que le titre soit lisible.

Un indicateur de l'image affichée

Il est possible de faire apparaître un indicateur pour savoir quelle image est affichée et aussi pour accéder directement à une image particulière. Il faut identifier le carrousel et le pointer avec l'attribut `data-target`. L'attribut `data-slide-to` permet d'identifier la diapositive concernée. On peut aussi sélectionner la diapositive à afficher en cliquant sur l'indicateur (voir figure suivante) :

Code : HTML

```

<div id="carousel" class="carousel slide">
  <ol class="carousel-indicators">
    <li data-target="#carousel" data-slide-to="0" class="active"></li>
    <li data-target="#carousel" data-slide-to="1"></li>
    <li data-target="#carousel" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="item active"> </div>
    <div class="item"> </div>
    <div class="item"> </div>
  </div>
</div>

```



Un indicateur de l'image affichée

Tester



Notez que l'indicateur vient au-dessus de l'image et qu'il faut donc adapter le style aux couleurs présentes sur l'image pour qu'il soit lisible.

Des boutons de défilement

On peut aussi afficher des boutons à droite et à gauche pour permettre un défilement manuel des images. Il faut identifier le carrousel pour le référencer dans les liens avec href (voir figure suivante) :

Code : HTML

```
<div class="carousel-inner">
  <div class="item active"> </div>
  <div class="item"> </div>
  <div class="item"> </div>
</div>
<a class="left carousel-control" href="#carousel" data-slide="prev">
<span class="icon-prev"></span>
</a>
<a class="right carousel-control" href="#carousel" data-
slide="next">
<span class="icon-next"></span>
</a>
```



Des boutons pour le défilement

des images

Tester



Notez que les chevrons viennent au-dessus de l'image et qu'il faut donc adapter le style aux couleurs présentes sur l'image pour qu'ils soient visibles.

Améliorer la présentation

On peut améliorer la présentation du diaporama avec du style, comme à la figure suivante :

Code : CSS

```
.carousel-inner {
  border-radius: 5px 5px 5px 5px;
  border: 5px solid white;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.3);
}
```



Amélioration de la présentation

Mais on peut aussi utiliser une classe de Bootstrap pour obtenir un effet sympathique sans ajouter de style (voir figure suivante) :

Code : HTML

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



Utilisation de la classe thumbnail

pour la présentation

Les options du carrousel

Délai de défilement

Code : JavaScript

```
$(function () {
  $(''.carousel').carousel({ interval: 2000 });
});
```

Testez !

Cycle et pause

On peut avec jQuery commander le démarrage du cycle ou la pause (voir figure suivante) :

Code : HTML

```
<div class="container">
  <div class="col-lg-2">
    <div class="btn-group" data-toggle="buttons">
      <label class="btn btn-primary active" id="cycle">
        <input type="radio" name="options">Cycle
      </label>
      <label class="btn btn-primary" id="pause">
        <input type="radio" name="options">Pause
      </label>
    </div>
  </div>
  <div class="col-lg-8">
    <div id="carousel" class="carousel slide">
      <div class="carousel-inner">
        <div class="item active"> </div>
```

```

        <div class="item"> </div>
        <div class="item"> </div>
    </div>
</div>
</div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
    $(function () {
        $('.carousel').carousel({ interval: 2000 });
        $('#cycle').click(function () {
            $('.carousel').carousel('cycle');
        });
        $('#pause').click(function () {
            $('.carousel').carousel('pause');
        });
    });
</script>

```



Commande du carrousel avec des

boutons

[Tester](#)

Les deux boutons commandent le diaporama. Une bonne occasion pour revoir comment configurer des boutons de type "radio" avec Bootstrap 🤪.

Autres commandes

Voici maintenant un exemple plus complet avec une palette de boutons de commande (voir figure suivante) :

Code : HTML

```

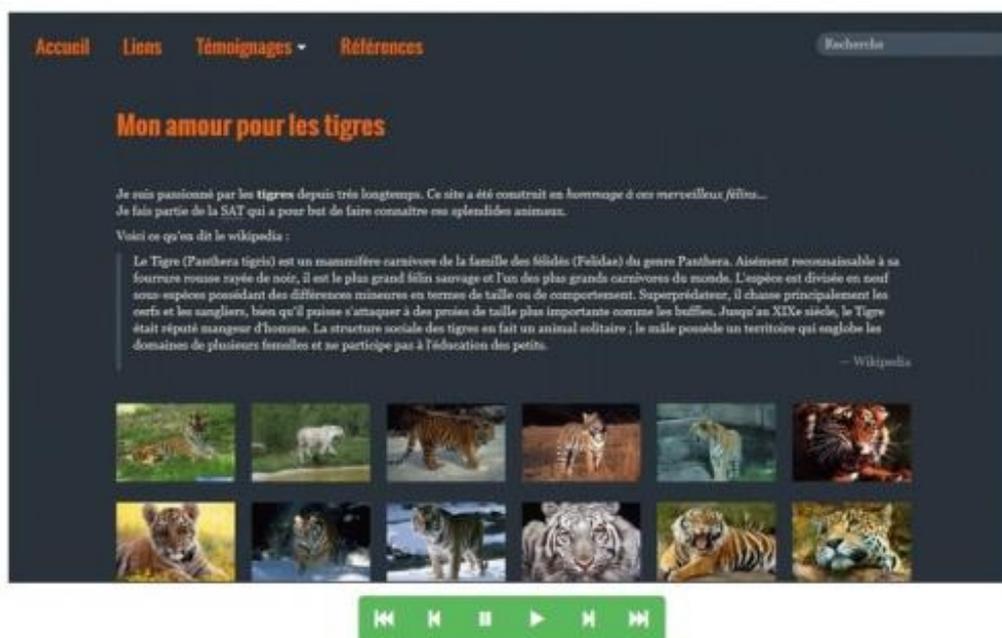
<div class="container">
    <div class="row">
        <div class="col-lg-offset-2 col-lg-8">
            <div id="carousel" class="carousel slide">
                <div class="carousel-inner">
                    <div class="item active"> </div>
                    <div class="item"> </div>
                    <div class="item"> </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="row">

```

```

    <div class="col-lg-12">
      <div class="btn-group" data-toggle="buttons">
        <label class="btn btn-success" id="first">
          <input type="radio" name="options"><span class="glyphicon
glyphicon-fast-backward"></span>
        </label>
        <label class="btn btn-success" id="previous">
          <input type="radio" name="options"><span class="glyphicon
glyphicon-step-backward"></span>
        </label>
        <label class="btn btn-success" id="pause">
          <input type="radio" name="options"><span class="glyphicon
glyphicon-pause"></span>
        </label>
        <label class="btn btn-success active" id="play">
          <input type="radio" name="options"><span class="glyphicon
glyphicon-play"></span>
        </label>
        <label class="btn btn-success" id="next">
          <input type="radio" name="options"><span class="glyphicon
glyphicon-step-forward"></span>
        </label>
        <label class="btn btn-success" id="last">
          <input type="radio" name="options"><span class="glyphicon
glyphicon-fast-forward"></span>
        </label>
      </div>
    </div>
  </div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
  $(function () {
    $('.carousel').carousel({ interval: 2000 });
    $('#first').click(function () { $('.carousel').carousel(0); });
    $('#previous').click(function () {
    $('.carousel').carousel('prev'); });
    $('#pause').click(function () { $('.carousel').carousel('pause');
    });
    $('#play').click(function () { $('.carousel').carousel('cycle');
    });
    $('#next').click(function () { $('.carousel').carousel('next');
    });
    $('#last').click(function () { $('.carousel').carousel(2); });
  });
</script>

```



Une barre de boutons pour la

commande du carousel

Tester

Les événements

Vous disposez de 2 événements pour ce composant :

Événement	Description
<code>slide.bs.carousel</code>	Se déclenche dès l'appel à la méthode <code>slide</code>
<code>slid.bs.carousel</code>	Se déclenche lorsque la translation est terminée

Complétons l'exemple précédent en prévoyant l'affichage de l'index de la diapositive en cours ainsi que du nombre total de diapositives (voir figure suivante). Voici le code avec ce complément :

Code : HTML

```
<div class="container">
  <div class="row">
    <div class="col-lg-offset-2 col-lg-8">
      <div id="carousel" class="carousel slide">
        <div class="carousel-inner">
          <div class="item active"> </div>
          <div class="item"> </div>
          <div class="item"> </div>
        </div>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-lg-12">
      <div class="btn-group" data-toggle="buttons">
        <label class="btn btn-success" id="first">
          <input type="radio" name="options"><span class="glyphicon
glyphicon-fast-backward"></span>
        </label>
        <label class="btn btn-success" id="previous">
          <input type="radio" name="options"><span class="glyphicon
```

```

glyphicon-step-backward"></span>
  </label>
  <label class="btn btn-success" id="pause">
    <input type="radio" name="options"><span class="glyphicon
glyphicon-pause"></span>
  </label>
  <label class="btn btn-success active" id="play">
    <input type="radio" name="options"><span class="glyphicon
glyphicon-play"></span>
  </label>
  <label class="btn btn-success" id="next">
    <input type="radio" name="options"><span class="glyphicon
glyphicon-step-forward"></span>
  </label>
  <label class="btn btn-success" id="last">
    <input type="radio" name="options"><span class="glyphicon
glyphicon-fast-forward"></span>
  </label>
</div>
<p><span class="label label-info"></span></p>
</div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
  $(function () {
    $('.carousel').carousel({ interval: 2000 });
    $('#first').click(function () { $('.carousel').carousel(0); });
    $('#previous').click(function () {
    $('.carousel').carousel('prev'); });
    $('#pause').click(function () { $('.carousel').carousel('pause');
    });
    $('#play').click(function () { $('.carousel').carousel('cycle');
    });
    $('#next').click(function () { $('.carousel').carousel('next');
    });
    $('#last').click(function () { $('.carousel').carousel(2); });
    $('.carousel').on('slid.bs.carousel', function () {
    var total = $('.item').length;
    var current = $('.item.active').index() + 1;
    $('.label').text(current + ' / ' + total);
    });
  });
</script>

```



Affichage de l'index et du nombre total de diapositives

[Tester](#)

Pour rester dans l'utilisation de Bootstrap, j'ai prévu l'affichage dans un badge. Pour déterminer le nombre de diapositives, on compte le nombre de fois où on trouve la classe `item`. Pour la diapositive en cours, on recherche la classe `active`.

Info-bulles et Popover

La page d'exemple

La page d'exemple est agrémenté d'infos-bulles (voir figure suivante).

Mon amour pour les tigres

Je suis passionné par les **tigres** depuis très longtemps. Ce site a été construit en *hommage à ces merveilleux félins...*
Je fais partie de la [SAT](#) qui a pour but de faire connaître ces splendides animaux.

Voici ce qu'en dit le wikipedia :

Classe de vertébrés

Le Tigre (*Panthera tigris*) est un [mammifère](#) carnivore de la famille des félidés (*Felidae*) du genre *Panthera*. Aisément plus grand félin sauvage et l'un des plus grands [carnivores](#) du monde. L'espèce est divisée en neuf sous-espèces par comportement. Superprédateur, il chasse principalement les cerfs et les sangliers, bien qu'il puisse s'attaquer à des proies plus petites. Au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle et les femelles ne participent pas à l'éducation des petits.

Des infos-bulles dans la page d'exemple

[Tester](#)

Ces deux plugins (Info-bulles et Popover) partagent une bibliothèque et ont un fonctionnement analogue, nous allons donc les traiter ensemble.

Info-bulle

Un exemple simple

Le but est d'obtenir de jolies info-bulles, comme à la figure suivante.

Classe de vertébrés

ra tigris) est un [mammifère](#) carnivore de la famille des félidés (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands [carnivores](#) du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement.

Voici le code correspondant à ce premier exemple :

Code : HTML

```
Le Tigre (Panthera tigris) est un mammifère carnivore de la famille des félidés (Felidae) du genre Panthera. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands carnivores du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement.
```

On utilise des balises `<a>` avec l'attribut `data-toggle` égal à `tooltip`. Le contenu de l'info-bulle se place dans l'attribut `title`. Si vous testez ce code il ne se passera rien parce qu'il faut initialiser le plugin avec jQuery :

Code : JavaScript

```
$(function () {
  $('a').tooltip();
});
```

Il suffit d'utiliser la fonction `tooltip` avec le bon sélecteur, ici on a choisit la balise `<a>` parce qu'on veut une info-bulle sur tous les liens de la page. Dans un contexte plus complexe il faudrait évidemment affiner la sélection, mais ce n'est pas l'objet de ce cours.

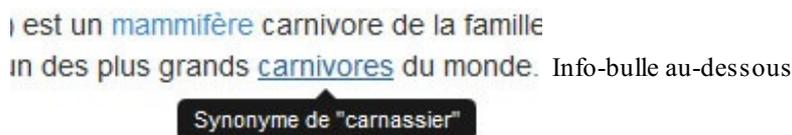
Positionnement de l'info-bulle

On peut changer le positionnement de l'info-bulle qui par défaut se situe juste au-dessus avec l'option `placement`. Les possibilités sont `top` (au-dessus, c'est l'option par défaut), `bottom` (au-dessous), `left` (à gauche) et `right` (à droite). Voici un exemple d'utilisation avec le même code HTML :

Code : JavaScript

```
$(function () {
  $('a:first-child').tooltip({ placement: 'left' });
  $('a:last-child').tooltip({ placement: 'bottom' });
});
```

Ici on utilise l'option `placement` en la définissant à `left` pour le premier lien et `bottom` pour le second (voir figure suivante).



Déclenchement de l'info-bulle

Par défaut l'info-bulle est déclenchée lorsque le curseur de la souris se déplace sur l'élément concerné ou si celui-ci a le focus. On peut modifier ce comportement avec l'option `trigger`. Les possibilités sont `click`, `hover`, `focus` et `manual`. Voici un exemple d'utilisation avec le même code HTML :

Code : JavaScript

```
$(function () {
  $('a:first-child').tooltip({ trigger: 'click' });
  $('a:last-child').tooltip({ placement: 'bottom', trigger: 'hover click' });
});
```

Ici on utilise l'option `trigger` en la définissant à `click` pour le premier lien et `hover` et `click` pour le second (voir figure suivante).



[Tester](#)

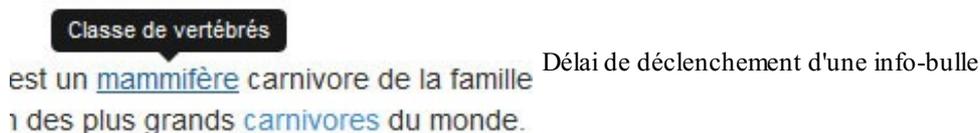
Délai de déclenchement de l'info-bulle

Par défaut l'info-bulle s'affiche immédiatement et disparaît avec la même célérité. On peut modifier ce comportement avec l'option `delay`. On peut soit définir un délai identique pour l'apparition et la disparition, soit différencier les deux. Voici un exemple d'utilisation avec le même code HTML :

Code : JavaScript

```
$(function () {  
  $('a:first-child').tooltip({ delay: 400 });  
  $('a:last-child').tooltip({ delay: { show: 400, hide: 200 } });  
});
```

Ici on utilise l'option `delay` en la définissant à 400 ms pour le premier lien pour l'affichage et la disparition. On différencie avec 400 ms pour l'affichage et 200 ms pour la disparition pour le second (voir figure suivante).

[Tester](#)

Popover

Un exemple simple

Un popover est une grosse info-bulle. Le fonctionnement est identique à ce que nous avons vu précédemment. Voici un premier exemple de mise en œuvre au niveau HTML :

Code : HTML

```
<a href="#" id="pop" class="btn btn-info" data-toggle="popover"  
data-content="C'est tout simple à faire !" title="Test du Popover">  
Cliquez sur moi pour le popover</a>
```

On utilise des balises `<a>` avec l'attribut `data-toggle` égal à `popover`. Le contenu se place dans l'attribut `data-content` et le titre dans l'attribut `title`. Si vous testez ce code, il ne se passera rien parce qu'il faut initialiser le plugin avec jQuery :

Code : JavaScript

```
$(function () {  
  $("#pop").popover();  
});
```

Il suffit d'utiliser la fonction `popover` avec le bon sélecteur, ici on a choisi tout simplement un `id`. Vous verrez le résultat à la figure suivante.



Positionnement

Comme pour l'info-bulle on peut modifier le positionnement avec l'option `placement` avec les mêmes possibilités `top` (au-dessus, c'est l'option par défaut), `bottom` (au-dessous), `left` (à gauche) et `right` (à droite) (voir figure suivante) :

Code : JavaScript

```
$("#pop").popover({placement: 'left'});
```



Délai

On peut aussi définir un délai d'apparition et de disparition :

Code : JavaScript

```
$("#pop").popover({delay: { show: 800, hide: 300 }});
```

Tester !

Déclenchement

Le comportement par défaut est un déclenchement au clic sur l'élément, mais on peut facilement changer ce comportement :

Code : JavaScript

```
$("#pop").popover({placement: 'bottom', trigger: 'hover'});
```

Ici on a utilisé l'option `trigger` pour avoir un déclenchement au passage (`hover`) sur l'élément (les autres possibilités sont `click`, `focus` et `manual`). On montre aussi qu'on peut cumuler plusieurs options puisqu'on a aussi défini le positionnement en bas avec `bottom`.

Tester

Avec l'option `manual` on peut définir une action quelconque pour déclencher le popover que l'on veut. Regardez cet exemple au niveau HTML :

Code : HTML

```
<div class="well">
  
  <a href="#" id="pop" class="btn btn-info pull-right" >Cliquez sur
moi plusieurs fois pour le popover</a>
</div>
```

Cette fois, on a défini le popover au niveau d'une image et on se propose de le déclencher à partir d'un bouton. Voici le code jQuery correspondant :

Code : JavaScript

```
$("#img").popover({trigger:'manual'});
$("#pop").click(function() {
  $('#img').popover('toggle');
});
```

On initialise le popover sur l'image en déclenchement manuel : `$("#img").popover({trigger:'manual'});`

On utilise l'événement `click` du bouton pour lancer la fonction `popover` avec `toggle` comme paramètre :

```
$('#img').popover('toggle');
```

Vous verrez le résultat à la figure suivante.



niveau d'une image

Tester

Du HTML dans le popover

Par défaut, vous ne pouvez mettre que du simple texte dans le popover. Toutefois, vous pouvez mettre l'option `html` à `true` pour changer ce fonctionnement (voir figure suivante) :

Code : HTML

```
<a href="#" id="pop" class="btn btn-info" data-toggle="popover"
data-html="true" data-content="<em>C'est tout simple à faire !</em>"
title="<h3>Test du popover</h3>">
Cliquez sur moi pour le popover</a>
```



Effet accordéon

Un exemple simple

Le plugin **accordéon** (Collapse) est bien adapté au composant **panel**, voici un premier exemple (voir figure suivante) :

Code : HTML

```

<div class="panel panel-default col-lg-6">
  <h4> Les plugins de Bootstrap</h4>
  <div class="panel-heading">
    <a class="accordion-toggle" href="#item1" data-
toggle="collapse"> Accordéon </a>
  </div>
  <div id="item1" class="panel-collapse collapse in">
    <div class="panel-body"> Ce plugin permet de créer des effets
"accordéon" totalement paramétrables</div>
  </div>
  <div class="panel-heading">
    <a class="accordion-toggle" href="#item2" data-
toggle="collapse"> Fenêtre modale </a>
  </div>
  <div id="item2" class="panel-collapse collapse">
    <div class="panel-body"> Ce plugin permet de créer des fenêtres
modales élégantes avec une grande simplicité. </div>
  </div>
  <div class="panel-heading">
    <a class="accordion-toggle" href="#item3" data-
toggle="collapse"> Carousel </a>
  </div>
  <div id="item3" class="panel-collapse collapse">
    <div class="panel-body"> Ce plugin permet de faire défiler des
images ou des vidéo, ou tout autre élément média avec une mise
en forme esthétique </div>
  </div>

```

Les plugins de Bootstrap

Accordéon

Ce plugin permet de créer des effets "accordéon" totalement paramétrables

Effet accordéon

Fenêtre modale

Carousel

[Tester](#)

Il faut distinguer deux éléments :

- Le **déclencheur** sur lequel on clique : ici c'est un lien avec une balise `<a>`. Il doit comporter la classe **accordion-toggle**, comporter la propriété **data-toggle="collapse"** et évidemment la cible avec **href**
- La **zone** à masquer ou à démasquer : elle est englobée avec un `<div>` équipé de la classe **collapse**

Le lien entre le déclencheur et la zone se fait par l'identifiant de la zone. La classe **in** permet de déterminer si une zone est affichée ou pas. Dans mon exemple, j'ai mis cette classe pour la première zone, celle-ci s'affiche donc dès le chargement. Si vous testez le composant, vous verrez qu'il se contente d'affecter cette classe. J'ai un peu triché avec l'utilisation du composant **panel**...

Des bordures pour bien délimiter les éléments

Nous allons voir maintenant une utilisation plus complète du composant **panel** (voir figure suivante):

Code : HTML

```

<div class="panel-group col-lg-6">
  <h3> Les plugins de Bootstrap</h3>
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item1" data-
toggle="collapse"> Accordéon </a>
      </h3>
    </div>
    <div id="item1" class="panel-collapse collapse in">
      <div class="panel-body"> Ce plugin permet de créer des
effets "accordéon" totalement paramétrables</div>
    </div>
  </div>
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item2" data-
toggle="collapse"> Fenêtre modale </a>
      </h3>
    </div>
    <div id="item2" class="panel-collapse collapse">
      <div class="panel-body"> Ce plugin permet de créer des
fenêtres modales élégantes avec une grande simplicité. </div>
    </div>
  </div>
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item3" data-
toggle="collapse"> Carousel </a>
      </h3>
    </div>
    <div id="item3" class="panel-collapse collapse">
      <div class="panel-body"> Ce plugin permet de faire défiler
des images ou des vidéo, ou tout autre élément média avec une mise
en forme esthétique </div>
    </div>
  </div>
</div>

```

Les plugins de Bootstrap

Accordéon

Ce plugin permet de créer des effets "accordéon" totalement paramétrables

Des bordures pour

Fenêtre modale

Carousel

distinguer les blocs

Maintenant les blocs sont bien distincts...

Afficher une seule zone

En général, lorsqu'on crée un effet accordéon, on veut que lorsqu'on affiche une zone, les autres s'effacent automatiquement. Voilà comment procéder (résultat à la figure suivante) :

Code : HTML

```
<div id="monaccordeon" class="panel-group col-lg-6">
  <h3> Les plugins de Bootstrap</h3>
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item1" data-
parent="#monaccordeon" data-toggle="collapse"> Accordéon </a>
      </h3>
    </div>
    <div id="item1" class="panel-collapse collapse in">
      <div class="panel-body"> Ce plugin permet de créer des
effets "accordéon" totalement paramétrables</div>
    </div>
  </div>
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item2" data-
parent="#monaccordeon" data-toggle="collapse"> Fenêtre modale </a>
      </h3>
    </div>
    <div id="item2" class="panel-collapse collapse">
      <div class="panel-body"> Ce plugin permet de créer des
fenêtres modales élégantes avec une grande simplicité. </div>
    </div>
  </div>
  <div class="panel panel-default">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item3" data-
parent="#monaccordeon" data-toggle="collapse"> Carrousel </a>
      </h3>
    </div>
    <div id="item3" class="panel-collapse collapse">
      <div class="panel-body"> Ce plugin permet de faire défiler
des images ou des vidéo, ou tout autre élément média avec une mise
en forme esthétique </div>
    </div>
  </div>
</div>
```

Les plugins de Bootstrap

Accordéon

Fenêtre modale

Ce plugin permet de créer des fenêtres modales élégantes avec une grande simplicité.

Carousel

Affichage d'une seule

zone à la fois

Il a fallu ajouter un identifiant (**#monaccordeon**) à la balise englobante et faire une référence dans les déclencheurs avec **data-parent="#monaccordeon"**.

De jolies en-têtes

Ça fonctionne bien mais c'est un peu triste sans couleur, comme on utilise le composant panel, c'est facile à arranger (voir figure suivante) :

Code : HTML

```
<div id="monaccordeon" class="panel-group col-lg-6">
  <h3> Les plugins de Bootstrap</h3>
  <div class="panel panel-info">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item1" data-
parent="#monaccordeon" data-toggle="collapse"> Accordéon </a>
      </h3>
    </div>
    <div id="item1" class="panel-collapse collapse in">
      <div class="panel-body"> Ce plugin permet de créer des
effets "accordéon" totalement paramétrables</div>
    </div>
  </div>
  <div class="panel panel-info">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item2" data-
parent="#monaccordeon" data-toggle="collapse"> Fenêtre modale </a>
      </h3>
    </div>
    <div id="item2" class="panel-collapse collapse">
      <div class="panel-body"> Ce plugin permet de créer des
fenêtres modales élégantes avec une grande simplicité. </div>
    </div>
  </div>
  <div class="panel panel-info">
    <div class="panel-heading">
      <h3 class="panel-title">
        <a class="accordion-toggle" href="#item3" data-
parent="#monaccordeon" data-toggle="collapse"> Carrousel </a>
      </h3>
    </div>
    <div id="item3" class="panel-collapse collapse">
      <div class="panel-body"> Ce plugin permet de faire défiler
```

```

des images ou des vidéo, ou tout autre élément média avec une mise
en forme esthétique </div>
</div>
</div>

```

Les plugins de Bootstrap

Accordéon

Fenêtre modale

Carousel

Ce plugin permet de faire défiler des images ou des vidéo, ou tout autre élément média avec une mise en forme esthétique

Des en-têtes

esthétiques

Effet accordéon sur une liste groupée

L'effet accordéon n'est pas limité au composant panel. On peut par exemple l'utiliser pour une liste groupée. Voici un exemple de mise en œuvre (voir aussi figure suivante) :

Code : HTML

```

<div class="col-sm-3 col-md-3">
  <div class="panel panel-info">
    <div class="panel-heading">
      <h4 class="panel-title">
        <span id="icone" class="glyphicon glyphicon-folder-
close"></span> <a data-toggle="collapse" href="#menu">Contenu</a>
      </h4>
    </div>
    <ul id="menu" class="list-group collapse in">
      <li class="list-group-item"><span class="glyphicon glyphicon-
pencil text-primary"></span>
<a href="http://fb.com/moinakbarali"> Articles</a></li>
      <li class="list-group-item"><span class="glyphicon glyphicon-
tag text-primary"></span>
<a href="http://fb.com/moinakbarali"> Nouvelles</a></li>
      <li class="list-group-item"><span class="glyphicon glyphicon-
file text-primary"></span>
<a href="http://fb.com/moinakbarali"> Lettre
d'information</a></li>
      <li class="list-group-item"><span class="glyphicon glyphicon-
comment text-success"></span>
<a class="text-success"
href="http://fb.com/moinakbarali">Commentaires</a><span
class="badge">28</span></li>
    </ul>
  </div>
</div>

```



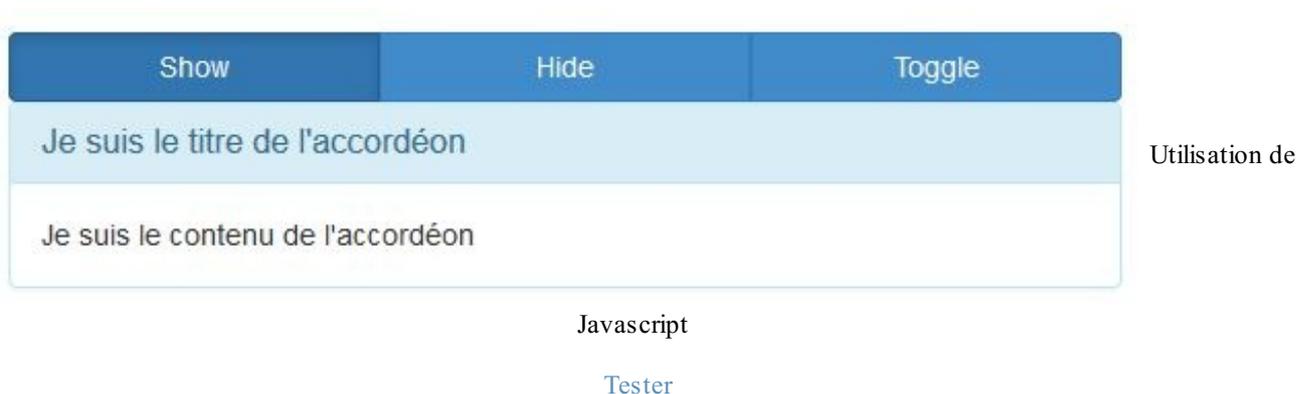
Effet accordéon sur une liste groupée

Commande en Javascript

On peut activer ce plugin avec Javascript et lancer des commandes sur n'importe quel élément. Voici un exemple (résultat à la figure suivante) :

Code : HTML

```
<div class="container">
  <div class="col-lg-6">
    <div class="btn-group btn-group-justified" data-
toggle="buttons">
      <label class="btn btn-primary active" id="show">
        <input type="radio" name="options">Show
      </label>
      <label class="btn btn-primary" id="hide">
        <input type="radio" name="options">Hide
      </label>
      <label class="btn btn-primary" id="toggle">
        <input type="radio" name="options">Toggle
      </label>
    </div>
    <div class="panel panel-info">
      <div class="panel-heading">
        <h3 class="panel-title"> Je suis le titre de
l'accordéon</h3>
      </div>
      <div class="panel-body"> Je suis le contenu de
l'accordéon</div>
    </div>
  </div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script type="text/javascript">
  $(function() {
    $('.panel').collapse();
    $('#show, #hide, #toggle').click(function() {
      $('.panel').collapse($(this).attr('id'));
    });
  });
</script>
```



L'effet accordéon est ici appliqué sur la `<div>` qui contient la classe `panel`. Le plugin est activé avec la méthode `collapse`. Ensuite une action sur l'un des trois boutons lance la commande correspondante : `show`, `hide` ou `toggle`. Pour rendre le code plus concis, j'ai fait en sorte que l'identifiant des boutons corresponde au nom d'une commande.

Utilisation des événements

Il y a plusieurs événements utilisables pour ce composant :

Événement	Description
<code>show.bs.collapse</code>	Se déclenche dès l'appel à la méthode <code>show</code>
<code>shown.bs.collapse</code>	Se déclenche lorsque la zone devient visible
<code>hide.bs.collapse</code>	Se déclenche dès l'appel à la méthode <code>hide</code>
<code>hidden.bs.collapse</code>	Se déclenche lorsque la zone est masquée

Voici un exemple d'utilisation de l'événement `shown.bs.collapse` (voir aussi à la figure suivante) :

Code : HTML

```
<div class="container">
  <div id="monaccordeon" class="panel-group col-lg-6">
    <h3> Les plugins de Bootstrap</h3>
    <div class="panel panel-info">
      <div class="panel-heading">
        <h3 class="panel-title">
          <a class="accordion-toggle" href="#item1" data-
parent="#monaccordeon" data-toggle="collapse"> Accordéon </a>
        </h3>
      </div>
      <div id="item1" class="panel-collapse collapse in">
        <div class="panel-body"> Ce plugin permet de créer des
effets "accordéon" totalement paramétrables</div>
      </div>
    </div>
    <div class="panel panel-info">
      <div class="panel-heading">
        <h3 class="panel-title">
          <a class="accordion-toggle" href="#item2" data-
parent="#monaccordeon" data-toggle="collapse"> Fenêtre modale </a>
        </h3>
      </div>
      <div id="item2" class="panel-collapse collapse">
        <div class="panel-body"> Ce plugin permet de créer des
fenêtres modales élégantes avec une grande simplicité. </div>
      </div>
    </div>
    <div class="panel panel-info">
      <div class="panel-heading">
        <h3 class="panel-title">
          <a class="accordion-toggle" href="#item3" data-
```

```

parent="#monaccordéon" data-toggle="collapse"> Carrousel </a>
  </h3>
</div>
  <div id="item3" class="panel-collapse collapse">
    <div class="panel-body"> Ce plugin permet de faire défiler
des images ou des vidéo, ou tout autre élément média avec une mise
en forme esthétique </div>
  </div>
</div>
<br>
<div id="affichage"><span class="label label-
warning">"Accordéon" a été affiché !</span></div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
$(function () {
  $("#item1").on("shown.bs.collapse", function () {
    $("#affichage").html('<span class="label label-
warning">"Accordéon" a été affiché !</span>');
  })
  $("#item2").on("shown.bs.collapse", function () {
    $("#affichage").html('<span class="label label-warning">"Fenêtre
modale" a été affiché !</span>');
  })
  $("#item3").on("shown.bs.collapse", function () {
    $("#affichage").html('<span class="label label-
warning">"Carrousel" a été affiché !</span>');
  })
});
</script>

```

Les plugins de Bootstrap

Accordéon

Fenêtre modale

Carousel

Utilisation des

Ce plugin permet de faire défiler des images ou des vidéo, ou tout autre élément média avec une mise en forme esthétique

"Carousel" a été affiché !

événements

[Tester](#)

Voyons à présent un second exemple, en reprenant l'effet accordéon sur une liste groupée vu plus haut. Il serait très esthétique de modifier l'icône en fonction de la visibilité de la liste est visible. Il suffit de modifier légèrement le code en identifiant la liste et en ajoutant un peu de Javascript (résultat à la figure suivante) :

Code : HTML

```

<div class="container">
  <div class="row">

```

```

<div class="col-sm-3 col-md-3">
  <div class="panel panel-info">
    <div class="panel-heading">
      <h4 class="panel-title">
        <span id="icone" class="glyphicon glyphicon-folder-
open"></span> <a data-toggle="collapse" href="#menu">Contenu</a>
      </h4>
    </div>
    <ul id="menu" class="list-group collapse in">
      <li class="list-group-item"><span class="glyphicon
glyphicon-pencil text-primary"></span>
<a href="http://fb.com/moinakbarali"> Articles</a></li>
      <li class="list-group-item"><span class="glyphicon
glyphicon-tag text-primary"></span>
<a href="http://fb.com/moinakbarali"> Nouvelles</a></li>
      <li class="list-group-item"><span class="glyphicon
glyphicon-file text-primary"></span>
<a href="http://fb.com/moinakbarali"> Lettre d'information</a></li>
      <li class="list-group-item"><span class="glyphicon
glyphicon-comment text-success"></span>
<a class="text-success"
href="http://fb.com/moinakbarali">Commentaires</a><span
class="badge">28</span></li>
    </ul>
  </div>
</div>
</div>
</div>
</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
<script>
$(function () {
  $('#menu').on('shown.bs.collapse', function () {
    $('#icone').removeClass('glyphicon-folder-
close').addClass('glyphicon-folder-open');
  })
  $('#menu').on('hidden.bs.collapse', function () {
    $('#icone').removeClass('glyphicon-folder-
open').addClass('glyphicon-folder-close');
  })
});
</script>

```



Les deux états

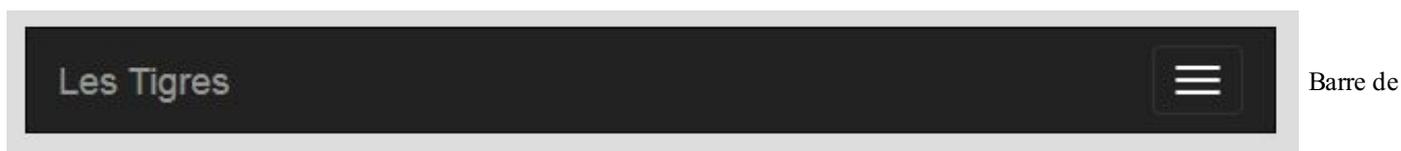
de la liste avec changement de l'icône

Tester

Une barre de navigation rétractable

Une utilisation du composant accordéon consiste à rendre une barre de navigation rétractable selon la dimension de l'affichage,

en d'autres termes de la rendre **responsive**. Nous l'avons fait pour la page d'exemple (voir la figure suivante).



navigation rétractable dans la page d'exemple

[Tester](#)

Voici la partie du code concernée :

Code : HTML

```
<nav class="navbar navbar-inverse">
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
  </button>
  <a class="navbar-brand" href="#">Les Tigres</a>
  </div>
  <div class="collapse navbar-collapse">
    <ul class="nav navbar-nav">
      <li><a href="#">Accueil</a> </li>
      <li class="dropdown">
        <a class="dropdown-toggle" data-toggle="dropdown"
href="#">Témoignages <b class="caret"></b></a>
        <ul class="dropdown-menu">
          <li><a href="#">Dompteurs</a></li>
          <li><a href="#">Zoos</a></li>
          <li><a href="#">Chasseurs</a></li>
          <li class="divider"></li>
          <li><a href="#">Autres témoignages</a></li>
        </ul>
      </li>
      <li><a href="#">Liens</a> </li>
      <li><a href="#">Références</a> </li>
    </ul>
    <form class="navbar-form navbar-right">
      <div class="input-group">
        <input type="text" style="width:150px" class="input-sm form-
control" placeholder="Recherche">
        <span class="input-group-btn">
          <button type="submit" class="btn btn-primary btn-
sm"><span class="glyphicon glyphicon-eye-open"></span>
Chercher</button>
        </span>
      </div>
    </form>
  </div>
</nav>
```

On ajoute un bouton chargé de commander l'effet accordéon aux lignes 3 à 7. Ensuite on englobe tout ce qui doit subir l'effet avec les classes **collapse** et **navbar-collapse**. Par défaut le point de transition (c'est-à-dire de déclenchement de l'effet) se produit lorsque l'affichage est inférieur à 768 pixels. Il est possible de changer ce point de transition en modifiant le fichier **LESS** de **Bootstrap**. Nous verrons dans un prochain chapitre comment justement modifier **Bootstrap** pour l'adapter à nos besoins.

Le Scrollspy

Mise en page avec le Scrollspy

Ce plugin permet de relier automatiquement des éléments de navigation avec des zones HTML en utilisant un défilement. Comme un exemple est souvent bien plus efficace qu'une explication, surtout pour ce genre de fonctionnement, voici un style de mise en page de plus en plus utilisé sur les sites (voir aussi figure suivante) :

Code : HTML

```

<!DOCTYPE HTML>
<html>
<head>
  <meta charset="utf-8">
  <link href="assets/css/bootstrap.css" rel="stylesheet">
</head>
<body data-spy="scroll" data-target=".navbar">

<div class="container-fluid">
  <nav class="navbar navbar-default navbar-fixed-top">
    <div class="navbar-header">
      <button type="button" data-target=".navbar-collapse" data-
toggle="collapse" class="navbar-toggle">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a href="#" class="navbar-brand">La boutique en délire</a>
    </div>
    <div class="collapse navbar-collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#accueil">Accueil</a></li>
        <li><a href="#produits">Produits</a></li>
        <li><a href="#nouvelles">Nouvelles</a></li>
        <li><a href="#localisation">Localisation</a></li>
      </ul>
    </div>
  </nav>

  <div class="jumbotron" id="accueil">
    <div class="container">
      <h1 class="well">Bienvenue dans notre espace !</h1>
      <p>"Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem
quia voluptas sit aspernatur aut odit aut fugit, sed quia
consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet,
consectetur, adipisci velit, sed quia non numquam eius modi
tempora incidunt ut labore et dolore magnam aliquam quaerat
voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem
ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi
consequatur? Quis autem vel eum iure reprehenderit qui in ea
voluptate velit esse quam nihil molestiae consequatur, vel illum
qui dolorem eum fugiat quo voluptas nulla pariatur?"
      </p>
    </div>
  </div>

  <div class="jumbotron" id="produits">
    <div class="container">
      <h1 class="well">Nos produits attractif !</h1>
      <p>"Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem
quia voluptas sit aspernatur aut odit aut fugit, sed quia
consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet,
consectetur, adipisci velit, sed quia non numquam eius modi

```

```

tempora incidunt ut labore et dolore magnam aliquam quaerat
voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem
ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi
consequatur? Quis autem vel eum iure reprehenderit qui in ea
voluptate velit esse quam nihil molestiae consequatur, vel illum
qui dolorem eum fugiat quo voluptas nulla pariatur?"
    </p>
    <hr>
    <div class="row">
      <div class="well">
        <div class="row">
          <div class="col-sm-4">
            <button type="button" class="btn btn-primary btn-lg
btn-block">
              <span class="glyphicon glyphicon-
wrench"></span><br>Outillage
            </button>
          </div>
          <div class="col-sm-4">
            <button type="button" class="btn btn-primary btn-lg
btn-block">
              <span class="glyphicon glyphicon-
fire"></span><br>Incendie
            </button>
          </div>
          <div class="col-sm-4">
            <button type="button" class="btn btn-primary btn-lg
btn-block">
              <span class="glyphicon glyphicon-
home"></span><br>Habitation
            </button>
          </div>
        </div>
      </div>
    </div>
    <hr>
  </div>

  <div class="jumbotron" id="nouvelles">
    <div class="container">
      <h1 class="well">Les nouvelles de nos activités !</h1>
      <p>"Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa
    quae ab illo inventore veritatis et quasi architecto beatae vitae
dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit
aspernatur aut odit aut fugit, sed quia consequuntur magni dolores
eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est,
qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit,
sed quia non numquam eius modi tempora incidunt ut labore et dolore
magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis
nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut
aliquid ex ea commodi consequatur? Quis autem vel eum iure
reprehenderit qui in ea voluptate velit esse quam nihil molestiae
consequatur,
    vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?"
      </p>
      <hr>
      <div class="row">
        <div class="well">
          <div class="row">
            <div class="col-sm-4">
              <button type="button" class="btn btn-info btn-lg btn-
block">
                <span class="glyphicon glyphicon-
wrench"></span><br>Outillage
              </button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        <button type="button" class="btn btn-info btn-lg btn-
block">
        <span class="glyphicon glyphicon-
fire"></span><br>Incendie
        </button>
    </div>
    <div class="col-sm-4">
        <button type="button" class="btn btn-info btn-lg btn-
block">
        <span class="glyphicon glyphicon-
home"></span><br>Habitation
        </button>
    </div>
</div>
</div>
</div>
</div>
<hr>
</div>
</div>

<div class="jumbotron" id="localisation">
    <div class="container">
        <h1 class="well">L'emplacement de nos boutiques !</h1>
        <p>"Sed ut perspiciatis unde omnis iste natus error sit
voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem
quia voluptas sit aspernatur aut odit aut fugit, sed quia
consequuntur magni dolores eos qui ratione voluptatem sequi
nesciunt.
Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet,
consectetur, adipisci velit, sed quia non numquam eius modi
tempora incidunt ut labore et dolore magnam aliquam quaerat
voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem
ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi
consequatur? Quis autem vel eum iure reprehenderit qui in ea
voluptate velit esse quam nihil molestiae consequatur, vel illum
qui dolorem eum fugiat quo voluptas nulla pariatur?"
        </p>
        <hr>
        <div class="row">
            <div class="well">
                <div class="row">
                    <div class="col-sm-4">
                        <button type="button" class="btn btn-warning btn-lg
btn-block">
                            <span class="glyphicon glyphicon-
wrench"></span><br>Outillage
                            </button>
                        </div>
                    <div class="col-sm-4">
                        <button type="button" class="btn btn-warning btn-lg
btn-block">
                            <span class="glyphicon glyphicon-
fire"></span><br>Incendie
                            </button>
                        </div>
                    <div class="col-sm-4">
                        <button type="button" class="btn btn-warning btn-lg
btn-block">
                            <span class="glyphicon glyphicon-
home"></span><br>Habitation
                            </button>
                        </div>
                    </div>
                </div>
            </div>
        </div>
        <hr>
    </div>
</div>

```

```

</div>
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>
</html>

```



Mise en page avec le Scrollspy

[Tester](#)

Le principe est simple. Il faut mettre en œuvre le plugin avec l'attribut `data-spy="scroll"` et on pointe la navigation avec `data-target` :

Code : HTML

```
<body data-spy="scroll" data-target=".navbar">
```

Ensuite on crée la navigation qui pointe des zones à faire apparaître sans ajout d'autres classes, le fonctionnement est automatique.

Un mouvement fluide

L'exemple fonctionne correctement, mais on peut regretter la brutalité du mouvement. La zone demandée s'affiche spontanément et on ne comprend pas vraiment qu'il se produit un défilement. Il serait plus élégant d'avoir un mouvement fluide pour passer d'une zone à une autre. Ce n'est pas prévu au niveau du plugin, mais avec quelques lignes de Javascript on peut réaliser cela :

Code : JavaScript

```

$(function () {
  $('li>a').click(function () {
    var pos = $($this).attr('href').offset().top;
    $('body, html').animate({scrollTop:pos}, 1000);
  });
});

```

[Tester](#)

Activation par Javascript

On a vu dans l'exemple que le plugin peut être simplement mis en œuvre avec un attribut "data". Il est aussi possible de le faire en Javascript. Pour tester cette possibilité, il faut retirer les attributs de la balise `body` et ajouter cette ligne de Javascript :

Code : JavaScript

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

[Tester](#)

Vous disposez aussi de l'événement `activate.bs.scrollspy` qui est déclenché lorsqu'une nouvelle zone est affichée. L'utilité de cet événement ne saute pas aux yeux, mais si vous en avez éventuellement besoin la syntaxe est la suivante :

Code : JavaScript

```
$('#monScrollspy').on('activate.bs.scrollspy', function () {  
    // Ici je fais quelque chose...  
})
```

En résumé :

- Bootstrap propose une collection de plugins jQuery faciles à mettre en œuvre.
- Les menus déroulants permettent l'affichage de liens textuels, auxquels on peut adjoindre des icônes et des en-têtes. Ils peuvent s'intégrer à une barre de navigation ou un bouton. On peut réaliser des sous-menus en ajoutant quelques styles au framework.
- Les fenêtres modales permettent de faire apparaître des informations. Elles peuvent comporter un titre, un contenu et un bas de page. On peut même y injecter une page HTML.
- Les onglets permettent sur un même espace d'afficher au choix plusieurs informations avec un simple clic. Les onglets sont faciles à organiser : horizontaux, empilés, à droite ou à gauche, simple liens ou boutons. On peut aussi intégrer un menu déroulant dans un onglet.
- On peut créer des boutons "bascule", ou avec effet "radio" ou "checkbox", ou pour attente de fin d'un processus.
- Le carrousel permet de faire défiler des images avec possibilité d'ajouter une légende par image, des boutons de défilement et de régler la vitesse de défilement.
- Les info-bulles permettent de faire apparaître un petit texte informatif en survolant un élément, les pop-overs également mais autorisent un contenu plus riche.
- L'effet accordéon permet de faire apparaître ou disparaître une zone de la page. Il est possible d'ajouter des en-têtes et d'intégrer des listes groupées. On l'utilise en particulier pour rendre une barre de navigation rétractable.
- Le Scrollspy permet de relier automatiquement des éléments de navigation avec des zones HTML en utilisant un défilement. Il est de plus en plus utilisé sur les sites web.

Configurer Bootstrap

Utiliser directement ce que nous offre Bootstrap c'est bien, mais le modifier pour le rendre exactement adapté à nos besoins c'est encore mieux. Alors, si vous voulez bidouiller Bootstrap, ce chapitre est fait pour vous !

Configuration simplifiée

Une page d'exemple

Supposons que nous voulions créer un site avec une certaine présentation. Quelque chose de simple, parce que je veux juste vous montrer le principe de la configuration. Voici le code HTML de base :

Code : HTML

```

<!DOCTYPE HTML>
<html>

  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.min.css" rel="stylesheet">
  </head>

  <body>
    <div class="container">
      <header class="row col-sm-12">
        <div class="page-header">
          <h1>Mon beau site</h1>
        </div>
      </header>
      <div class="row col-sm-12">
        <nav class="col-sm-2">
          <ul class="nav nav-pills nav-stacked">
            <li class="active"> <a href="#"> <span class="glyphicon
glyphicon-home"></span> Accueil </a> </li>
            <li> <a href="#"> <span class="glyphicon glyphicon-
book"></span> Livres </a> </li>
            <li> <a href="#"> <span class="glyphicon glyphicon-
facetime-video"></span> Films </a> </li>
            <li> <a href="#"> <span class="glyphicon glyphicon-
headphones"></span> Musiques </a> </li>
          </ul>
        </nav>
        <section class="col-sm-10">
          <div class="panel panel-default">
            <div class="panel-heading">
              <h3 class="panel-title">Bienvenue sur mon site !</h3>
            </div>
            <div class="panel-body">
              <p>"Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia
voluptas
sit aspernatur aut odit aut fugit, sed quia consequuntur magni
dolores eos qui ratione voluptatem sequi nesciunt. Neque porro
quisquam est,
qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit,
sed quia non numquam eius modi tempora incidunt ut labore et dolore
magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis
nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut
aliquid ex ea commodi consequatur? Quis autem vel eum iure
reprehenderit qui in ea voluptate velit esse quam nihil molestiae
consequatur,
vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?"</p>
              <p>"Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia

```

```

voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur
magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro
quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,
adipisci velit, sed quia non numquam eius modi tempora incidunt ut
labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad
minima veniam, quis nostrum exercitationem ullam corporis suscipit
laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem
vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil
molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas
nulla pariatur?"</p>

```

```

<p>"Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia
voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur
magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro
quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,
adipisci velit, sed quia non numquam eius modi tempora incidunt ut
labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad
minima veniam, quis nostrum exercitationem ullam corporis suscipit
laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem
vel eum iure reprehenderit qui in ea voluptate velit esse quam
nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo
voluptas nulla pariatur?"</p>

```

```

</div>
</div>
</section>
</div>
<footer class="row col-sm-12">
  <div class="panel panel-body">
    <p>Tous droits réservés par moi...</p>
  </div>
</footer>
</div>
</body>
</html>

```

La page est organisée en 4 parties :

1. bannière supérieure
2. menu à gauche
3. contenu à droite
4. pied de page en bas

Bootstrap permet de réaliser cette mise en page avec facilité. La mise en forme effectuée par Bootstrap donne la figure suivante.



Une page d'exemple

Tester

C'est déjà pas si mal, mais on va améliorer la présentation avec du style :

Code : CSS

```

body {
  background-color: #bbb;
  color: #048;
  font-size: 12px;
  line-height: 16px;
}
a {
  color: #028;
}
footer {
  text-align: center;
}
nav {
  background-color: #bcd;
  border-radius: 8px;
  padding: 0 !important;
  border: 1px solid #ddd;
  margin-bottom: 20px;
}
section {
  padding-right: 0 !important;
}
.panel {
  background-color: #bcd;
  border-radius: 8px;
  padding-bottom: 5px;
}
h1 {
  text-shadow: 4px 4px 5px #777;
  color: #449;
}
.nav > li > a:hover, .nav > li > a:focus {
  background-color: #cde;
}
.panel-default > .panel-heading, .nav-pills > li.active > a, .nav-
pills > li.active > a:hover, .nav-pills > li.active > a:focus {
  color: #048;
  background-color: #abc;
  border-radius: 8px;
}

```

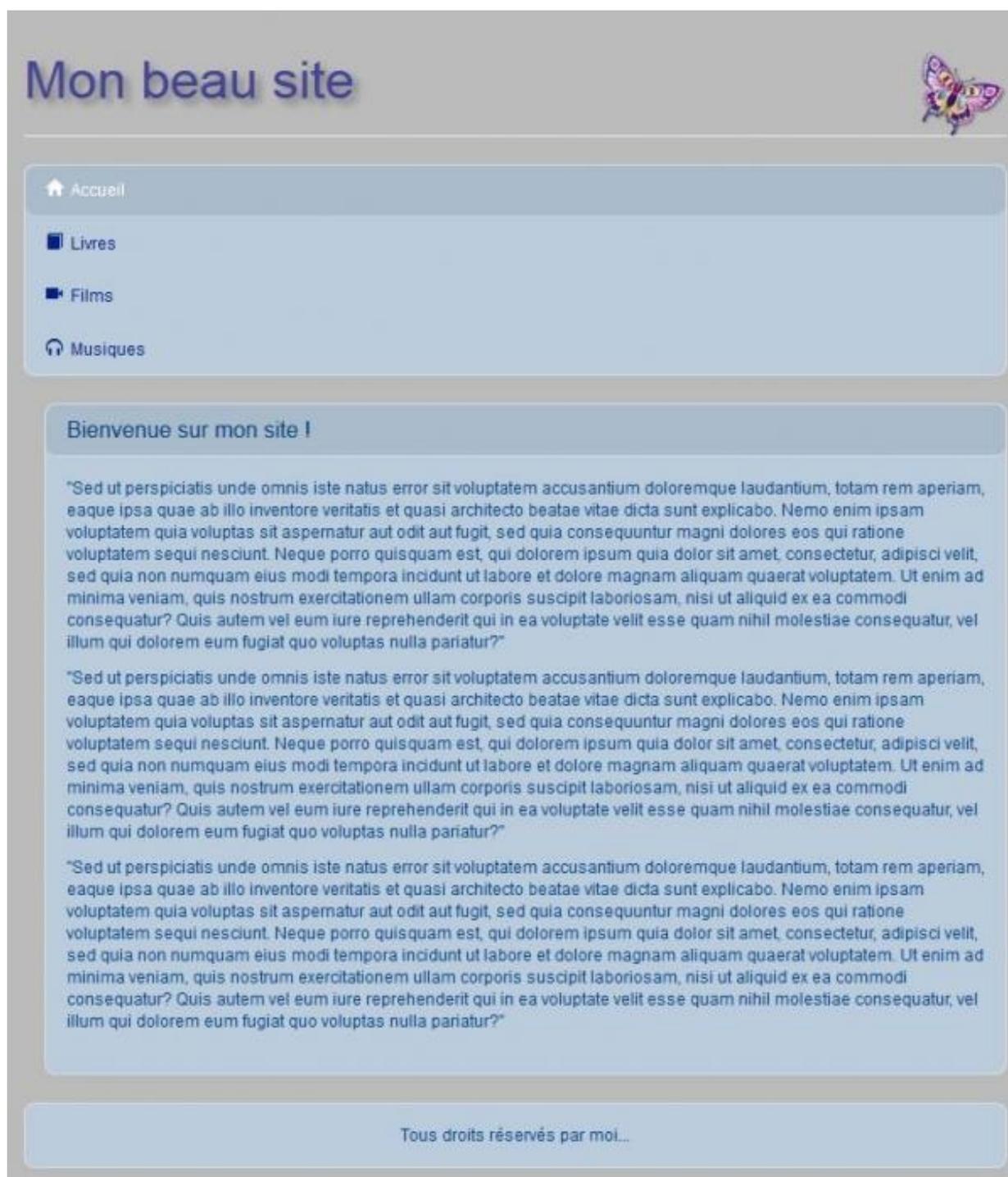
Ce qui nous donne la figure suivante.



La même page stylisée

Tester

Avec une réaction correcte quand on réduit l'affichage au-dessous de 768 pixels, comme le montre la figure suivante.



Effet de la

réduction de l'affichage

J'ai profité de l'occasion pour montrer comment créer facilement un menu vertical élégant avec Bootstrap. Mais ce qui va nous intéresser concerne les styles. Parce que toute une partie de la stylisation est en fait une modification des paramètres de base de Bootstrap. J'ai changé la couleur de fond, la couleur de premier plan, la couleur des liens, la taille des caractères... On peut se demander s'il n'est pas plus simple de modifier directement Bootstrap...

Configurer en ligne sur le site de Bootstrap

Les composants

Allez sur [cette page](#). Vous trouvez un grand formulaire qui vous permet de mettre Bootstrap non seulement à votre sauce côté présentation, mais aussi de confectionner un fichier le plus léger possible. Vous sélectionnez les composants que vous utilisez. Pour la page d'exemple, cela ressemble à la figure suivante.

LESS files

Choose which LESS files to compile into your custom build of Bootstrap. Not sure which files to use? Read through the [CSS](#) and [Components](#) pages in the docs

Common CSS

- Print media styles
- Typography
- Code
- Grid system
- Tables
- Forms
- Buttons

Components

- Glyphicons
- Button groups
- Input groups
- Navs
- Navbar
- Breadcrumbs
- Pagination
- Pager
- Labels
- Badges
- Jumbotron
- Thumbnails
- Alerts
- Progress bars
- Media items
- List groups
- Panels
- Wells
- Close icon

JavaScript components

- Dropdowns
- Tooltips
- Popovers
- Modals
- Carousel

Utilities

- Basic utilities
- Responsive utilities
- Component animations (for JS)

Composants sélectionnés pour la page d'exemple

Comme on n'utilise aucun plugin jQuery, on décoche tout avec le bouton `Toggle all` (voir figure suivante).

jQuery plugins

Choose which jQuery plugins should be included in your custom JavaScript files. Unsure what to include? Read the [JavaScript](#) page in the docs.

Linked to components

- Alert dismissal
- Advanced buttons
- Carousel functionality
- Dropdowns
- Modals
- Tooltips
- Popovers (requires Tooltips)
- Toggable tabs

Magic

- Affix
- Collapse
- Scrollspy
- Transitions (required for any kind of animation)

Tous les composants jQuery désélectionnés

On ne tient pas compte pour le moment des variables LESS. Il n'y a plus qu'à cliquer sur le bouton pour obtenir un Bootstrap personnalisé.

On obtient 2 fichiers : `bootstrap.css` et `bootstrap.min.css`. Il pèsent respectivement 49 Ko et 40 Ko, alors que les versions complètes font 122 Ko et 101 Ko. La différence est loin d'être négligeable ! Il faut quand même vérifier que la page fonctionne encore correctement.

Les variables

Il est possible d'aller plus loin dans la personnalisation. Regardons les variables disponibles. Par exemple dans les variables du scaffolding, sur la même page de configuration, on trouve celle qui concerne la couleur de fond de la page `@body-bg` (voir figure suivante).

Scaffolding

@body-bg

La variable @body-bg

Background color for `<body>`.

Par défaut, cette valeur est fixée à `#fff`, autrement dit la couleur blanche. Or pour ma page, il me faut la valeur `#bbb`. Je peux donc changer la valeur à ce niveau, je n'aurai ainsi plus besoin de fixer cette propriété dans ma feuille de style personnelle. Quelles autres variables sont intéressantes dans le cas de notre page ? Je trouve par exemple la variable `@text-color` (voir figure suivante).

@text-color

La variable @text-color

Global text color on `<body>`.

Je peux la fixer à `#048`. On va en rester là pour cet exemple, et créer un nouveau fichier Bootstrap avec ce paramétrage. On compile le fichier, on le télécharge et on lui donne un nouveau nom :

Code : HTML

```
<link href="assets/css/bootstrap.min.v1_1.css" rel="stylesheet">
```

Il faut aussi supprimer les styles devenus redondants :

Code : CSS

```
body {  
  /* background-color:#bbb;  
  color:#048; */  
  font-size: 12px;  
  line-height: 16px;  
}  
a {  
  color: #028;  
}  
footer {
```

```
    text-align:center;
  }
  nav {
    background-color:#bcd;
    border-radius: 8px;
    padding: 0 !important;
    border: 1px solid #ddd;
    margin-bottom: 20px;
  }
  section {
    padding-right: 0 !important;
  }
  .panel {
    background-color: #bcd;
    border-radius: 8px;
    padding-bottom: 5px;
  }
  h1 {
    text-shadow: 4px 4px 5px #777;
    color: #449;
  }
  .nav > li > a:hover, .nav > li > a:focus {
    background-color: #cde;
  }
  .panel-default > .panel-heading, .nav-pills > li.active > a, .nav-
  pills > li.active > a:hover, .nav-pills > li.active > a:focus {
    color: #048;
    background-color: #abc;
    border-radius: 8px;
  }
}
```

Tester !

On obtient le même résultat. On peut évidemment se demander l'intérêt de procéder ainsi pour gagner juste quelques règles de style. C'est effectivement se donner bien du mal pour pas grand chose. En revanche, cette démarche peut devenir intéressante dans le cadre de la création d'un thème spécifique avec beaucoup de réglages de variables.

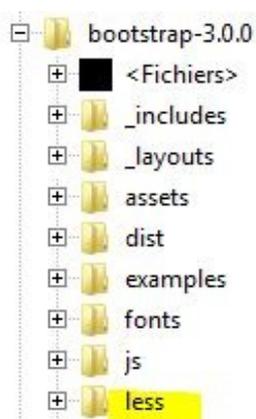
LESS, un langage de style dynamique

Bootstrap est totalement paramétrable parce qu'il est basé sur LESS. C'est un langage qui permet d'étendre les possibilités du CSS avec des variables, des opérations, des fonctions. La version 3.1 de Bootstrap a vu également l'arrivée très attendue d'un portage sur [Sass](#), qui est un autre langage qui permet de gérer le CSS. Pour le moment, celui-ci se situe à part sur [une page GitHub](#).

Faisons à présent un petit passage en revue des possibilités de LESS...

Les sources des fichiers LESS de Bootstrap

Vous pouvez trouver les sources LESS de Bootstrap dans le pack [que vous téléchargez sur le site](#) (voir figure suivante).



Le dossier des fichiers LESS de Bootstrap

Nous allons les prendre comme exemple pour illustrer les possibilités de LESS.

Des variables

Une des principales fonctionnalités de LESS est de permettre la création de variables et d'ainsi centraliser les valeurs et faciliter les modifications. Ouvrez le fichier `variables.less` qui définit la valeur de toutes les variables utilisées par Bootstrap. En voici un extrait :

Code : CSS

```
...
// Buttons
// -----

@btn-font-weight:          normal;

@btn-default-color: #333;
@btn-default-bg:         #fff;
@btn-default-border:     #ccc;
...
```

Une variable est déclarée avec une syntaxe simple : il suffit d'utiliser le caractère `@` suivi du nom de la variable, par exemple on a ici la variable `@btn-default-color`. Une variable prend une valeur, ici la variable `@btn-default-color` a la valeur `#333`. On peut ensuite utiliser cette variable pour affecter la valeur correspondante. Ouvrez le fichier `buttons.less` et trouvez ces lignes :

Code : CSS

```
.btn {
  ...
  &:focus {
    color: @btn-default-color;
    text-decoration: none;
  }
}
```

On voit qu'on utilise la variable `@btn-default-color` pour affecter la valeur de la propriété `color` pour le focus des boutons. Remarquez au passage l'utilisation du signe `&` pour faciliter l'écriture du code. Pour la même classe `.btn`, on peut regrouper des sélecteurs dans un même bloc sans tout réécrire. On trouve la même variable utilisée un peu plus loin dans le code :

Code : CSS

```
.btn-default {
  .button-variant(@btn-default-color; @btn-default-bg;
  @btn-default-border);
}
```

Il suffit donc de modifier la valeur de la variable pour que toutes ces affectations soient changées avec cette seule action.

Et si on regarde le résultat, on retrouve bien la couleur `#333` affectée à la couleur du bouton pour le focus :

Code : CSS

```
.btn:focus {
  color: #333333;
  text-decoration: none;
}
```

Cette factorisation est pratique si elle est bien organisée, ce qui est le cas avec Bootstrap qui centralise toutes les variables dans le même fichier. Rien n'empêche aussi d'affecter la valeur d'une variable à une autre variable comme ici :

Code : CSS

```
@brand-primary:          #428bca;
...
@link-color:             @brand-primary;
```

Inclusion de fichier

Les possibilités de LESS sont loin de s'arrêter à la définition de variables. On peut inclure un fichier avec la commande `@import`. Par exemple, le fichier `bootstrap.less` ne fait que des inclusions :

Code : CSS

```
...
// Core variables and mixins
@import "variables.less";
@import "mixins.less";

// Reset
@import "normalize.less";
@import "print.less";
...
```

On peut importer des fichiers `less` ou `css`. S'il s'agit d'un fichier `less`, il est évidemment traité s'il comporte des commandes spéciales, alors que dans le cas d'un fichier `css`, il est pris tel quel sans traitement particulier.

Les mixins

On peut aussi inclure des classes dans d'autres classes (ce qui se nomme `mixins` dans le langage LESS), comme si elle était une propriété. On peut le voir ici dans le fichier `grid.less` :

Code : CSS

```
// Set the container width, and override it for fixed navbars in
media queries
.container {
  .container-fixed();
}
```

On voit que la classe `.container-fixed` est utilisée comme une propriété. Examinons cette classe appelée qui se situe dans le fichier `mixin.less` :

Code : CSS

```
// Centered container element
.container-fixed() {
  margin-right: auto;
  margin-left: auto;
  padding-left: (@grid-gutter-width / 2);
  padding-right: (@grid-gutter-width / 2);
  &:extend(.clearfix all);
}
```

On y trouve un certain nombre de règles, essentiellement pour centrer l'élément et même un autre appel à une classe, ici `.clearfix`. Vous avez remarqué sans doute la syntaxe particulière avec l'ajout de parenthèses comme s'il s'agissait d'une fonction. Cette analogie va plus loin, puisqu'on peut même transmettre des paramètres (on parle alors de mixin paramétré) à une classe incluse comme ici dans le fichier `button-groups.less` :

Code : CSS

```
.btn-group.open .dropdown-toggle {
  .box-shadow(inset 0 3px 5px rgba(0,0,0,.125));
}
```

Voyons la classe appelée qui se situe encore dans le fichier `mixin.less` :

Code : CSS

```
// Drop shadows
.box-shadow(@shadow) {
  -webkit-box-shadow: @shadow; // iOS <4.3 & Android <4.1
  box-shadow: @shadow;
}
```

On voit que cette classe réalise un ombrage selon le paramètre transmis. Si on regarde le résultat, on a bien ce qui est demandé :

Code : CSS

```
.btn-group.open .dropdown-toggle {
  -webkit-box-shadow: inset 0 3px 5px rgba(0, 0, 0, 0.125);
  box-shadow: inset 0 3px 5px rgba(0, 0, 0, 0.125);
}
```

Sélecteurs emboîtés

On peut inclure des sélecteurs dans une définition, comme ici dans le fichier `dropdowns.less` :

Code : CSS

```
// Open state for the dropdown
// -----
.open {
  // Show the menu
  > .dropdown-menu {
    display: block;
  }

  // Remove the outline when :focus is triggered
  > a {
    outline: 0;
  }
}
```

Cela simplifie l'écriture et rend le code plus compact. Voici le résultat :

Code : CSS

```
.open > .dropdown-menu {
```

```

    display: block;
}
.open > a {
    outline: 0;
}

```

On a vu plus haut que pour les pseudo-classes, on peut utiliser le signe &, comme par exemple ici dans le fichier `pagination.less` :

Code : CSS

```

.pagination {
    ...
    > li > a,
    > li > span {
        &:hover,
        &:focus {
            color: @pagination-hover-color;
            background-color: @pagination-hover-bg;
            border-color: @pagination-hover-border;
        }
    }
}

```

On trouve ce résultat :

Code : CSS

```

.pagination > li > a:hover,
.pagination > li > span:hover,
.pagination > li > a:focus,
.pagination > li > span:focus {
    background-color: #eeeeee;
}

```

Opérations

On peut aussi effectuer des opérations comme ici dans le fichier `forms.less` :

Code : CSS

```

legend {
    display: block;
    width: 100%;
    padding: 0;
    margin-bottom: @line-height-computed;
    font-size: (@font-size-base * 1.5);
    line-height: inherit;
    color: @gray-dark;
    border: 0;
    border-bottom: 1px solid @legend-border-color;
}

```

Avec le résultat :

Code : CSS

```

legend {
    display: block;
}

```

```

width: 100%;
padding: 0;
margin-bottom: 20px;
font-size: 21px;
line-height: inherit;
color: #333333;
border: 0;
border-bottom: 1px solid #e5e5e5;
}

```

Comme on a une dimension de base de 14 pixels, le résultat $14 * 1.5$ donne bien 21 pixels.

Fonctions

LESS possède des fonctions bien pratiques, regardez ce code :

Code : CSS

```

@brand-primary:          #428bca;
...
@link-color:             @brand-primary;
@link-hover-color:      darken(@link-color, 15%);

```

On utilise la fonction intégrée `darken`, qui assombrit la couleur du pourcentage indiqué. Ici la valeur de la couleur est issue d'une variable, dont la valeur est elle-même issue d'une autre variable ! Au final, on assombrit de 15% la couleur `#428bca`. On trouve aussi la fonction opposée `lighten`:

Code : CSS

```

@gray-dark:             lighten(#000, 20%); // #333

```

Ici, on éclaircit le noir `#000` de 20%, ce qui donne `#333`. On trouve d'autres fonctions classiques comme `ceil`, qui retourne l'entier immédiatement supérieur à une valeur :

Code : CSS

```

@font-size-base:       14px;
@font-size-large:      ceil(@font-size-base * 1.25); // ~18px

```

Le résultat de $14 * 1.25$ est 17.5, l'arrondi par excès donne bien 18. On trouve la fonction `floor` pour l'arrondi par défaut, ainsi que bien d'autres fonctions que vous pouvez découvrir [dans la documentation](#).

Un exemple synthétique

Pour illustrer les possibilités et l'élégance de LESS je vais prendre un exemple qui regroupe la plupart de ses possibilités en l'analysant. Vous trouvez dans le fichier `jumbotron.less` ce code :

Code : CSS

```

.jumbotron {
padding: @jumbotron-padding;
margin-bottom: @jumbotron-padding;
color: @jumbotron-color;
background-color: @jumbotron-bg;
}

```

```

h1,
.h1 {
  color: @jumbotron-heading-color;
}
p {
  margin-bottom: (@jumbotron-padding / 2);
  font-size: @jumbotron-font-size;
  font-weight: 200;
}

.container & {
  border-radius: @border-radius-large; // Only round corners at
higher resolutions if contained in a container
}

.container {
  max-width: 100%;
}

@media screen and (min-width: @screen-sm-min) {
  padding-top: (@jumbotron-padding * 1.6);
  padding-bottom: (@jumbotron-padding * 1.6);

  .container & {
    padding-left: (@jumbotron-padding * 2);
    padding-right: (@jumbotron-padding * 2);
  }

  h1,
  .h1 {
    font-size: (@font-size-base * 4.5);
  }
}

```

Considérons ces instructions pas à pas.

Les propriétés directes

Il y a d'abord un premier groupe de propriétés :

Code : CSS

```

padding: @jumbotron-padding;
margin-bottom: @jumbotron-padding;
color: @jumbotron-color;
background-color: @jumbotron-bg;

```

Les valeurs sont renseignées au moyen de variables. On peut trouver leurs valeurs dans le fichier `variables.less` :

Code : CSS

```

@jumbotron-padding: 30px;
@jumbotron-color: inherit;
@jumbotron-bg: @gray-lighter;

```

Il faut chercher encore un peu dans le même fichier pour obtenir la valeur pour la couleur de fond :

Code : CSS

```

@gray-lighter: lighten(#000, 93.5%); // #eee

```

Après compilation on trouve les valeurs suivantes :

Code : CSS

```
.jumbotron {  
  padding: 30px;  
  margin-bottom: 30px;  
  color: inherit;  
  background-color: #eee;  
}
```

Les sélecteurs emboîtés

On trouve ensuite des sélecteurs emboîtés :

Code : CSS

```
h1,  
.h1 {  
  color: @jumbotron-heading-color;  
}  
p {  
  margin-bottom: (@jumbotron-padding / 2);  
  font-size: @jumbotron-font-size;  
  font-weight: 200;  
}
```

Dans les blocs de ces sélecteurs, on a une propriété directe pour `font-weight`, et les autres dont la valeur est issue d'une variable et même d'un calcul pour `margin-bottom`. Il faut à nouveau regarder dans le fichier des variables pour trouver les valeurs :

Code : CSS

```
@jumbotron-padding: 30px;  
@jumbotron-color: inherit;  
@jumbotron-font-size: ceil(@font-size-base * 1.5);
```

On voit que la dimension du texte est dépendante de la dimension de base pour la page, ce qui est très logique. Le résultat après compilation :

Code : CSS

```
.jumbotron h1,  
.jumbotron .h1 {  
  color: inherit;  
}  
.jumbotron p {  
  margin-bottom: 15px;  
  font-size: 21px;  
  font-weight: 200;  
}
```

Classe emboîtée

On trouve ensuite une classe emboîtée :

Code : CSS

```
.container & {
  border-radius: @border-radius-large;
}
```

Remarquez l'utilisation de l'opérateur de concaténation &. On veut donc associer dans le sélecteur la classe `.container` et la classe `jumbotron`. On retrouve donc logiquement ce résultat :

Code : CSS

```
.container .jumbotron {
  border-radius: 6px;
}
```

Medias Queries

On trouve enfin un bloc qui concerne les Medias Queries :

Code : CSS

```
@media screen and (min-width: @screen-tablet) {
  padding-top: (@jumbotron-padding * 1.6);
  padding-bottom: (@jumbotron-padding * 1.6);

  .container & {
    padding-left: (@jumbotron-padding * 2);
    padding-right: (@jumbotron-padding * 2);
  }

  h1 {
    font-size: (@font-size-base * 4.5);
  }
}
```

À l'intérieur de ce bloc, on a le même genre d'organisation que celle nous avons vu précédemment avec des propriétés directement renseignées, un sélecteur et une classe emboîtés. Le résultat à la compilation donne :

Code : CSS

```
@media screen and (min-width: 768px) {
  .jumbotron {
    padding-top: 48px;
    padding-bottom: 48px;
  }
  .container .jumbotron {
    padding-right: 60px;
    padding-left: 60px;
  }
  .jumbotron h1 {
    font-size: 63px;
  }
}
```

Il y aurait encore beaucoup à dire sur LESS, je vous laisse le découvrir dans [la documentation du site](#) qui est très bien faite.

Utilisation de LESS dans Bootstrap

Un programme pour LESS

Vous pouvez modifier Bootstrap en tripatouillant les variables et en compilant avec les outils proposés... Je vous propose d'utiliser un programme très bien fait : [Crunch](#). Au moment où j'écris ce cours, j'ai trouvé la version 1.8.1.

Le programme est simple et efficace, vous pouvez créer ou modifier des fichiers LESS et les compiler en CSS. Sur le site de Bootstrap, il est préconisé d'utiliser [RECESS](#), qui est une application en ligne de commande beaucoup moins ergonomique. Mais les essais que j'ai effectués avec Crunch ne m'ont donné aucun souci.

Exemple simple de configuration

Nous allons reprendre la configuration effectuée dans un chapitre précédent à partir de l'interface simplifiée de Bootstrap, mais cette fois en intervenant directement dans les fichiers LESS. Voici la procédure avec Crunch :

1. Rapatrier tous les fichiers LESS de la [dernière version de Bootstrap](#) et les mettre dans un répertoire local.
2. Lancer Crunch et ouvrir le fichier `variables.less`
3. Modifier les variables concernées dans le fichier `variables.less` (voir figure suivante).
4. Enregistrer la nouvelle version du fichier `variables.less`
5. Ouvrir le fichier `bootstrap.less`
6. Pour être cohérent avec notre précédente configuration, nous devons supprimer les éléments inutiles
7. Compiler le fichier `bootstrap.less`
8. Enregistrer le fichier `bootstrap.css` généré par Crunch
9. Renommer le fichier `bootstrap.css` en `bootstrap.min.v1_2.css`.
10. Modifier le lien des styles dans la page web
11. Supprimer les styles devenus inutiles dans la page web

```
23 //== Scaffolding
24 //
25 // ## Settings for some of the most global styles.
26
27 /** Background color for `<body>`.
28 @body-bg:          #bbb;
29 /** Global text color on `<body>`.
30 @text-color:       #048;
31
```

Changement de valeurs dans le fichier

variables.less

Code : CSS

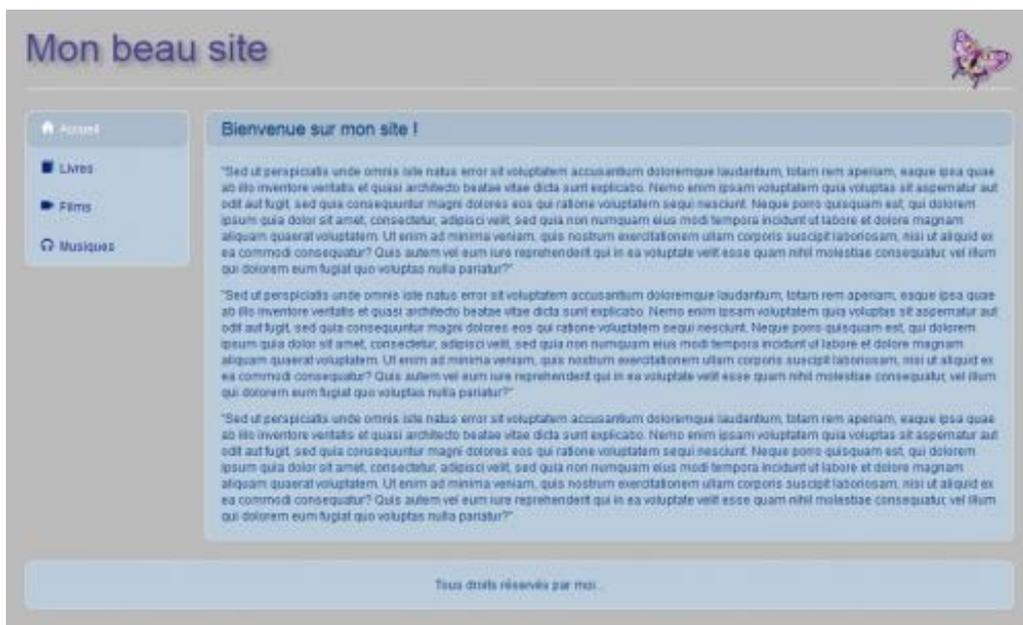
```
body {
  /* background-color:#bbb;
  color:#048; */
  font-size: 12px;
  line-height: 16px;
}
a {
  color: #028;
}
footer {
  text-align:center;
}
nav {
  background-color:#bcd;
  border-radius: 8px;
  padding: 0 !important;
  border: 1px solid #ddd;
  margin-bottom: 20px;
}
section {
```

```

padding-right: 0 !important;
}
.panel {
background-color: #bcd;
border-radius: 8px;
padding-bottom: 5px;
}
h1 {
text-shadow: 4px 4px 5px #777;
color: #449;
}
.nav > li > a:hover, .nav > li > a:focus {
background-color: #cde;
}
.panel-default > .panel-heading, .nav-pills > li.active > a, .nav-
pills > li.active > a:hover, .nav-pills > li.active > a:focus {
color: #048;
background-color: #abc;
border-radius: 8px;
}

```

On obtient alors la même présentation que précédemment, comme le montre la figure suivante.



Apparence obtenue

Tester

Réflexions sur la configuration de Bootstrap avec LESS

Nous avons vu la procédure à employer avec Crunch. Les seuls fichiers qui ont été modifiés sont `bootstrap.less` et `variables.less`. Ce dernier fichier contient un certain nombre de variables qui influent sur la présentation générale et sur les composants. Pour notre exemple, j'ai pu changer les éléments suivants :

- Couleur de fond de la page
- Couleur du texte de la page

J'aurais pu également intervenir sur ceux-là :

- Dimension des polices (et de façon automatisé la hauteur des lignes)
- Couleur des liens
- Couleur de fond des panneaux

Mais tout le reste correspond à des ajustements de la page qui ne rentrent pas dans le cadre du fichier `variable.less`. On peut du coup se demander si le jeu en vaut la chandelle... Dans le cadre d'un site web ponctuel, la réponse est clairement "non". À quoi bon aller bidouiller les fichiers LESS pour économiser quelques lignes de code ? On pourrait aller plus loin et intervenir sur les autres fichiers pour effectuer tous les ajustements nécessaires, par exemple, pour le "padding" inférieur du panneau, il faudrait intervenir sur le fichier `panels.less` :

Code : CSS

```
// Optional heading
.panel-heading {
padding: 10px 15px;
border-bottom: 1px solid @panel-border;
border-top-radius(@panel-border-radius - 1);
}
```

On arriverait de proche en proche à ajuster complètement Bootstrap pour notre site et ainsi n'avoir plus qu'un fichier CSS à charger. Cette conception se heurte à plusieurs inconvénients, dont le fait de ne pas pouvoir faire de mise à jour de Bootstrap sans refaire toutes nos interventions. Envisageons donc une autre approche...

Puisque nous disposons de LESS, autant l'utiliser aussi pour nos styles complémentaires. Créons un fichier `main.less` avec nos styles dans le dossier qui contient tous les fichiers LESS.

Code : CSS

```
body {
background-color:#bbb;
color:#048;
font-size: 12px;
line-height: 16px;
}
a {
color: #028;
}
footer {
text-align:center;
}
nav {
background-color:#bcd;
border-radius: 8px;
padding: 0 !important;
border: 1px solid #ddd;
margin-bottom: 20px;
}
section {
padding-right: 0 !important;
}
.panel {
background-color: #bcd;
border-radius: 8px;
padding-bottom: 5px;
}
h1 {
text-shadow: 4px 4px 5px #777;
color: #449;
}
.nav > li > a:hover, .nav > li > a:focus {
background-color: #cde;
}
.panel-default > .panel-heading, .nav-pills > li.active > a, .nav-
pills > li.active > a:hover, .nav-pills > li.active > a:focus {
color: #048;
background-color: #abc;
border-radius: 8px;
}
```

Une fois ce fichier créé, il faut informer le fichier `bootstrap.less` qu'il doit en tenir compte (voir figure suivante).

```
57 // Utility classes
58 @import "utilities.less";
59 //@import "responsive-utilities.less";
60
61 // Mes styles
62 @import "main.less";
```

Importation des styles de la page

Il suffit ensuite de compiler tout ça ; on obtient un fichier minifié de Bootstrap qui contient toutes les règles. On se retrouve donc avec un seul appel de fichier CSS (on peut supprimer le style inclus dans la page) pour un résultat identique pour notre page d'exemple :

[Tester !](#)

Le résultat est très « propre », avec un seul fichier CSS à charger et aucune règle de style dans la page. D'autre part, s'il y a une mise à jour à faire, il suffit de recompiler.

Les mixins

La grille avec les mixins

Il existe un fichier très intéressant, c'est `mixins.less`, qui ne contient que des mixins (j'ai parlé dans un chapitre précédent de ces entités). Pour comprendre son utilité il faut se poser une petite question. On place les classes de Bootstrap directement dans le code HTML. En agissant ainsi, on rend le code très dépendant de Bootstrap. D'autre part, cette façon de procéder s'éloigne du principe de base qui consiste à bien séparer le contenu de la forme. Lorsque j'écris ce code dans mon exemple :

Code : HTML

```
<div class="row col-sm-12">
  <nav class="col-sm-2"> ... </nav>
  <section class="col-sm-10"> ... </section>
</div>
```

Je dis clairement que je veux une rangée avec deux colonnes en indiquant la proportion de chacune d'elles. Si je n'utilisais pas Bootstrap, ce serait uniquement mes styles qui décideraient de cela. D'autre part, si je veux changer de framework un jour, je dois intervenir au niveau de mon code HTML. Nous allons voir comment créer une grille en utilisant uniquement LESS. Regardez ces parties du fichier `mixins.less` :

Code : CSS

```
// Creates a wrapper for a series of columns
.make-row(@gutter: @grid-gutter-width) {
  margin-left: (@gutter / -2);
  margin-right: (@gutter / -2);
  &:extend(.clearfix all);
}
...
// Generate the small columns
.make-sm-column(@columns; @gutter: @grid-gutter-width) {
  position: relative;
  min-height: 1px;
  padding-left: (@gutter / 2);
  padding-right: (@gutter / 2);

  @media (min-width: @screen-sm) {
```

```

float: left;
width: percentage((@columns / @grid-columns));
}
}

```

On trouve deux mixins paramétrés intéressants pour notre exemple : `make-row()` et `make-sm-column()`. On va utiliser ces mixins dans notre exemple en modifiant le code HTML :

Code : HTML

```

<!DOCTYPE HTML>
<html>

  <head>
    <meta charset="utf-8">
    <link href="assets/css/bootstrap.min.v1_4.css" rel="stylesheet">
  </head>

  <body>
    <div class="container">
  <header>
    <div class="page-header">
      <h1>Mon beau site</h1>
    </div>
  </header>
  <div class="contenu">
  <nav>
    <ul class="nav nav-pills nav-stacked">
      <li class="active"> <a href="#"> <span class="glyphicon
glyphicon-home"></span>&nbsp; Accueil </a> </li>
      <li> <a href="#"> <span class="glyphicon glyphicon-
book"></span>&nbsp; Livres </a> </li>
      <li> <a href="#"> <span class="glyphicon glyphicon-
facetime-video"></span>&nbsp; Films </a> </li>
      <li> <a href="#"> <span class="glyphicon glyphicon-
headphones"></span>&nbsp; Musiques </a> </li>
    </ul>
  </nav>
  <!-- <div class="clearfix visible-xs"></div> -->
  <section>
    <div class="panel panel-default">
      <div class="panel-heading">
        <h3 class="panel-title">Bienvenue sur mon site !</h3>
      </div>
      <div class="panel-body">
        <p>Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia
voluptas
sit aspernatur aut odit aut fugit, sed quia consequuntur magni
dolores eos qui ratione voluptatem sequi nesciunt. Neque porro
quisquam est,
qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit,
sed quia non numquam eius modi tempora incidunt ut labore et dolore
magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis
nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut
aliquid ex ea commodi consequatur? Quis autem vel eum iure
reprehenderit qui in ea voluptate velit esse quam nihil molestiae
consequatur,
vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?"</p>
        <p>Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia
voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur

```

```
magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro
quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,
adipisci velit, sed quia non numquam eius modi tempora incidunt ut
labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad
minima veniam, quis nostrum exercitationem ullam corporis suscipit
laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem
vel eum iure reprehenderit qui in ea voluptate velit esse quam
nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo
voluptas nulla pariatur?"</p>
```

```
<p>"Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto
beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia
voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur
magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro
quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur,
adipisci velit, sed quia non numquam eius modi tempora incidunt ut
labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad
minima veniam, quis nostrum exercitationem ullam corporis suscipit
laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem
vel eum iure reprehenderit qui in ea voluptate velit esse quam
nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo
voluptas nulla pariatur?"</p>
```

```
</div>
</div>
</section>
</div>
<footer>
  <div class="panel panel-body">
    <p>Tous droits réservés par moi...</p>
  </div>
</footer>
</div>
</body>
</html>
```

J'ai supprimé les classes `row`, `col-sm-2`, `col-sm-10` et `col-sm-12` et juste ajouté la classe `contenu`. Mon code est maintenant plus "propre", mais je dois mettre à jour mon fichier `main.less` :

Code : CSS

```
.contenu, header, footer {
  .make-row();
  .make-sm-column(12);
}
body {
  background-color:#bbb;
  color:#048;
  font-size: 12px;
  line-height: 16px;
}
a {
  color: #028;
}
footer {
  text-align:center;
}
nav {
  .make-sm-column(2);
  background-color:#bcd;
  border-radius: 8px;
  padding: 0 !important;
  border: 1px solid #ddd;
  margin-bottom: 20px;
}
```

```

section {
  .make-sm-column(10);
  padding-right: 0 !important;
}
.panel {
  background-color: #bcd;
  border-radius: 8px;
  padding-bottom: 5px;
}
h1 {
  text-shadow: 4px 4px 5px #777;
  color: #449;
}
.nav > li > a:hover, .nav > li > a:focus {
  background-color: #cde;
}
.panel-default > .panel-heading, .nav-pills > li.active > a, .nav-
pills > li.active > a:hover, .nav-pills > li.active > a:focus {
  color: #048;
  background-color: #abc;
  border-radius: 8px;
}

```

[Tester !](#)

Je vous conseille d'explorer les possibilités des fichiers mixins `.less`. Et rien ne vous empêche évidemment de créer vos propres classes pour rendre votre code HTML complètement indépendant de Bootstrap.

Les images "Retina"

Prenons un autre exemple de l'utilité des mixins avec le problème des images dites "Retina". Ce terme a été créé par Apple pour désigner les écrans ayant une résolution très élevée (320 dpi pour les iPhone 4 et 5). Sous Android, on parle de HDPI. Quel que soit le terme utilisé, le problème est le même. Mais quel est-il ? Si vous avez par exemple une image de 100 x 100 px, celle-ci va bien s'afficher sur un écran avec une résolution standard. En revanche, avec un écran haute résolution, l'image va être étirée et ne va plus être aussi jolie. Est-ce à cause de la résolution de l'écran ? Pas complètement. C'est surtout parce que l'appareil ne semble pas au courant de la résolution dont il dispose. Par exemple, un iPhone 4 avec une résolution de 320 dpi et une largeur d'écran de 640 px pense disposer seulement de 320 px. Du coup notre image de 100 px de côté va apparaître floue. La solution est donc de régler le pixel-ratio (le rapport entre les pixels physiques et les pixels déclarés). Voici le mixin proposé par Bootstrap pour régler ce problème :

Code : CSS

```

.img-retina(@file-1x; @file-2x; @width-1x; @height-1x) {
  background-image: url("@{file-1x}");

  @media
  only screen and (-webkit-min-device-pixel-ratio: 2),
  only screen and ( min--moz-device-pixel-ratio: 2),
  only screen and ( -o-min-device-pixel-ratio: 2/1),
  only screen and ( min-device-pixel-ratio: 2),
  only screen and ( min-resolution: 192dpi),
  only screen and ( min-resolution: 2dppx) {
    background-image: url("@{file-2x}");
    background-size: @width-1x @height-1x;
  }
}

```

On constate qu'on joue effectivement sur le pixel-ratio. Pour utiliser ce mixin, il faut renseigner les paramètres :

- Image de dimension de base (par exemple 100px * 100px)
- Image de double dimension de base (200px * 200px)
- Largeur de base en pixels

- Hauteur de base en pixels

Concrètement, on va écrire cette règle :

Code : CSS

```
.img-retina("image_de_base.jpg", "image_double.jpg", 100px, 100px);
```

Et les règles de styles seront automatiquement générées pour nous !

Utilisation dynamique de LESS

Dans l'exemple précédent, nous avons compilé un fichier CSS avec Crunch pour ensuite le mettre dans le site. Il existe une façon dynamique d'utiliser LESS qui consiste à référencer directement le fichier source, et de prévoir également un fichier Javascript pour l'interpréter à l'arrivée. C'est une méthode plus lourde côté client qui ne me paraît pas vraiment adaptée en mode production, mais c'est celle qui est préconisée sur le site de référence. Elle peut toutefois trouver sa justification en phase de développement pour tester rapidement des nouveaux réglages.

Reprenons notre exemple de page cette fois avec cette procédure. Voici la différence au niveau de l'en-tête :

Code : HTML

```
<link rel="stylesheet/less" type="text/css"
href="assets/less/bootstrap.less">
<script src="assets/js/less-1.4.1.min.js"></script>
```

[Tester !](#)

Évidemment, cela ne marche que si vous mettez bien tous les fichiers LESS nécessaires sur le serveur. Vous constaterez un temps de chargement relativement long. Mais ensuite, il suffit de changer un élément dans un fichier LESS pour obtenir le résultat. Par exemple si je change la couleur de fond dans le fichier `variables.less`, il suffit de recharger la page pour voir le résultat immédiatement.

Si on regarde le code généré par le navigateur, on constate que le CSS a bien été construit (voir figure suivante).

```

3   <link rel="stylesheet/less" type="text/css" href="assets/less/bootstrap.less"><style
4   * Bootstrap v3.0.0
5   *
6   * Copyright 2013 Twitter, Inc
7   * Licensed under the Apache License v2.0
8   * http://www.apache.org/licenses/LICENSE-2.0
9   *
10  * Designed and built with all the love in the world by @mdo and @fat.
11  */
12  /*! normalize.css v2.1.0 | MIT License | git.io/normalize */
13  article,
14  aside,
15  details,
16  figcaption,
17  figure,
18  footer,
19  header,
20  hgroup,
21  main,
22  nav,
23  section,
24  summary {
25    display: block;
26  }
27  audio,
28  canvas,
29  video {
30    display: inline-block;
31  }
32  audio:not([controls]) {
33    display: none;
34    height: 0;
35  }
36  [hidden] {
37    display: none;
38  }
39  html {
40    font-family: sans-serif;
41    -webkit-text-size-adjust: 100%;
42    -ms-text-size-adjust: 100%;
43  }
44  body {
45    margin: 0;
46  }
47  a:focus {
48    outline: thin dotted;
49  }

```

Code généré

Un thème pour Bootstrap

Le thème optionnel

La version 3 de Bootstrap est livrée avec un thème optionnel. Voyons ce que nous propose ce thème décrit comme devant améliorer notre « expérience visuelle ». Pour cela, on va le tester sur notre page d'exemple, il suffit de référencer le fichier :

Code : HTML

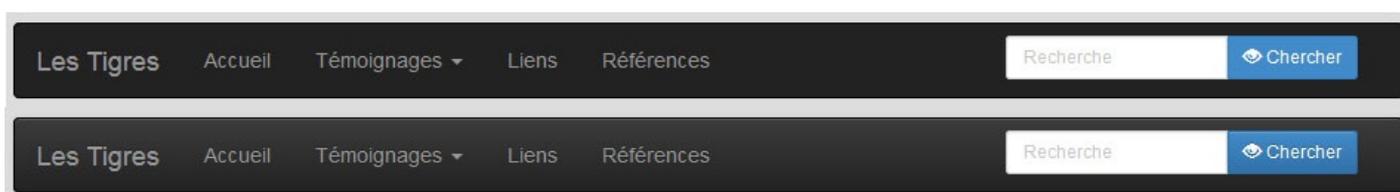
```

<link href="assets/css/bootstrap.css" rel="stylesheet">
<link href="assets/css/bootstrap-theme.min.css" rel="stylesheet">

```

[Tester !](#)

Pour se rendre compte des différences, la figure suivante représente les 2 versions de la barre de navigation :



Barre sans effet et barre avec thème de Bootstrap

On se rend facilement compte que la barre originelle est « plate », alors que la nouvelle a du relief. L'effet est obtenu grâce à un dégradé. Si on regarde dans le code, on trouve effectivement une déclaration de dégradé :

Code : CSS

```
.navbar-inverse {  
  background-image: -webkit-linear-gradient(top, #3c3c3c 0%, #222 100%);  
  background-image: linear-gradient(to bottom, #3c3c3c 0%, #222 100%);  
  filter:  
  progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff3c3c3c', endColorstr='#ff222222', GradientType=0);  
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
  background-repeat: repeat-x;  
}
```

Il en est de même pour le bouton :

Code : CSS

```
.btn-primary {  
  background-image: -webkit-linear-gradient(top, #428bca 0%, #2d6ca2 100%);  
  background-image: linear-gradient(to bottom, #428bca 0%, #2d6ca2 100%);  
  filter:  
  progid:DXImageTransform.Microsoft.gradient(startColorstr='#ff428bca', endColorstr='#ff2d6ca2', GradientType=0);  
  filter: progid:DXImageTransform.Microsoft.gradient(enabled = false);  
  background-repeat: repeat-x;  
  border-color: #2b669a;  
}
```

On retrouve cet effet de dégradé dans les alertes, les barres de progression, les panneaux... Il est prévu également des ombrages. Vous pouvez voir tous ces effets sur [la page d'exemple](#) du site de Bootstrap. On peut se demander pourquoi ces effets ne font pas partie de fichier de base de Bootstrap. La réponse se situe au niveau des performances. Il a été choisi d'éviter les effets trop gourmands en ressources.

D'autres thèmes

On peut trouver des thèmes intéressants et gratuits sur le site [Bootswatch](#). Voyons à la figure suivante ce que donne le thème Amelia sur notre page d'exemple.

Les Tigres Accueil Témoignages + Liens Références Recherche Chercher

Mon amour pour les tigres

Je suis passionné par les **tigres** depuis très longtemps. Ce site a été construit en *hommage à ces merveilleux félins...*
Je fais partie de la [SAT](#) qui a pour but de faire connaître ces splendides animaux.

Voici ce qu'en dit le wikipedia :

Le Tigre (*Panthera tigris*) est un **mammifère** carnivore de la famille des félidés (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands **carnivores** du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement. Superprédateur, il chasse principalement les cerfs et les sangliers, bien qu'il puisse s'attaquer à des proies de taille plus importante comme les buffles. Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle possède un territoire qui englobe les domaines de plusieurs femelles et ne participe pas à l'éducation des petits.

— Wikipedia

Vidéo 1 Vidéo 2 Vidéo 3 Tigre du bengale / La terre du tigre / Document...



Effet du thème Amelia sur la page d'exemple

[Tester](#)

Ce n'est pas mal, mais trop contrasté à mon goût. Mon préféré est Slate, visible à la figure suivante.

Les Tigres
Accueil
Témoignages ▾
Liens
Références

Recherche

Chercher

Mon amour pour les tigres

Je suis passionné par les **tigres** depuis très longtemps. Ce site a été construit en *hommage à ces merveilleux félins...*
Je fais partie de la SAT qui a pour but de faire connaître ces splendides animaux.

Voici ce qu'en dit le wikipedia :

Le Tigre (*Panthera tigris*) est un **mammifère** carnivore de la famille des **félidés** (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands carnivores du monde. L'espèce est divisée en neuf sous-espèces possédant des différences mineures en termes de taille ou de comportement. Superprédateur, il chasse principalement les cerfs et les sangliers, bien qu'il puisse s'attaquer à des proies de taille plus importante comme les buffles. Jusqu'au XIXe siècle, le Tigre était réputé mangeur d'homme. La structure sociale des tigres en fait un animal solitaire ; le mâle possède un territoire qui englobe les domaines de plusieurs femelles et ne participe pas à l'éducation des petits.

— Wikipedia

Vidéo 1

Vidéo 2

Vidéo 3

Un tigre domestique dans son canapé










Effet du thème Slate sur la page d'exemple

[Tester](#)

Composer son thème

Vous pouvez évidemment composer votre thème personnalisé. Ce n'est pas facile, il faut bien comprendre l'organisation des variables LESS, il faut aussi trouver un bon équilibre de couleurs. J'ai trouvé [un site](#) qui facilite ce paramétrage. Les variables y sont mieux organisées que dans [la page de personnalisation](#) de Bootstrap. Vous pouvez aussi aller directement jouer sur les valeurs dans le fichier `variables.less`. Nous allons voir un exemple simple de ce qu'il est possible d'obtenir. Admettons que je désire un site de tonalité verte. Je vais donc changer les gris pour les teinter légèrement en vert :

Code : CSS

```

@gray-darker:          lighten(#232, 13.5%);
@gray-dark:           lighten(#232, 20%);
@gray:                lighten(#232, 33.5%);
@gray-light:          lighten(#232, 60%);
@gray-lighter:        lighten(#232, 93.5%);
```

Je vais aussi modifier les 5 couleurs de base (Brand) pour les rendre plus adaptées à une tonalité verte :

Code : CSS

```
@brand-primary:      #51c1d1;
@brand-success:      #5cb85c;
@brand-warning:      #f0ad4e;
@brand-danger:       #d9534f;
@brand-info:         #5bc0de;
```

Je vais ensuite choisir une nouvelle couleur de fond de page :

Code : CSS

```
@body-bg:            #E0E4CC;
```

Je vais enfin modifier la couleur de bordure des thumbnails pour les rendre visibles :

Code : CSS

```
@thumbnail-border:  #aca;
```

Pour tester tout ça je prends une version un peu adaptée de l'exemple de thème de Bootstrap :

Code : HTML

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">

    <title>Theme Template for Bootstrap</title>

    <!-- Bootstrap core CSS -->
    <link href="assets/css/bootstrap.min.t03.css" rel="stylesheet">

    <!-- Bootstrap theme -->
    <link href="assets/css/bootstrap-theme.min.css"
rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="assets/css/theme.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements
and media queries -->
    <!--[if lt IE 9]>
<script src="../../assets/js/html5shiv.js"></script>
<script src="../../assets/js/respond.min.js"></script>
<![endif]-->
  </head>

  <body>

    <div class="container theme-showcase">

      <!-- Main jumbotron for a primary marketing message or call
to action -->
      <div class="jumbotron">
        <h1>Hello, world!</h1>
        <p>This is a template for a simple marketing or
```

informational website. It includes a large callout called the hero unit and three supporting pieces of content. Use it as a starting point to create something more unique.

```
<p><a class="btn btn-primary btn-lg">Learn more
&raquo;</a></p>
</div>
```

```
<div class="page-header">
  <h1>Buttons</h1>
</div>
<p>
  <button type="button" class="btn btn-lg btn-
default">Default</button>
  <button type="button" class="btn btn-lg btn-
primary">Primary</button>
  <button type="button" class="btn btn-lg btn-
success">Success</button>
  <button type="button" class="btn btn-lg btn-
info">Info</button>
  <button type="button" class="btn btn-lg btn-
warning">Warning</button>
  <button type="button" class="btn btn-lg btn-
danger">Danger</button>
  <button type="button" class="btn btn-lg btn-
link">Link</button>
</p>
<p>
  <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>
</p>
<p>
  <button type="button" class="btn btn-sm btn-
default">Default</button>
  <button type="button" class="btn btn-sm btn-
primary">Primary</button>
  <button type="button" class="btn btn-sm btn-
success">Success</button>
  <button type="button" class="btn btn-sm btn-
info">Info</button>
  <button type="button" class="btn btn-sm btn-
warning">Warning</button>
  <button type="button" class="btn btn-sm btn-
danger">Danger</button>
  <button type="button" class="btn btn-sm btn-
link">Link</button>
</p>
<p>
  <button type="button" class="btn btn-xs btn-
default">Default</button>
  <button type="button" class="btn btn-xs btn-
primary">Primary</button>
  <button type="button" class="btn btn-xs btn-
success">Success</button>
  <button type="button" class="btn btn-xs btn-
info">Info</button>
  <button type="button" class="btn btn-xs btn-
warning">Warning</button>
  <button type="button" class="btn btn-xs btn-
```

```

danger">Danger</button>
  <button type="button" class="btn btn-xs btn-
link">Link</button>
</p>

<div class="page-header">
  <h1>Thumbnails</h1>
</div>


<div class="page-header">
  <h1>Dropdown menus</h1>
</div>
<div class="dropdown theme-dropdown clearfix">
  <a id="dropdownMenu1" href="#" role="button" class="sr-only
dropdown-toggle" data-toggle="dropdown">Dropdown <b
class="caret"></b></a>
  <ul class="dropdown-menu" role="menu" aria-
labelledby="dropdownMenu1">
    <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">Action</a></li>
    <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">Another action</a></li>
    <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">Something else here</a></li>
    <li role="presentation" class="divider"></li>
    <li role="presentation"><a role="menuitem" tabindex="-1"
href="#">Separated link</a></li>
  </ul>
</div>

<div class="page-header">
  <h1>Navbars</h1>
</div>

<div class="navbar navbar-default">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Project name</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-
toggle="dropdown">Dropdown <b class="caret"></b></a>
          <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li><a href="#">Something else here</a></li>
            <li class="divider"></li>
            <li class="dropdown-header">Nav header</li>
            <li><a href="#">Separated link</a></li>
            <li><a href="#">One more separated link</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</div>

```

```

        </li>
      </ul>
      <ul class="nav navbar-nav navbar-right">
        <li><a href="../navbar/">Default</a></li>
        <li><a href="../navbar-static-top/">Static
top</a></li>
        <li class="active"><a href=".">Fixed top</a></li>
      </ul>
    </div><!--/.nav-collapse -->
  </div>
</div>

<div class="navbar navbar-inverse">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Project name</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-
toggle="dropdown">Dropdown <b class="caret"></b></a>
          <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li><a href="#">Something else here</a></li>
            <li class="divider"></li>
            <li class="dropdown-header">Nav header</li>
            <li><a href="#">Separated link</a></li>
            <li><a href="#">One more separated link</a></li>
          </ul>
        </li>
      </ul>
      <ul class="nav navbar-nav navbar-right">
        <li><a href="../navbar/">Default</a></li>
        <li><a href="../navbar-static-top/">Static
top</a></li>
        <li class="active"><a href=".">Fixed top</a></li>
      </ul>
    </div><!--/.nav-collapse -->
  </div>
</div>

<div class="page-header">
  <h1>Alerts</h1>
</div>
<div class="alert alert-success">
  <strong>Well done!</strong> You successfully read this
important alert message.
</div>
<div class="alert alert-info">
  <strong>Heads up!</strong> This alert needs your attention,
but it's not super important.
</div>
<div class="alert alert-warning">
  <strong>Warning!</strong> Best check yo self, you're not
looking too good.
</div>
<div class="alert alert-danger">

```

```

    <strong>Oh snap!</strong> Change a few things up and try
    submitting again.
  </div>

  <div class="page-header">
    <h1>Progress bars</h1>
  </div>
  <div class="progress">
    <div class="progress-bar" role="progressbar" aria-
    valuenow="60" aria-valuemin="0" aria-valuemax="100" style="width:
    60%;">
    <span class="sr-only">60% Complete</span></div>
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-success"
      role="progressbar" aria-valuenow="40" aria-valuemin="0" aria-
      valuemax="100" style="width: 40%;">
      <span class="sr-only">40% Complete (success)</span></div>
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-info"
      role="progressbar" aria-valuenow="20" aria-valuemin="0" aria-
      valuemax="100" style="width: 20%;">
      <span class="sr-only">20% Complete</span></div>
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-warning"
      role="progressbar" aria-valuenow="60" aria-valuemin="0" aria-
      valuemax="100" style="width: 60%;">
      <span class="sr-only">60% Complete (warning)</span></div>
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-danger"
      role="progressbar" aria-valuenow="80" aria-valuemin="0" aria-
      valuemax="100" style="width: 80%;">
      <span class="sr-only">80% Complete (danger)</span></div>
    </div>
    <div class="progress">
      <div class="progress-bar progress-bar-success" style="width:
      35%;"><span class="sr-only">35% Complete (success)</span></div>
      <div class="progress-bar progress-bar-warning" style="width:
      20%;"><span class="sr-only">20% Complete (warning)</span></div>
      <div class="progress-bar progress-bar-danger" style="width:
      10%;"><span class="sr-only">10% Complete (danger)</span></div>
    </div>

  <div class="page-header">
    <h1>List groups</h1>
  </div>
  <div class="row">
    <div class="col-sm-4">
      <ul class="list-group">
        <li class="list-group-item">Cras justo odio</li>
        <li class="list-group-item">Dapibus ac facilisis in</li>
        <li class="list-group-item">Morbi leo risus</li>
        <li class="list-group-item">Porta ac consectetur ac</li>
        <li class="list-group-item">Vestibulum at eros</li>
      </ul>
    </div><!-- /.col-sm-4 -->
    <div class="col-sm-4">
      <div class="list-group">
        <a href="#" class="list-group-item active">
          Cras justo odio
        </a>
        <a href="#" class="list-group-item">Dapibus ac facilisis
in</a>

```

```

        <a href="#" class="list-group-item">Morbi leo risus</a>
        <a href="#" class="list-group-item">Porta ac consectetur
ac</a>
        <a href="#" class="list-group-item">Vestibulum at
eros</a>
    </div>
</div><!-- /.col-sm-4 -->
<div class="col-sm-4">
    <div class="list-group">
        <a href="#" class="list-group-item active">
            <h4 class="list-group-item-heading">List group item
heading</h4>
            <p class="list-group-item-text">Donec id elit non mi
porta gravida at eget metus. Maecenas sed diam eget risus varius
blandit.</p>
        </a>
        <a href="#" class="list-group-item">
            <h4 class="list-group-item-heading">List group item
heading</h4>
            <p class="list-group-item-text">Donec id elit non mi
porta gravida at eget metus. Maecenas sed diam eget risus varius
blandit.</p>
        </a>
        <a href="#" class="list-group-item">
            <h4 class="list-group-item-heading">List group item
heading</h4>
            <p class="list-group-item-text">Donec id elit non mi
porta gravida at eget metus. Maecenas sed diam eget risus varius
blandit.</p>
        </a>
    </div>
</div><!-- /.col-sm-4 -->
</div>

```

```

<div class="page-header">
    <h1>Panels</h1>
</div>
<div class="row">
    <div class="col-sm-4">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h3 class="panel-title">Panel title</h3>
            </div>
            <div class="panel-body">
                Panel content
            </div>
        </div>
        <div class="panel panel-primary">
            <div class="panel-heading">
                <h3 class="panel-title">Panel title</h3>
            </div>
            <div class="panel-body">
                Panel content
            </div>
        </div>
    </div><!-- /.col-sm-4 -->
    <div class="col-sm-4">
        <div class="panel panel-success">
            <div class="panel-heading">
                <h3 class="panel-title">Panel title</h3>
            </div>
            <div class="panel-body">
                Panel content
            </div>
        </div>
        <div class="panel panel-info">
            <div class="panel-heading">
                <h3 class="panel-title">Panel title</h3>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="panel-body">
            Panel content
        </div>
    </div>
</div><!-- /.col-sm-4 -->
<div class="col-sm-4">
    <div class="panel panel-warning">
        <div class="panel-heading">
            <h3 class="panel-title">Panel title</h3>
        </div>
        <div class="panel-body">
            Panel content
        </div>
    </div>
    <div class="panel panel-danger">
        <div class="panel-heading">
            <h3 class="panel-title">Panel title</h3>
        </div>
        <div class="panel-body">
            Panel content
        </div>
    </div>
</div><!-- /.col-sm-4 -->
</div>

<div class="page-header">
    <h1>Wells</h1>
</div>
<div class="well">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    Maecenas sed diam eget risus varius blandit sit amet non magna.
    Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent
    commodo cursus magna, vel scelerisque nisl consectetur et. Cras
    mattis consectetur purus sit amet fermentum. Duis mollis, est non
    commodo luctus, nisi erat porttitor ligula, eget lacinia odio sem
    nec elit. Aenean lacinia bibendum nulla sed consectetur.</p>
</div>

</div> <!-- /container -->

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster
-->
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>
</html>

```

Et j'obtiens la figure suivante.

Buttons



Thumbnails



Thème personnalisé

[Tester !](#)

Il ne s'agit évidemment ici que d'une modification sommaire de Bootstrap mais elle vous montre la démarche à adopter. Si vous avez du mal à composer une palette colorée pour vos pages, vous pouvez trouver des inspirations sur le site communautaire [Colour Lover](#).

En résumé :

- On peut générer une version personnalisée de Bootstrap à partir de l'interface proposée en ligne en modifiant des paramètres.
- Bootstrap utilise LESS, un langage qui simplifie et organise l'écriture des fichiers CSS. On peut créer une version adaptée de Bootstrap en modifiant directement les fichiers LESS.
- Bootstrap propose un thème optionnel. Il est aussi possible de créer son propre thème.

Voilà, vous êtes arrivés au bout de ce cours. 😊 Je suis conscient de n'avoir pas épuisé le sujet. D'autre part ce framework est appelé à évoluer rapidement et j'essaierai de maintenir mon cours à jours.

J'espère que ce cours vous aura permis de découvrir ce framework et donné envie de l'utiliser 😊