



Bootstrap de Twitter : un kit CSS et plus !

Par bestmomo



*Licence Creative Commons 6 2.0
Dernière mise à jour le 12/10/2012*

Sommaire

Sommaire	2
Lire aussi	2
Bootstrap de Twitter : un kit CSS et plus !	4
Partie 1 : Premiers pas	5
Mise en route	5
Un framework ?	5
Découverte de Bootstrap	6
Origine de Bootstrap	6
Contenu du kit	6
Installation	6
Une trame pour démarrer	8
Une grille ?	10
Présentation	11
Le principe de la grille	11
Organisation de la grille	11
La grille fixe	13
Une seule rangée	13
Plusieurs rangées	13
Sauter des colonnes	14
Imbrication d'éléments	15
Un petit TP	16
La grille élastique	17
Un peu d'élasticité	17
Mise en page	20
Mise en page fixe	20
Mise en page élastique	21
Un peu de pratique	23
Un exemple de page	23
Un petit TP	27
Partie 2 : Mise en forme des éléments	28
Eléments de base	29
Listes et tableaux	29
Les listes	31
Les tableaux	32
Formulaires	36
Boutons, icônes et images	51
Un petit TP	58
Les composants intégrés	59
La navigation	60
Une barre de navigation	60
La pagination	65
Effets typographiques	67
Hero Unit	67
Des libellés colorés	69
Des badges colorés	69
Thumbnails	69
Alertes et barres de progression	72
Alertes	72
Barres de progression	74
Partie 3 : Mise en forme avancée	77
Les plugins jQuery	77
Menu déroulant	77
Menu déroulant dans la barre de navigation	77
Menu déroulant dans un bouton	78
Sous-menu	80
Fenêtre modale	81
Onglets	84
Carousel	87
Info-bulles et Popover	92
Info-bulle	92
Popover	93
Boutons	96
Bouton bascule	96
Boutons effet "checkbox"	96
Boutons effet "radio"	97
Boutons pour attente de processus	97
Typeahead	99
Effet accordéon	100
Un exemple de réalisation	105
Le résultat final à l'œuvre	105
Organisation de la page	107
La barre de menu	109
Les barres de boutons et les canvas	110
Le pied de page	111

Les pages modales	112
Configurer Bootstrap	113
Configuration simplifiée	113
Configurer en ligne sur le site de Bootstrap	113
Configurer en ligne sur le site Stylebootstrap	115
Varier les styles	115
Un générateur de thèmes	118
Utilisation de LESS	118
C'est quoi LESS ?	119
Exemple simple de configuration	121
Utilisation dynamique de LESS	125
Partie 4 : Bootstrap et son environnement	127
Le responsive design	127
Présentation	127
Présentation	127
Un exemple sans Bootstrap	128
Un exemple avec Bootstrap	136
Plugins et extensions	142
Plugins	143
Bootstrap Image Gallery	143
Colorpicker and Datepicker for Twitter Bootstrap	147
Extensions	154
De jolis boutons	154
Des icônes vectorielles	157
Kickstrap	160
Intégration à divers environnements	162
jQuery UI Bootstrap	162
jQuery-Mobile-Bootstrap-Theme	163
Django-bootstrap-toolkit	163
Thèmes pour Wordpress	163
Thème pour Joomla	164
Extension pour Typo	165
Extension pour Yii	165
Un template pour Photoshop	165
Une extension pour Fireworks	166
Une boîte à outils pour facebook	166

Bootstrap de Twitter : un kit CSS et plus !

Par [bestmomo](#)

Mise à jour : 12/10/2012

Difficulté : Intermédiaire  Durée d'étude : 20 jours



3 848 visites depuis 7 jours, classé 40/798

Vous créez des pages web et vous passez beaucoup 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 outil. **Bootstrap** est un framework qui peut rendre votre vie de 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 prêt ? Bon on y va mais il serait bien pour profiter pleinement de ce cours de posséder quelques connaissances :

- La base indispensable est de bonnes notions en **HTML** et **CSS**. Vous avez des lacunes ? Comblez-les avec le cours de [Mateo](#)
- 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 [ici](#).
- 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 ? C'est [ici](#).



Ce tutoriel a été mis à jour et complété pour la version 2.1.

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**. Nous allons 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 un cas pratique 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.

Un framework ?



Un framework, c'est quoi ?

C'est 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 les langages Java, Python, Ruby, PHP...

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 des pages web : organisation, aspect, animation...

Ils semblent devenus à la mode et il en existe un certain nombre :

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



Quel en est l'intérêt ?

- Les navigateurs sont pleins de fantaisies et ont des comportements très différents, surtout pour les anciennes versions. Les frameworks sont cross-browser et on a la garantie d'une présentation 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 offrent souvent des éléments complémentaires : boutons esthétiques, barres de navigation...
- La grande diffusion de nouveaux moyens de visualisation du web (iPod, iPad...) impose désormais la prise en compte de tailles d'écran très variées, les frameworks CSS prennent généralement en compte cette contrainte.



Et 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 (surtout avec un joli tuto comme celui-ci ) 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 publié en août 2011 pour répondre à des besoins internes de développement de cette entreprise. Il a été mis à disposition du public sous licence Apache. La version 2.1 est sortie en août 2012.

Contenu du kit

Bootstrap propose :

- une mise en page basée sur une **grille** de 12 colonnes. Vous avez donc une grille à disposition bien pratique. Bien sûr, si vous avez besoin de plus de 12 colonnes, ou de moins, il est toujours possible de modifier la configuration.
- un **reset CSS** basé sur [Normalize.css](#)
- du code fondé sur **HTML5 et CSS 3**
- 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**, en prenant même en compte **IE7**, donc une garantie de compatibilité maximale
- une bonne documentation sur le site de Twitter
- la garantie d'une évolution permanente
- 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 **reset 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. Le reset consiste à mettre à zéro toutes les valeurs pour ainsi uniformiser le rendu. Le reset de **Bootstrap** est basé sur [celui d'Eric Meyer](#), allégé et adapté.

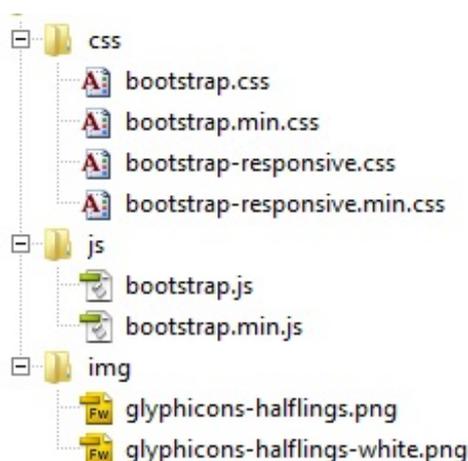
Installation

L'installation de Bootstrap est simple : cliquez sur le bouton de téléchargement [sur le site de Twitter](#).



Mise à jour avec la version 2.1

Quand vous téléchargez la librairie vous obtenez un fichier zippé contenant un répertoire **bootstrap** avec 3 sous-répertoires contenant quelques fichiers :



Voilà 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** sans les commentaires ;
- **bootstrap-responsive.css** comporte les classes de base pour le responsive design ;
- **bootstrap-responsive.min.css** comporte les mêmes classes de base que **bootstrap-responsive.css** sans les commentaires ;
- **glyphicons-halflings.png** comporte une collection d'icônes fournies par [glyphicons](#) ;
- **glyphicons-halflings-white.png** comporte la même collection d'icônes mais blanche ;
- **bootstrap.js** contient des fonctions JavaScript des composants de Bootstrap ;
- **bootstrap.min.js** contient les mêmes fonctions JavaScript sans les commentaires. C'est le fichier à utiliser en production.

Les fichiers "min" (**bootstrap.min.css** et **bootstrap-responsive.min.css**) contiennent le même code que leurs équivalents (**bootstrap.css** et **bootstrap-responsive.css**) mais ont été épurés des commentaires et ainsi allégés pour accélérer leur chargement.

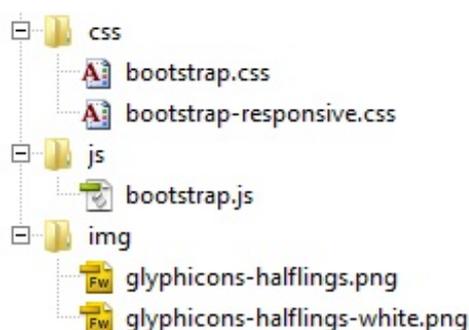


D'accord, mais que mettre sur un site pour que **Bootstrap** fonctionne ?

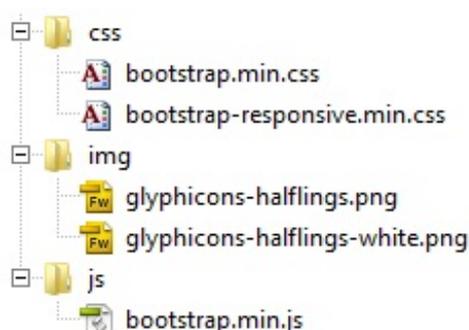
Il faut distinguer les deux situations :

- Pendant la phase de développement, il est intéressant de pouvoir naviguer dans le code de **Bootstrap**, il faut donc mettre les fichiers **CSS** non compressés ;
- 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").

Voilà les fichiers utiles pendant la phase de développement (le fichier **bootstrap-responsive.css** n'est à prévoir que si vous avez besoin du **responsive design**, le fichier **bootstrap.js** n'est à prévoir que si vous utilisez des plugins **jQuery**) :



Voilà les fichiers utiles lorsque le site est en ligne (le fichier **bootstrap-responsive.min.css** n'est à prévoir que si vous avez besoin du **responsive design**, le fichier **bootstrap.min.js** n'est à prévoir que si vous utilisez des plugins **jQuery**) :



Dans tous les exemples de ce tuto je pars du principe qu'un répertoire **bootstrap** a été créé à la racine du site avec les trois répertoires **css**, **img** et **js** comportant les fichiers cités ci-dessus.

Pour que **bootstrap** fonctionne il faut le déclarer dans les pages **HTML** qui doivent d'ailleurs être impérativement au format **HTML5**, donc :

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"
type="text/css">
</head>
```

Si votre application est destinée à être utilisée sur des supports variés (tablettes, iPhones...) vous devez aussi déclarer le fichier **bootstrap-responsive.min.css** (ou **bootstrap-responsive.css**) dans l'en-tête de la page web :

Code : HTML

```
<head>
...
<link href="bootstrap/css/bootstrap-responsive.min.css"
rel="stylesheet" type="text/css">
</head>
```

A partir de là toutes les classes sont accessibles... Évidemment vous devez adapter le lien selon la localisation de vos fichiers, c'est le cas d'ailleurs sur mon site de démonstration où vous trouverez le lien ainsi libellé :

Code : HTML

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

Pour remonter dans l'arbre des répertoires.

Une trame pour démarrer

Vous trouvez aussi sur le site une [page de démarrage](#) 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.

Vous y trouvez les appels aux fichiers CSS :

Code : HTML

```
<!-- Le styles -->
<link href="../../assets/css/bootstrap.css" rel="stylesheet">
<style>
  body {
    padding-top: 60px; /* 60px to make the container go all the
way to the bottom of the topbar */
  }
</style>
<link href="../../assets/css/bootstrap-responsive.css"
rel="stylesheet">
```

Il faut évidemment adapter les liens à votre situation. Nous verrons l'utilité du style ajouté lors de l'examen des barres de navigation.

Je rappelle que le fichier **bootstrap-responsive.css** est utile si votre application doit s'adapter à une variété de tailles d'écrans. Je développerai cette notion un peu délicate dans un chapitre ultérieur.

On trouve ensuite les icônes pour les appareils d'Apple (iPhone, iPod Touch et iPad):

Code : HTML

```
<!-- Le fav and touch icons -->
<link rel="shortcut icon" href="../assets/ico/favicon.ico">
<link rel="apple-touch-icon-precomposed" sizes="114x114"
href="../assets/ico/apple-touch-icon-114-precomposed.png">
<link rel="apple-touch-icon-precomposed" sizes="72x72"
href="../assets/ico/apple-touch-icon-72-precomposed.png">
<link rel="apple-touch-icon-precomposed"
href="../assets/ico/apple-touch-icon-57-precomposed.png">
```

A adapter évidemment avec vos icônes personnelles 😊. Je rappelle que ces icônes sont pertinentes pour qu'il n'y ait pas une création automatique d'une icône à partir d'une capture d'écran de votre site.

Une trame de la barre de navigation (je vous explique tout ça en détail dans ce cours) :

Code : HTML

```
<div class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container">
      <a class="btn btn-navbar" data-toggle="collapse" data-
target=".nav-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a>
      <a class="brand" href="#">Project name</a>
      <div class="nav-collapse">
        <ul class="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>
```

Et en fin de page les appels Javascript (utiles si vous utilisez des plugins **jQuery**, comme nous allons le voir dans ce cours):

Code : HTML

```
<!-- Le 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-transition.js"></script>
<script src="../assets/js/bootstrap-alert.js"></script>
<script src="../assets/js/bootstrap-modal.js"></script>
<script src="../assets/js/bootstrap-dropdown.js"></script>
<script src="../assets/js/bootstrap-scrollspy.js"></script>
<script src="../assets/js/bootstrap-tab.js"></script>
<script src="../assets/js/bootstrap-tooltip.js"></script>
<script src="../assets/js/bootstrap-popover.js"></script>
<script src="../assets/js/bootstrap-button.js"></script>
```

```
<script src="../../assets/js/bootstrap-collapse.js"></script>
<script src="../../assets/js/bootstrap-carousel.js"></script>
<script src="../../assets/js/bootstrap-typeahead.js"></script>
```

Là aussi il ne faut conserver que les appels utiles et adapter les liens.



Ces fichiers ne se trouvent pas dans le paquetage de **Bootstrap**, on les trouve où ?

Le fichier **jquery.js** peut être téléchargé sur [le site de jQuery](#). Les autres fichiers correspondent aux plugins **jQuery** de **Bootstrap**. Tout le code se trouve également dans le fichier **bootstrap.js** (ou **bootstrap.min.js**) qui se trouve lui dans le paquetage. Il y a là une petite incohérence dans la documentation de **Bootstrap**. Vous avez donc le choix entre utiliser le fichier **bootstrap.js** (ou **bootstrap.min.js**) pour avoir tous les plugins disponibles avec une seule référence comme je vous le propose dans le chapitre correspondant ou récupérer individuellement les fichiers.

Ainsi donc si vous optez pour le fichier contenant tous les plugins, ce que je vous conseille, le code devient plus simple :

Code : HTML

```
<!-- Le 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>
```

Bon maintenant vous savez tout pour démarrer avec **Bootstrap**, on va pouvoir commencer les choses intéressantes 😊

📄 Une grille ?

L'organisation spatiale des pages web est toujours la première préoccupation lorsqu'on crée un site. Est-ce qu'on prévoit 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 par contre 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 de la grille

La grille est tout simplement un découpage en cellules de mêmes dimensions :

On peut alors décider d'organiser du contenu en utilisant pour chaque élément une ou plusieurs cellules :

	Élément 1										
				Élément 2							

La grille de **Bootstrap** comporte 12 colonnes comme dans l'illustration ci-dessus. Vous commencez sans doute à comprendre l'utilité de cette organisation 😊

Organisation de la grille

Bootstrap est essentiellement un fichier **CSS**. Il comporte de très 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 s'arrangeant pour en avoir 12 en tout. Pour définir le nombre de colonnes utilisées pour chaque élément on dispose d'une batterie de 12 classes :

- span1
- span2
- span3
- span4
- span5
- span6
- span7
- span8
- span9
- span10
- span11
- span12

Ces classes sont très simples, elles héritent d'une définition commune :

Code : CSS

```
[class*="span"] {
  float: left;
  margin-left: 20px;
}
```



Il y a donc une marge de 20px entre les colonnes. On trouve aussi une fixation individuelle de la largeur, par exemple :

Code : CSS

```
.span4 {
  width: 300px;
}
```



Pourquoi 300px ?

Si vous regardez la largeur pour le "span1" vous trouvez la valeur 60. On pourrait donc s'attendre à une valeur de $4 * 60 = 240$ pour le "span4", mais il ne faut pas oublier les 3 marges intermédiaires $3 * 20 = 60$, au final $240 + 60 = 300$ 😊

On a dit que la grille complète fait 940px, hors si chaque colonne avec sa marge occupe 80px on a en tout $80 * 12 = 960$ px. D'où vient cette différence de 20px ? Tout simplement de la marge gauche de la première colonne qui est inutile. Cette différence est gérée par la classe **row** :

Code : CSS

```
.row {
  margin-left: -20px;
}
```



Un petit exemple ?

Pour avoir un élément de 4 colonnes de large accouplé avec un élément de 8 colonnes de large on a :

Code : HTML

```
<div class="row">
  <div class="span4">4 colonnes</div>
  <div class="span8">8 colonnes</div>
</div>
```

Voici une illustration de ce code :



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 tuto dans un fichier **CSS** particulier :

Code : CSS - css/tuto 1.css

```
[class*="span"] {
  background-color: lightgreen;
  margin-bottom: 10px;
  border-radius: 5px 5px 5px 5px;
  line-height: 80px;
  text-align: center;
}
[class="span10"] {
  background-color: green;
}
```

Le seul but de ces règles de style est de faire apparaître nettement les éléments à l'écran et tant qu'à faire de façon élégante...

La grille fixe

Une seule rangée

Dans ce premier exemple, on va déclarer une seule rangée avec deux éléments :

Code : HTML- tuto01_01

```
<!DOCTYPE HTML>
<html>
<head>
  <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
  type="text/css">
  <link href="css/tuto1.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="row">
<div class="span4">4 colonnes</div>
<div class="span8">8 colonnes</div>
</div>
</body>
</html>
```



Mise en page rapide et simple non ? Cliquez sur l'image pour voir l'effet réel. Il en sera de même pour presque toutes les illustrations de ce tutoriel.

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- tuto 01_02

```
<!DOCTYPE HTML>
```

```
<html>
<head>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="css/tuto1.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="row">
  <div class="span1">1 col</div>
  <div class="span2">2 colonnes</div>
  <div class="span3">3 colonnes</div>
  <div class="span6">6 colonnes</div>
</div>
<div class="row">
  <div class="span12">12 colonnes</div>
</div>
<div class="row">
  <div class="span4">4 colonnes</div>
  <div class="span8">8 colonnes</div>
</div>
</body>
</html>
```



Sauter des colonnes

Bootstrap permet aussi de sauter des colonnes :

Il y a quelques classes d'offset pour ça :

- offset1
- offset2
- offset3
- offset4
- offset5
- offset6
- offset7
- offset8
- offset9
- offset10
- offset11

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

Code : CSS



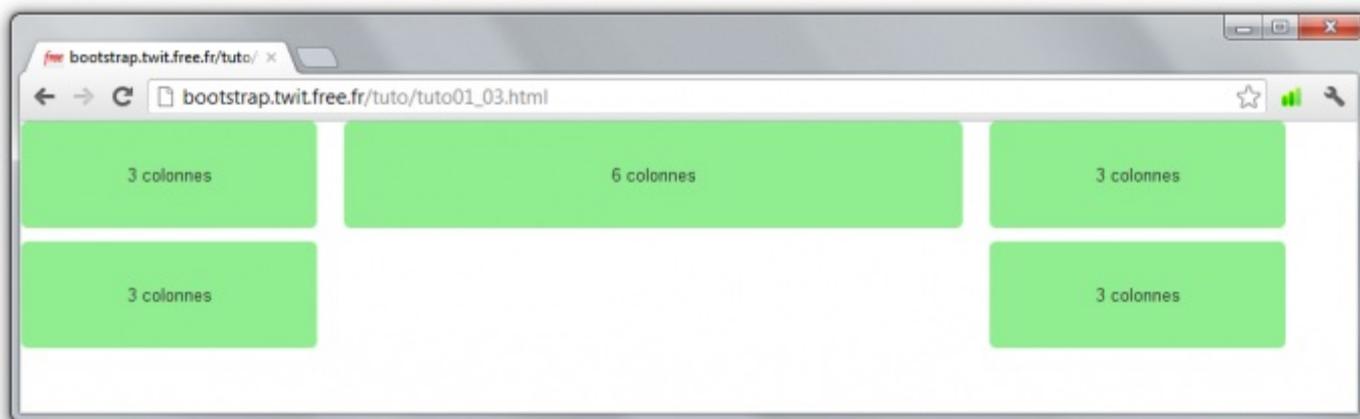
```
.offset6 {
  margin-left: 500px;
}
```

Dans ce cas 6 colonnes à 60px + 7 marges gauches à 20px 😊

Voici un exemple :

Code : HTML - tuto 01_03

```
<!DOCTYPE HTML>
<html>
<head>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="css/tuto1.css" rel="stylesheet" type="text/css">
</head>
<body>
  <div class="row">
    <div class="span3">3 colonnes</div>
    <div class="span6">6 colonnes</div>
    <div class="span3">3 colonnes</div>
  </div>
  <div class="row">
    <div class="span3">3 colonnes</div>
    <div class="span3 offset6">3 colonnes</div>
  </div>
</body>
</html>
```



On a sauté 6 colonnes grâce à la classe **offset6**.

Imbrication d'éléments

On doit souvent imbriquer des éléments dans une page web, est-ce possible avec **Bootstrap** ?

Autrement dit d'inclure un **row** dans un **span**. Et bien on va tester ça tout de suite :

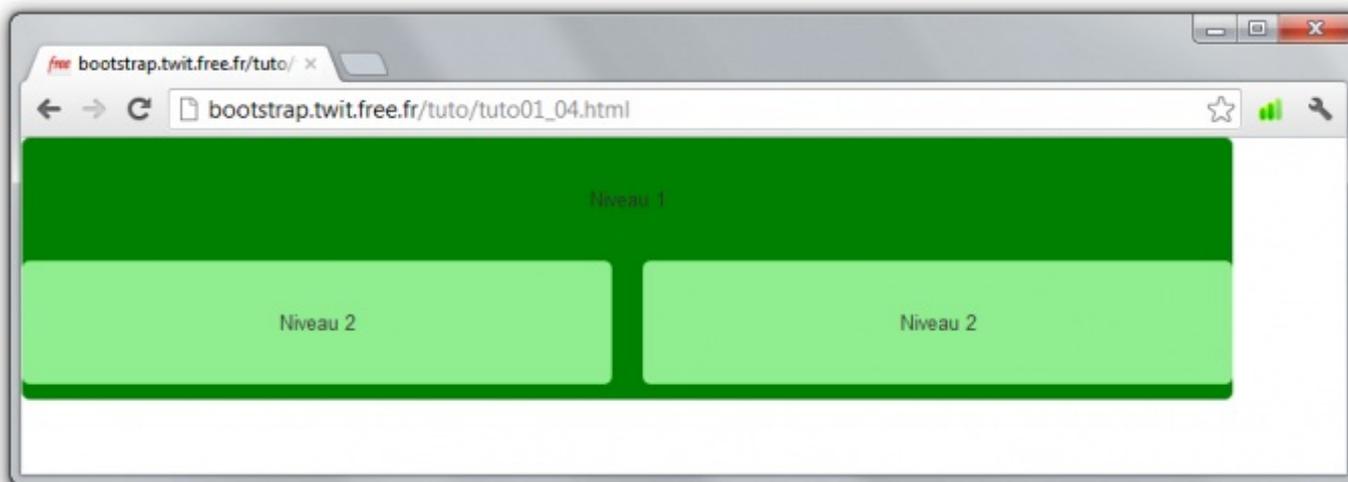
Code : HTML - tuto 01_04

```
<!DOCTYPE HTML>
```

```

<html>
<head>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="css/tuto1.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="row">
  <div class="span10">Niveau 1
  <div class="row">
    <div class="span5">Niveau 2</div>
    <div class="span5">Niveau 2</div>
  </div>
</div>
</body>
</html>

```



On dirait bien que ça fonctionne 😊

Un petit TP

Pour vous entraîner, je vous propose d'obtenir le résultat suivant :



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

Voici la solution :

Secret (cliquez pour afficher)

Code : HTML

```

<div class="row">
  <div class="span6">Niveau 1

```

```

    <div class="row">
      <div class="span2">Niveau 2</div>
      <div class="span4">Niveau 2</div>
    </div>
  </div>
  <div class="span6">Niveau 1
    <div class="row">
      <div class="span2">Niveau 2</div>
      <div class="span2 offset2">Niveau 2</div>
    </div>
  </div>
</div>

```

La grille élastique Un peu d'élasticité

Dans le chapitre précédent, nous avons vu comment utiliser une grille fixe, chaque colonne ayant une largeur fixée en pixels. Il existe une autre possibilité dans laquelle les largeurs ne sont plus établies en pixels mais en pourcentages.

Reprenons le code du "tuto 01_01" de la partie précédente. Voici ce que donne l'affichage :



La largeur de la fenêtre étant plus grande que celle d'un **row** on constate juste l'apparition d'une marge à droite. Que se passe-t-il si on réduit la largeur de la fenêtre ?



Les colonnes ayant des largeurs fixes, le deuxième élément passe sous le premier puisqu'il n'a plus suffisamment de place. Maintenant, voyons ce que ça donne avec une grille élastique. La seule modification à effectuer est au niveau de la classe **row** qui devient **row-fluid**.

Voici la classe "row-fluid" :
Code : CSS

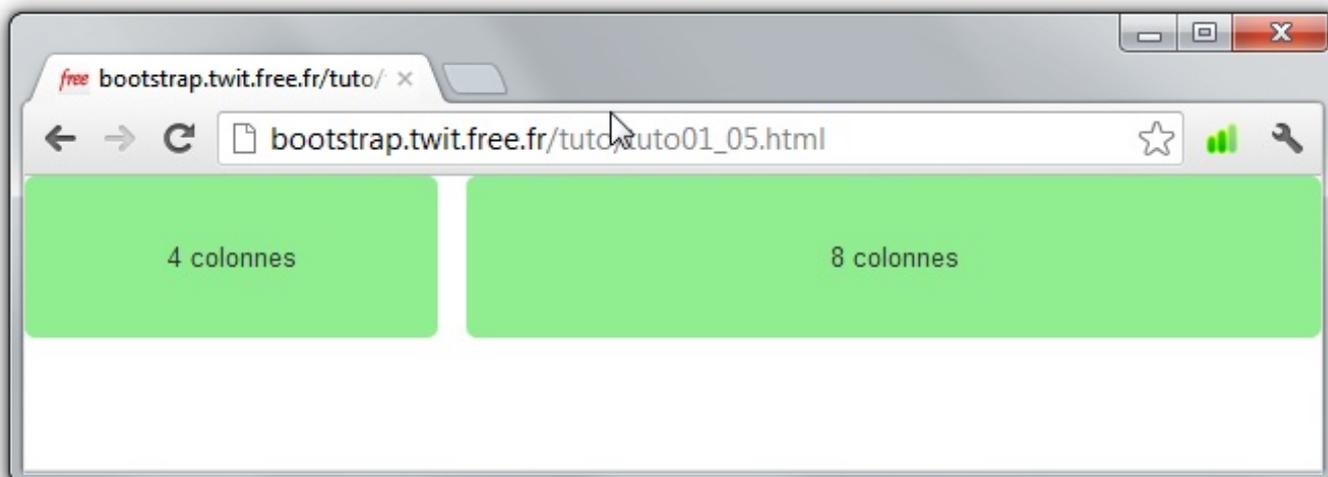


```
.row-fluid {  
  width: 100%;  
}
```

On a donc une occupation totale du contenant.

Code : HTML- tuto 01_05

```
<!DOCTYPE HTML>  
<html>  
<head>  
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"  
type="text/css">  
<link href="css/tuto1.css" rel="stylesheet" type="text/css">  
</head>  
<body>  
<div class="row-fluid">  
  <div class="span4">4 colonnes</div>  
  <div class="span8">8 colonnes</div>  
</div>  
</body>  
</html>
```



Cette fois les éléments occupent toute la largeur de la fenêtre dans tous les cas étant donné qu'ils sont affectés en pourcentage.



Mais on a vu que les classes **span** définissent une largeur en pixel, comment sont-elles maintenant définies en pourcentage ?

Regardez cette classe :

Code : CSS

```
.row-fluid .span4 {  
  width: 31.914893617021278%;  
}
```

Hormis cette élasticité on conserve les mêmes principes qu'avec une grille fixe.



Il y a toutefois une petite question qu'on peut se poser : que se passe-t-il si on introduit moins de colonnes dans le contenu que dans le contenant dans cette version élastique ?

On va tester...

Code : HTML- tuto 01_06

```
<!DOCTYPE HTML>
<html>
<head>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="css/tuto1.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="row">
  <div class="span10">Niveau 1
    <div class="row">
      <div class="span5">Niveau 2</div>
      <div class="span2">Niveau 2</div>
    </div>
  </div>
</div>
</body>
</html>
```



De façon très logique les éléments occupent la totalité des colonnes qui leurs sont assignées et rien de plus...



La version 2.1 a vu l'apparition des offsets aussi pour les grilles élastiques

Ainsi on peut désormais faire ce genre de composition en grille élastique :

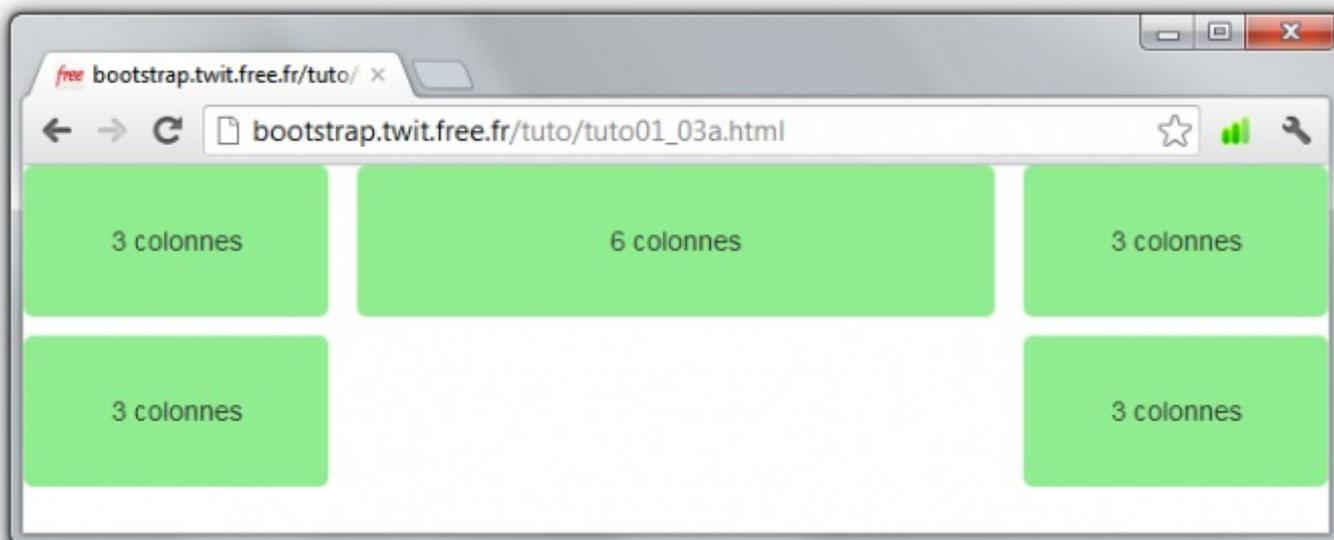
Code : HTML

```
<!DOCTYPE HTML>
<html>
<head>
<link href="../../bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="../../css/tuto1.css" rel="stylesheet" type="text/css">
</head>
<body>
  <div class="row-fluid">
    <div class="span3">3 colonnes</div>
```

```

    <div class="span6">6 colonnes</div>
    <div class="span3">3 colonnes</div>
  </div>
  <div class="row-fluid">
    <div class="span3">3 colonnes</div>
    <div class="span3 offset6">3 colonnes</div>
  </div>
</body>
</html>

```



Mise en page

L'intérêt principal d'une grille est de réaliser une mise en page. Nous allons voir les deux façons de réaliser cela avec **Bootstrap**, avec toujours la même simplicité de mise en œuvre.

Mise en page fixe

La classe englobante pour réaliser une mise en page est **container** :

Code : HTML

```

<body>
<div class="container">
  ...
</div>
</body>

```

Voici un premier exemple :

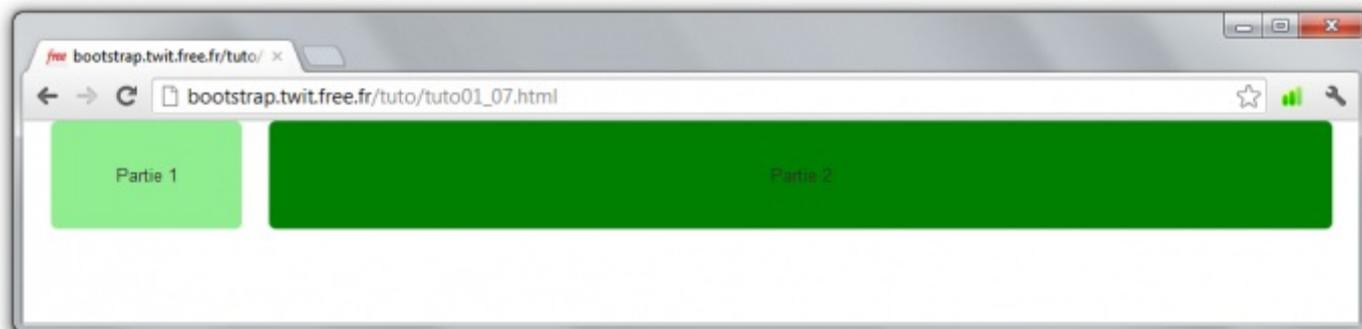
Code : HTML - tuto 01_07

```

<!DOCTYPE HTML>
<html>
<head>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="css/tuto1.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="container">

```

```
<div class="row">
  <div class="span2">Partie 1</div>
  <div class="span10">Partie 2</div>
</div>
</body>
</html>
```



On constate un centrage de la grille utilisée.

Il est intéressant de jeter un œil sur la classe **container** :

Code : CSS



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

La mise en "auto" des marges gauche et droite provoque le centrage.

Mise en page élastique

Pour une mise en page élastique le principe est le même, mais la classe à utiliser est **container-fluid** :

Code : HTML

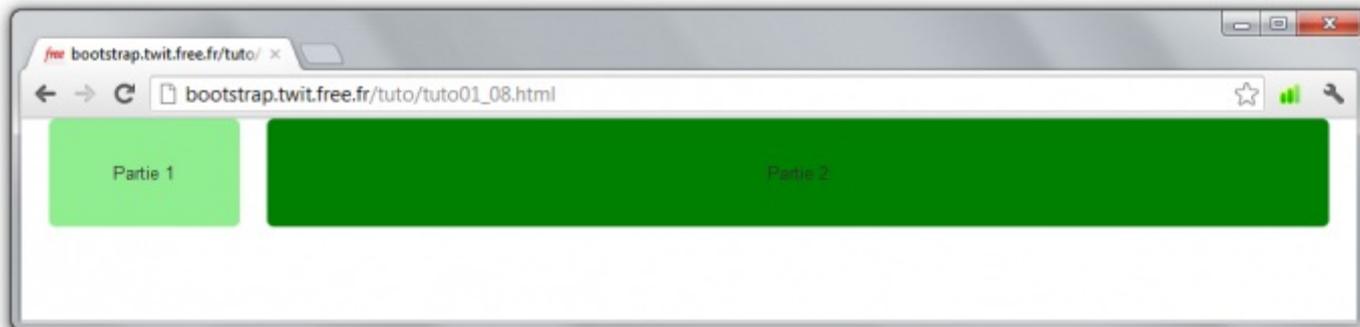
```
<body>
<div class="container-fluid">
  ...
</div>
</body>
```

Voici un exemple simple :

Code : HTML - tuto 01_08

```
<!DOCTYPE HTML>
<html>
<head>
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="css/tuto1.css" rel="stylesheet" type="text/css">
```

```
</head>
<body>
<div class="container-fluid">
  <div class="row-fluid">
    <div class="span2">Partie 1</div>
    <div class="span10">Partie 2</div>
  </div>
</div>
</body>
</html>
```



Maintenant on peut redimensionner la fenêtre, on a une adaptation des éléments.

Voyons un peu cette classe **container-fluid** :

Code : CSS



```
.container-fluid {
  padding-right: 20px;
  padding-left: 20px;
}
```

Un petit décalage du contenu de 20px des deux côtés...

Maintenant plus rien de la mise en page avec **Bootstrap** ne vous échappe alors nous allons passer à un exemple pratique...

Un peu de pratique

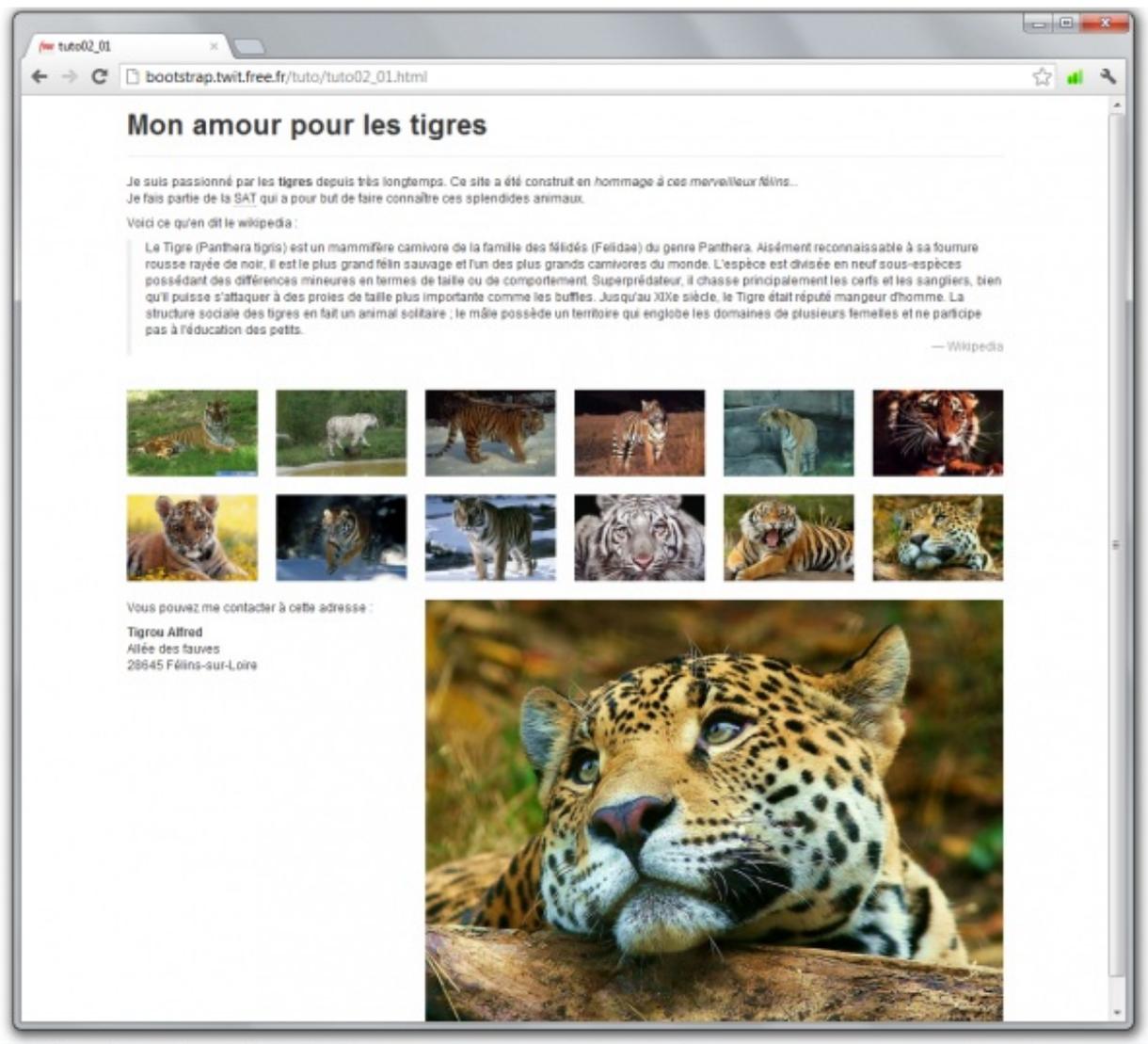
Il est temps maintenant de mettre en application ce que je vous ai expliqué précédemment 😊

Un exemple de page

Voici le premier jet d'une page web de test que je vais faire évoluer progressivement, pour le moment avec ce template :

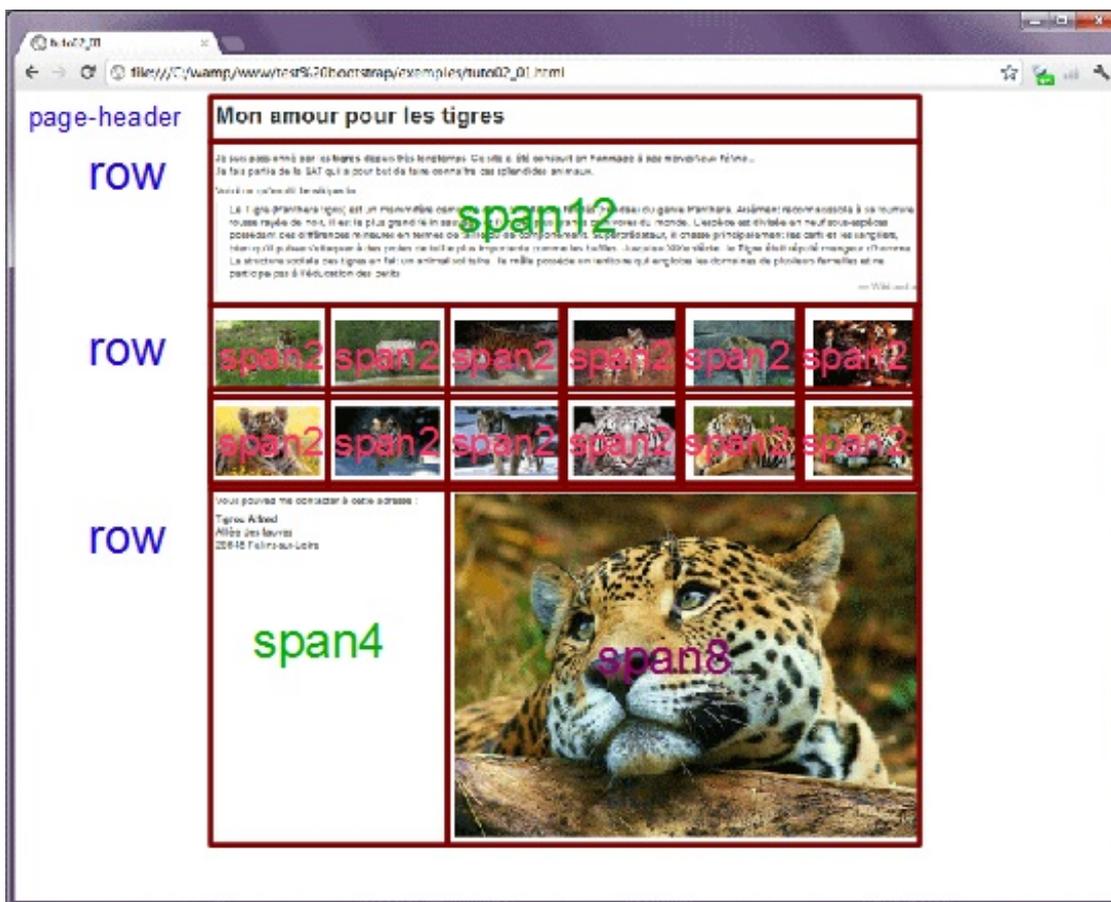
Code : HTML

```
<div class="container">
  <div class="page-header">...</div>
  <div class="row">
    <div class="span12"> ... </div>
  </div>
  <div class="row">
    <div class="span2">...</div>
    <div class="span2">...</div>
  </div>
  <div class="row">
    <div class="span4">...</div>
    <div class="span8">...</div>
  </div>
</div>
```



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 par contre en comporte 12 d'une largeur de 2 colonnes. Si on calcule bien ça donne une largeur de 24 colonnes, le résultat est que les éléments se répartissent en fait sur deux niveaux à l'écran comme on peut le constater. La dernière rangée enfin comporte deux éléments.

Voici une visualisation du découpage :



La classe page-header

Cette classe fixe quelques valeurs :

Code : CSS - classe page-header

```
.page-header {
  border-bottom: 1px solid #EEEEEE;
  margin: 18px 0;
  padding-bottom: 17px;
}
```

Une marge gauche de 18 pixels, une ligne inférieure de 1px de couleur grise avec un écart du contenu par rapport à cette ligne de 17px. 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

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

Mon amour pour les tigres

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 la qualité du popup lorsqu'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

Une citation

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

Code : HTML

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

Le résultat est sobre et élégant :

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 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;
}
```

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 Félines-sur-Loire<br>
</address>
```

Pour un résultat vraiment satisfaisant :

Vous pouvez me contacter à cette adresse :

Tigrou Alfred
Allée des fauves
28645 Félines-sur-Loire

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 **span** :

Code : CSS

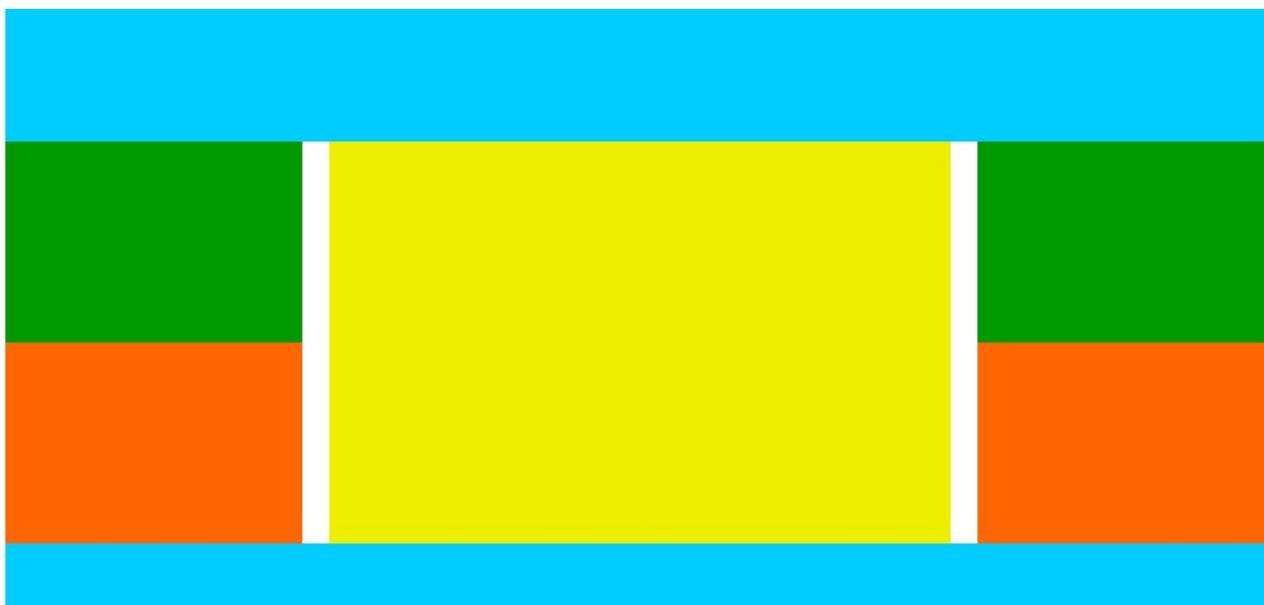
```
[class*="span"] {
  margin-bottom: 20px;
}
```

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

Un petit TP

Maintenant un petit TP pour voir si vous avez compris tout ce que j'ai raconté dans ce chapitre.

Le but est d'obtenir une mise en page fixe de ce type :



Donc un bandeau supérieur, une zone de contenu centrale, deux zones latérales comportant chacune deux blocs superposés et un pied de page.

Réfléchissez quand même un peu avant de regarder ma solution 🤖

Secret (cliquez pour afficher)

Code : HTML

```
<head>
<meta charset="utf-8">
<link href="../../bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<style>
/* Styles utiles à la mise en page */
header {
    height: 100px;
}
section {
    height: 300px;
}
footer {
    height: 50px;
}
.bloc {
    height: 150px;
}
/* Les styles suivants sont juste pour faire apparaître la mise
en page */
.span12 {
    background-color:#0CF;
}
.span6 {
    background-color:#EE0;
}
[class="span3 bloc"] {
    background-color:#F60;
}
[class="span3 bloc"]:first-child {
    background-color:#090;
}
</style>
</head>
<body>
<div class="container">
    <div class="row">
        <header class="span12"></header>
        <aside class="span3">
            <div class="row">
                <div class="span3 bloc"></div>
                <div class="span3 bloc"></div>
            </div>
        </aside>
        <section class="span6"></section>
        <aside class="span3">
            <div class="row">
                <div class="span3 bloc"></div>
                <div class="span3 bloc"></div>
            </div>
        </aside>
        <footer class="span12"></footer>
    </div>
</div>
</body>
```

Pour voir ce que ça donne, c'est [ici](#).

Maintenant que vous savez tout sur la mise en page nous allons passer au contenu et voir ce que **Bootstrap** a à nous proposer dans ce domaine...

Partie 2 : Mise en forme des éléments

Dans cette deuxième partie, nous allons voir ensemble comment mettre en page, mettre en forme et agrémenter un peu le site.

Éléments de base

Dans cette partie nous allons voir ce que **Bootstrap** peut nous apporter dans la mise en forme et la présentation des éléments de base d'une page web : listes, tableaux, formulaires, boutons , icônes...

Listes et tableaux

Voici le deuxième jet de la page web de test avec ce template :

Code : HTML

```
<div class="container">
  <div class="page-header">...</div>
  <div class="row">
    <div class="span12">...</div>
  </div>
  <div class="row">
    <div class="span2">...</div>
    <div class="span2">...</div>
  </div>
  <div class="row">
    <div class="span4">...</div>
    <div class="span8">...</div>
  </div>
  <div class="row">
    <div class="span9">...</div>
    <div class="span3">...</div>
  </div>
</div>
```

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 :

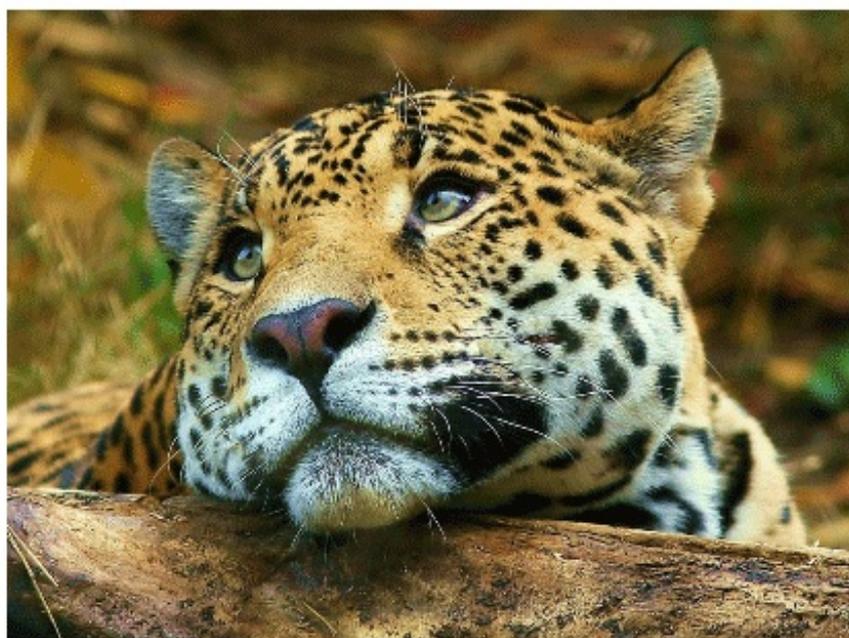
Le Tigre (*Panthera tigris*) est un mammifère carnivore de la famille des félidés (*Felidae*) du genre *Panthera*. Aisément reconnaissable à sa fourrure rousse rayée de noir, il est le plus grand félin sauvage et l'un des plus grands carnivores du monde. L'espèce est divisée en neuf sous-espèces possédant 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



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
28845 Félins-sur-Loire

La présentation s'est enrichie d'une rangée et de nouveaux éléments. Voici une visualisation du découpage :

page-header

row

row

row

row

Mon amour pour les tigres

Je suis passionné par les tigres depuis mes langentes. Ce site a été construit en hommage à ces merveilleux félins. Un très grand merci à qui a permis de faire connaître une espèce si menacée.

Voici ce que dit le Wikipédia :

Le tigre (Panthera tigris) est un mammifère de la famille des félins. C'est le plus grand félin existant et le plus grand chat du monde. L'espèce est divisée en neuf sous-espèces, mais il n'en reste que six en liberté. Les autres ont disparu à cause de la destruction de leur habitat et de la chasse, bien qu'il puisse atteindre de très grandes tailles, comme les buffles. Jusqu'au X^e siècle, le tigre était respecté par les Indiens. Il n'est plus considéré comme un animal sacré, mais il est toujours considéré comme un animal sacré par les Indiens.

— Wikipédia

Voici ses sous-espèces :

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

Les menaces pour les tigres

Menace	Description
Grand Malakong	Demande croissante de certaines parties de l'animal pour la médecine chinoise traditionnelle et l'usage à des fins médicinales.
Le défrichage	Progrès rapide de la déforestation et de la perte d'habitat.
Chasse	Chasse pour la viande et les os.
Malaisie	Les tigres sont considérés comme des animaux sacrés.
Carpe	Crois appât pour l'huile de palme.
Uganda	Commence à être considéré comme un animal sacré.

Voici ses sous-espèces :

span9

span3

Les listes

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>
    
```

Le résultat :

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

A noter l'existence d'une classe "unstyled" à appliquer à la balise , ce qui donne donc <ul class="unstyled">, qui supprime les puces. Voici 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



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>
    </tr>
  </tbody>
</table>
```

```

        <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>

```

Le résultat :

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

On va analyser les classes utilisées...

Classe "table"

C'est la classe de base pour les tableaux, elle fixe quelques styles et crée une bordure supérieure :

Code : CSS

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

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

Classe "table-bordered"

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

Code : CSS

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

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

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.

Bootstrap fait une utilisation intensive des nouvelles possibilités du CSS3 comme ici pour changer la couleur de fond des lignes impaires avec cette élégante syntaxe :

Code : CSS



```
.table-striped tbody tr:nth-child(odd) td, .table-striped
tbody tr:nth-child(odd) th {
  background-color: #f9f9f9;
}
```

Mais il paraît que IE ne les digère pas, je n'ai pas testé... 😬

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">
```

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	Pâte à papier, l'huile de palme et le caoutchouc

C'est la version finale que j'ai adoptée. Vous avez remarqué qu'on peut cumuler l'application des classes sans problème, comme je l'ai fait dans cette démonstration progressive.

Un peu de couleur dans les tableaux

La version 2.1 introduit la possibilité de colorer le fond des lignes des tableaux avec les classes **success**, **error** et **info**. Voici le tableau de notre exemple un peu animé de couleurs avec ces classes :

Code : HTML

```

<div class="container"> <br>
  <div class="row">
    <div class="span9 offset1">
      <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="error">
            <td>Île de Sumatra</td>
            <td>Production d'huile de palme et de pâtes à
papiers</td>
          </tr>
          <tr class="info">
            <td>Indonésie et Malaisie</td>
            <td>Pâte à papier, l'huile de palme et le
caoutchouc</td>
          </tr>
          <tr class="success">
            <td>États-Unis</td>
            <td>Les tigres captifs représentent un danger pour les
tigres sauvages</td>
          </tr>
          <tr class="error">
            <td>Europe</td>
            <td>Gros appétit pour l'huile de palme</td>
          </tr>
          <tr class="info">
            <td>Népal</td>
            <td>Commerce illégal de produits dérivés de tigres</td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>
</div>

```

```
</table>
</div>
</div>
</div>
```

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

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

Formulaire

Pour cette troisième version j'ai ajouté un peu de style et des formulaires :

tuto02_03a

bootstrap.twit.free.fr/tuto/tuto02_03a.html

Mon amour pour les tigres

Recherche

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.

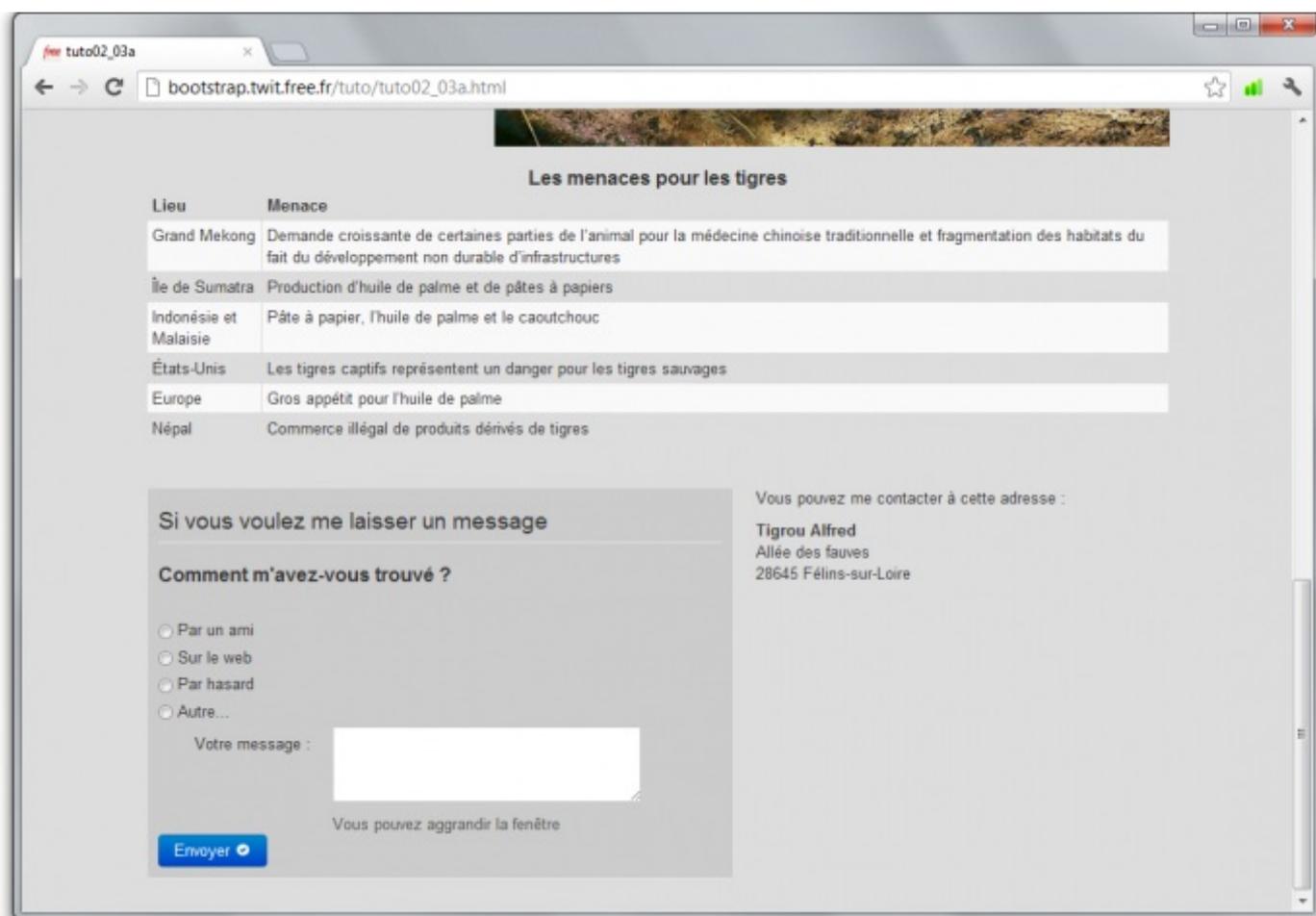
— Wikipédia



Voici les sous-espèces des tigres :

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





Au niveau du style juste une petite touche :

Code : CSS

```
body {
  background-color: #DDD;
}
[class*="span"] {
  margin-top: 10px;
  margin-bottom: 10px;
}
#formulaire {
  background-color: #CCC;
  padding: 10px;
}
```

Le fond d'un léger gris met bien en valeur le texte et les photos, les petites marges de 10px délimitent bien les éléments. Quant au formulaire c'est l'objet de ce chapitre...

Un formulaire de base

Bootstrap propose de base une mise en forme élégante des formulaires. Voici le formulaire de la page sans **Bootstrap** :

Code : HTML

```
<form>
  <fieldset>
    <legend>Si vous voulez me laisser un message</legend>
    <h4>Comment m'avez-vous trouvé ?</h4>
```

```
<label for="ami">
  <input type="radio" name="origine" value="ami" id="ami" />
  Par un ami </label>
<br/>
<label for="web">
  <input type="radio" name="origine" value="web" id="web" />
  Sur le web </label>
<br/>
<label for="hasard">
  <input type="radio" name="origine" value="hasard" id="hasard"
/>
  Par hasard </label>
<br/>
<label for="autre">
  <input type="radio" name="origine" value="autre" id="autre" />
  Autre... </label>
<br/>
<label for="textarea">Votre message :</label>
<textarea id="textarea" rows="3"></textarea>
<p>Vous pouvez agrandir la fenêtre</p>
<button type="submit">Envoyer</button>
</fieldset>
</form>
```

Et le résultat à l'écran :

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

Bon ce n'est pas très folichon, juste ce que génère le navigateur lorsqu'on l'aide pas. Voyons maintenant ce que ça donne en utilisant **Bootstrap** avec le même code **HTML** :

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

Envoyer

On a gagné en qualité typographique, mais au niveau organisation c'est un peu la panique. Nous allons arranger ça...

Les classes "radio" et "checkbox"

Bootstrap nécessite l'utilisation d'une classe particulière pour les cases à cocher (checkbox) et les zones d'option (radio). Voyons ce que ça donne en utilisant la classe **radio** pour notre formulaire :

Code : HTML

```
<form>
  <fieldset>
    <legend>Si vous voulez me laisser un message</legend>
    <h4>Comment m'avez-vous trouvé ?</h4>
    <br/>
    <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="3"></textarea>
    <p>Vous pouvez agrandir la fenêtre</p>
    <button type="submit">Envoyer</button>
```

```
</fieldset>
</form>
```

J'ai aussi supprimé les sauts de ligne qui deviennent inutiles. Résultat :

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

Maintenant les zones d'option sont bien organisées. Mais l'ensemble manque un peu d'homogénéité.

La classe "well" et la largeur des input

Voilà une classe que j'aime bien, elle provoque une jolie stylisation. Si j'apporte ce petit complément au formulaire précédent :

Code : CSS

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

On obtient déjà quelque chose de plus joli sans trop d'effort :

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

Mais je trouve la zone de texte un peu petite, je pourrais l'élargir en lui appliquant directement un style, mais **Bootstrap** nous propose des classes pratiques, par exemple **input-xlarge** :

Code : HTML

```
<textarea id="textarea" class="input-xlarge" rows="3"></textarea>
```

Cette classe ne fait pas grande chose, elle se contente de fixer la largeur à 270px :

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

Les valeurs possibles pour la largeur :

Code : CSS



```
.input-mini {width: 60px; }
.input-small {width: 90px; }
.input-medium {width: 150px; }
.input-large {width: 210px; }
.input-xlarge {width: 270px; }
.input-xxlarge {width: 530px; }
```

Le résultat commence à être pas mal du tout. La dernière chose qui me gêne est l'emplacement de la zone de texte et la place occupée par le texte d'aide "Vous pouvez agrandir la fenêtre". J'aimerais qu'il soit plus discret. Mais pour arranger ça, il va falloir un peu creuser les possibilités de **Bootstrap** au niveau des types de formulaires.

Les quatre types de formulaires

Bootstrap propose 4 types de formulaires :

- **Vertical** (valeur par défaut) : la classe correspondante était **form-vertical** dans les versions précédentes mais elle a judicieusement disparue avec la version 2.1 puisqu'elle n'était jamais utilisée, étant la valeur par défaut. Dans cette mise en forme de base les libellés sont au-dessus de leur contrôle :

Votre message :

- **Inline** : les libellés sont alignés à gauche suivis de leur contrôle. La classe correspondante est **form-inline**. En gros les éléments deviennent en ligne et alignés à gauche. Ça peut être pratique pour ne pas encombrer une page verticalement mais ne peut s'envisager que pour un petit formulaire sans beaucoup de contrôles sinon ça devient vite illisible. Voici un exemple avec quelques contrôles :

Code : HTML

```
<form class="form-inline">
  <input type="text" class="input-small">
  <label class="checkbox">
    <input type="checkbox"> Bouh !
  </label>
  <label class="radio">
    <input type="radio"> Hey !
  </label>
  <select class="input-small">
    <option>1</option>
    <option>2</option>
    <option>3</option>
  </select>
  <button type="submit" class="btn">Ahh !!!</button>
  <span class="help-inline">Attention !</span>
</form>
```

- **Search** : forme arrondie spéciale des zones de saisie de texte. La classe correspondante est **form-search**. Nous allons l'utiliser pour le formulaire de recherche situé dans l'en-tête de la page.
- **Horizontal** : les contrôles ont un alignement flottant à gauche avec leur libellé aligné à droite sur la même ligne. La classe correspondante est **form-horizontal**. C'est ce que j'aimerais pour la zone de texte, alors nous allons utiliser cette classe.

Les classes "form-horizontal", "control-group", "control-label", "controls" et "help-block"

Lorsqu'on applique la classe **form-horizontal** au formulaire :

Code : HTML

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

On obtient ce résultat :

Autrement dit on a rien changé ! Que se passe-t-il ? En analysant, on se rend compte que le `<label>` possède ce style :

Code : CSS

```
label {
  display: block;
  margin-bottom: 5px;
}
```

On a donc encore un comportement de type "block" avec un retour à la ligne. Autrement dit avec la classe `form-horizontal` on a un comportement sans doute correct pour les contrôles standards mais il faut un complément pour un `<label>`. **Bootstrap** a prévu pour cela la classe `control-label`. Appliquons là pour voir l'effet :

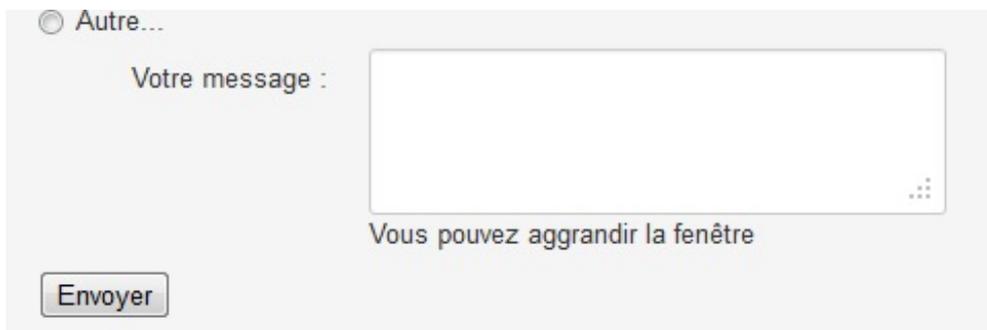
Code : HTML

```
<label class="control-label" for="textarea">Votre message :</label>
```

Cette fois on a bien un comportement "en ligne" mais il faut améliorer la mise en forme. Le libellé est trop collé à la zone de texte. On va arranger ça avec la classe `controls` :

Code : HTML

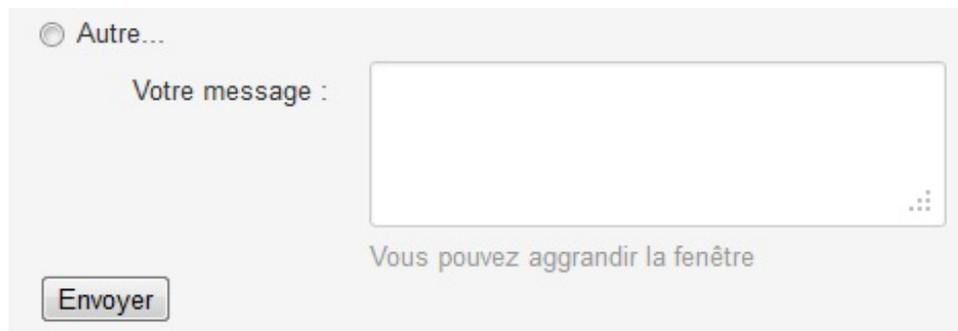
```
<div class="controls">
  <textarea id="textarea" class="input-xlarge"
rows="3"></textarea>
  <p>Vous pouvez agrandir la fenêtre</p>
</div>
```

A screenshot of a web form. At the top left, there is a radio button labeled "Autre...". Below it, the text "Votre message :" is followed by a large text area. Underneath the text area, the text "Vous pouvez agrandir la fenêtre" is displayed. At the bottom left, there is a button labeled "Envoyer".

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**:

Code : HTML

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

A screenshot of the same web form as above, but the text "Vous pouvez agrandir la fenêtre" is now styled as a help block. It is smaller, lighter in color, and has a subtle background, making it less prominent than the text area above it.

On est enfin arrivé à ce qu'on voulait ! Mais pour peaufiner l'aspect j'ai ajouté ma petite touche personnelle en changeant la couleur de fond et la marge interne :

Code : CSS

```
#formulaire {
  background-color: #CCC;
  padding: 10px;
}
```

En identifiant mon formulaire pour l'application du style :

Code : HTML

```
<form class="form-horizontal well" id="formulaire">
```

Et voici le résultat final :

Enfin pas vraiment final parce que nous allons voir bientôt l'aspect du bouton...

Un formulaire pour la recherche

La page comporte aussi un formulaire de recherche dans l'en-tête. La version 2.1 a apporté des améliorations dans l'aspect de ce type de formulaire. Nous allons utiliser la classe **form-search** pour le formulaire et **search-query** pour la zone de texte :

Code : HTML

```
<form class="form-search">
  <div class="input-append">
    <input type="text" class="search-query">
    <button class="btn btn-primary"
type="submit">Recherche</button>
  </div>
</form>
```



La classe **input-append** a pour but de découper le bouton sur son côté gauche pour l'accoler à la zone de texte. Il existe aussi sa réciproque **input-prepend** avec évidemment le résultat inverse :

Code : HTML

```
<form class="form-search">
<div class="input-prepend">
  <button class="btn btn-primary"
type="submit">Recherche</button>
```

```

    <input type="text" class="search-query">
  </div>
</form>

```



Les classes `input-append` et `input-prepend` à l'oeuvre

Ces deux classes que nous avons utilisées dans les formulaires de recherche pour accoler un bouton à un **input** peuvent aussi servir, associées à la classe **add-on**, à accoler une information :

Code : HTML

```

<div class="input-prepend">
  <span class="add-on">€</span>
  <input class="input-mini" type="text" value="100">
</div>
<div class="input-append">
  <input class="input-small" type="text" style="text-align:right"
value="100,50">
  <span class="add-on">€</span>
</div>

```



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

Code : HTML

```

<div class="input-prepend input-append">
  <span class="add-on">€</span>
  <input class="span1" id="appendedPrependedInput" type="text">
  <span class="add-on">.00</span>
</div>

```



Remarquez l'utilisation de la classe **span1** à l'intérieur de **input**. C'est une alternative à l'utilisation de la série des **input-xxx**.

On peut évidemment accoler un bouton comme nous l'avons vu dans le formulaire de recherche, mais on peut également en accoler plusieurs :

Code : HTML

```

<div class="input-prepend input-append">
  <span class="add-on">€</span>
  <input class="input-mini" id="appendedPrependedInput" type="text">
  <button class="btn" type="button">Vérifier</button>

```

```

<button class="btn" type="button">Annuler</button>
<button class="btn" type="button">Valider</button>
</div>

```

Stylisation selon l'état

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

Code : HTML

```

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

```

Avec ce résultat (cliquez sur l'image pour voir en action) :

Validation, erreur et succès

Bon en fait d'action il y en a peu parce que je n'ai pas prévu d'interactivité. Alors un petit exemple 😊

Nous voulons faire un formulaire de ce genre :

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 :

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="span5">
  <form class="form-inline well">
    <div class="control-group">
      <div class="controls">
        <input type="text" placeholder="Texte ici">
        <button type="submit" class="btn btn-primary pull-right">Envoyer</button>
        <div class="alert alert-error hide">
          <h4 class="alert-heading">Erreur !</h4>
          Vous devez entrer au moins 4 caractères ! </div>
        </div>
      </div>
    </div>
  </form>
</div>
<script src="../../js/jquery.js"></script>
<script>
  $(function() {
    $("form").on("submit", function() {
      if($("input").val().length < 4) {
        $("div.control-group").addClass("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**.

Mise en forme des listes de choix

La version 2.1 a introduit une mise en forme des listes avec sélection multiple. Voici un exemple avec deux listes identiques mais la deuxième a été équipée de la sélection multiple :

Code : HTML

```

<form class="form-horizontal">
  <div class="control-group">
    <label class="control-label" for="simple">Sélection simple
  :</label>
    <div class="controls">
      <select id="simple" class="input-mini">
        <option>1</option>
        <option>2</option>
        <option>3</option>
        <option>4</option>
        <option>5</option>
      </select>
    </div>
  </div>
  <div class="control-group">
    <label class="control-label" for="multiple">Sélection
multiple :</label>
    <div class="controls">
<select id="multiple" multiple="multiple" class="input-mini">
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
    </select>
    </div>
  </div>
</form>

```

Sélection simple : Sélection multiple :

Boutons, icônes et images

Bootstrap propose quelques classes intéressantes pour l'aspect des boutons. Nous allons en appliquer sur la page d'exemple au niveau des deux formulaires.

Un joli bouton pour les formulaires

Reprenons le formulaire de recherche en lui ajoutant une mise en forme du bouton :

Code : HTML

```

<form class="well form-search">
  <input type="text" class="input-medium search-query">
  <button class="btn" type="submit">Recherche</button>
</form>

```

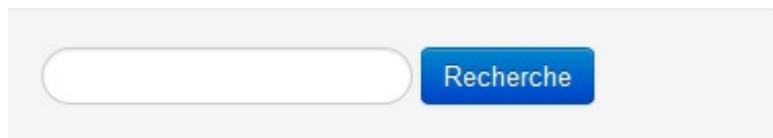
La classe de base pour les boutons est **btn**, voici ce que ça donne :



Le bouton est en harmonie avec les zone de texte. On peut aussi améliorer avec des couleurs :

Code : HTML

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



C'est la couleur de base donnée par la classe **btn-primary**. Il y en a quelques autres :



Les classes correspondantes sont :



- btn-info
- btn-success
- btn-warning
- btn-danger
- bnt-inverse

A utiliser selon les situations et à faire varier selon l'humeur 😊.

On peut aussi ajuster la dimension du bouton :

Code : HTML

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



Les valeurs possibles sont :



- btn-mini
- btn-small
- btn-large

La version 2.1 introduit la nouvelle classe **btn-block** qui règle la largeur du bouton sur la dimension de son contenant. Voici un exemple :

Code : HTML

```
<div class="container">
  <br>
  <div class="row">
    <div class="span2">
      <a class="btn btn-medium btn-block btn-danger" href="#">Span
2</a>
    </div>
    <div class="span4">
      <a class="btn btn-small btn-block btn-info" href="#">Span 4</a>
    </div>
    <div class="span3">
      <a class="btn btn-large btn-block btn-inverse" href="#">Span
3</a>
    </div>
  </div>
</div>
```

Span 2

Span 4

Span 3

Grouper des boutons

Il est possible de grouper des boutons avec la classe **btn-group**. Ce groupement peut s'effectuer de façon verticale avec la classe **btn-group-vertical** depuis la version 2.1. Voici un exemple avec deux groupements :

Code : HTML

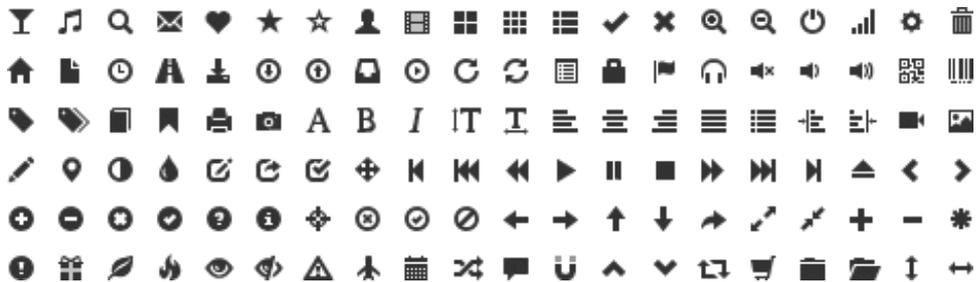
```
<div class="span1 offset1">
  <div class="btn-group">
    <a class="btn btn-danger" href="#">1</a>
    <a class="btn btn-info" href="#">2</a>
    <a class="btn btn-warning" href="#">3</a>
  </div>
</div>
<div class="span1 offset1">
  <div class="btn-group btn-group-vertical">
    <a class="btn btn-danger" href="#">4</a>
    <a class="btn btn-info" href="#">5</a>
    <a class="btn btn-warning" href="#">6</a>
  </div>
</div>
```



De jolies icônes

Bootstrap propose aussi des icônes. Elles sont toutes regroupées dans deux fichiers :

- glyphicons-halflings.png



- glyphicons-halflings-white.png



Bon là je peux pas vous montrer les images parce qu'elles sont blanches et sur le fond blanc de la page on va pas voir grand-chose 😞. Mais ce sont exactement les mêmes qu'on utilisera évidemment sur des fonds sombres.

Le principe utilisé est celui des sprites. On a toutes les icônes sur une seule image et on s'arrange pour afficher une seule icône. Pour tout savoir sur cette méthode, vous pouvez voir [le tuto de warpShadow](#) sur le sujet.

L'intégration d'une icône à un bouton est très facile parce qu'il suffit d'utiliser la balise `<i>`. On va compléter le formulaire de contact de la page d'exemple en améliorant le bouton et le complétant par une icône :

Code : HTML

```
<form class="form-horizontal well" id="formulaire">
  <legend>Si vous voulez me laisser un message</legend>
  <h4>Comment m'avez-vous trouvé ?</h4>
  <br/>
  <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 class="control-label" for="textarea">Votre message
  </label>
    <div class="controls">
      <textarea id="textarea" class="input-xlarge"
rows="3"></textarea>
```

```

    <p class="help-block">Vous pouvez agrandir la fenêtre</p>
  </div>
  <button class="btn btn-primary" type="submit">Envoyer <i
class="icon-white icon-ok-sign"></i> </button>
</fieldset>
</form>

```



La syntaxe est toute simple. On utilise une balise `<i>` avec deux classes. La principale est **icon-ok-sign** qui indique quelle icône on désire afficher. La classe **icon-white** indique que je veux une icône blanche puisque le fond est sombre.



Et si on veut l'icône noire ?

Alors il suffit de ne pas mettre la classe **icon-white** 😊

Évidemment on peut mettre ces icônes ailleurs que sur des boutons 😊. Par exemple dans un menu.

Créer son pack d'icônes

On peut aussi se créer son pack d'icônes personnelles en gardant le gabarit pour profiter directement des classes de **Bootstrap**. Mais on peut aussi utiliser les services d'un site comme [CSS Sprite Generator](#). Celui-ci permet, à partir d'une collection de fichiers d'icônes, non seulement de créer le **sprite** mais aussi le fichier **CSS** ! Vous pouvez trouver la procédure complète de réalisation d'un pack de cette manière sur le site [fabulous](#) avec en prime des centaines d'icônes libres de droits. Voici ce que j'ai pu réaliser en 5 minutes :

Code : HTML

```

<link href="../bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="../fabulous/cus-icons.css" rel="stylesheet"
type="text/css">
...
<div class="btn-group">
  <a class="btn" href="#"><i class="cus-lock"></i> Verrouiller</a>
  <a class="btn" href="#"><i class="cus-compress"></i>
Compresser</a>
  <a class="btn" href="#"><i class="cus-wrench"></i> Réparer</a>
</div>

```



Une barre de boutons

La classe **btn-toolbar** permet de faire très facilement des barres de boutons. Voici un exemple :

Code : HTML

```

<div class="span5 offset1 btn-toolbar">
  <div class="btn-group">
    <button class="btn btn-danger"><i class="icon-align-
left"></i></button>
    <button class="btn btn-info"><i class="icon-align-

```

```

center"></i></button>
  <button class="btn btn-warning"><i class="icon-align-
right"></i></button>
  <button class="btn btn-success"><i class="icon-align-
right"></i></button>
</div>
<div class="btn-group">
  <button class="btn btn-danger"><i class="icon-
print"></i></button>
  <button class="btn btn-info"><i class="icon-
time"></i></button>
  <button class="btn btn-warning"><i class="icon-
picture"></i></button>
</div>
</div>

```



On pourrait avoir envie de faire une barre de boutons verticale en contraignant des groupes de boutons à se superposer dans un `span1`, mais on se rend compte d'un petit problème :

Code : HTML

```

<div class="span1 offset1 btn-toolbar">
  <div class="btn-group btn-group-vertical">
    <button class="btn btn-danger"><i class="icon-align-
left"></i></button>
    <button class="btn btn-info"><i class="icon-align-
center"></i></button>
    <button class="btn btn-warning"><i class="icon-align-
right"></i></button>
    <button class="btn btn-success"><i class="icon-align-
right"></i></button>
  </div>
  <div class="btn-group btn-group-vertical">
    <button class="btn btn-danger"><i class="icon-
print"></i></button>
    <button class="btn btn-info"><i class="icon-
time"></i></button>
    <button class="btn btn-warning"><i class="icon-
picture"></i></button>
  </div>
</div>

```



Le décalage constaté sur le deuxième groupe est dû à la règle :

Code : CSS

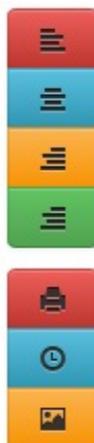
```
.btn-group + .btn-group {
  margin-left: 5px;
}
```

Celle-ci est utile pour décaler les groupes de boutons lorsqu'ils sont placés horizontalement. Alors pour avoir un aspect correct il faut ajouter une règle :

Code : CSS

```
.btn-group + .btn-group-vertical {
  margin-left: 0;
}
```

Et alors l'alignement devient celui qu'on attend :



Mise en forme des images

La version 2.1 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-polaroid	Image avec liseré blanc

Voici un exemple illustratif de ces trois effets :

Code : HTML

```
<div class="container"> <br>
  <div class="span2 offset1">
    
  </div>
  <div class="span2">
    
  </div>
  <div class="span2">
    
  </div>
</div>
```



Un petit TP

Pour faire le point sur les formulaires et les boutons, je vous propose un petit TP. Il faut réaliser ce formulaire :

Se souvenir de moi

Une zone de saisie pour un courriel, une autre pour un mot de passe, une petite case à cocher et un joli bouton, le tout bien organisé...

Secret (cliquez pour afficher)

Code : HTML

```
<div class="span5">
  <form class="well form-inline">
    <input type="text" class="input-medium" placeholder="Email">
    <input type="password" class="input-medium pull-right"
placeholder="Mot de passe">
    <div class="controls"> <br/>
      <label class="checkbox">
        <input id="optionsCheckbox" type="checkbox"
value="option1">
        Se souvenir de moi </label>
```

```

<button type="submit" class="btn btn-primary btn-small pull-right"> <i class="icon-user icon-white"></i> Envoyer</button>
</div>
</form>
</div>

```

Vous aviez trouvé évidemment 😊 . Voici le formulaire en action.

Maintenant si on faisait une petite validation de ce formulaire à la sauce CSS3 ?

Secret (cliquez pour afficher)

Code : HTML

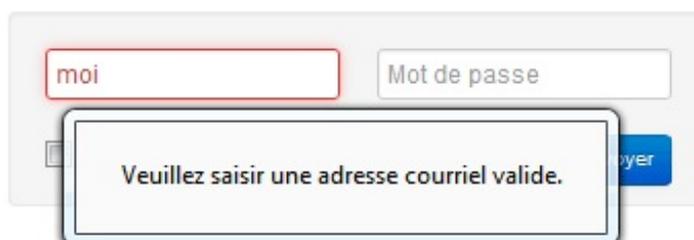
```

<div class="span5">
  <form class="well form-inline">
    <input type="email" class="input-medium" autofocus required
    placeholder="monadresse@chez.moi">
    <input type="password" class="input-medium pull-right" required
    placeholder="Mot de passe">
    <div class="controls"> <br/>
      <label class="checkbox">
        <input id="options" type="checkbox" value="option1">
        Se souvenir de moi </label>
      <button type="submit" class="btn btn-primary btn-small pull-right"> <i class="icon-user icon-white"></i> Envoyer</button>
    </div>
  </form>
</div>

```

J'ai ajouté quelques propriétés comme **required** pour signaler que la saisie est obligatoire, **autofocus** pour que le contrôle correspondant reçoive le focus. J'ai aussi défini le type **email** pour le contrôle de l'adresse du courriel. Rien de bien extraordinaire mais [testez maintenant ce formulaire](#).

Si vous avez un navigateur pas trop antique, vous devriez voir ce genre de réaction :



Plutôt sympathique non ? 😊

Après ce petit tour d'horizon des éléments de bases qui constituent une page web nous allons passer aux éléments spécifiques intégrés dans **Bootstrap**.

Les composants intégrés

Bootstrap propose des composants bien pratiques à mettre en œuvre : barre de navigation, effets typographiques, 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 mises en œuvre...

Allez c'est parti pour une visite guidée !

La navigation

Une barre de navigation

Principe de base

Bootstrap propose de superbes classes pour créer une barre de navigation. A la base une barre nécessite cette simple structure :

Code : HTML

```
<nav class="navbar">
  <div class="navbar-inner">
    <div class="container">
      ...
    </div>
  </div>
</nav>
```

La classe englobante est **navbar**. Elle est toute simple :

Code : CSS

```
.navbar {
  *position: relative;
  *z-index: 2;
  margin-bottom: 20px;
  overflow: visible;
  color: #555555;
}
```

Donc une petite marge basse pour bien distinguer la barre du reste de la page et une précaution pour que tous les éléments apparaissent.

La classe suivante est **navbar-inner**. Celle-là est plus chargée, c'est elle qui réalise la mise en forme :

Code : CSS

```
.navbar-inner {
  min-height: 40px;
  padding-right: 20px;
  padding-left: 20px;
  background-color: #fafafa;
  background-image: -moz-linear-gradient(top, #ffffff, #f2f2f2);
  background-image: -webkit-gradient(linear, 0 0, 0 100%,
from(#ffffff), to(#f2f2f2));
  background-image: -webkit-linear-gradient(top, #ffffff, #f2f2f2);
  background-image: -o-linear-gradient(top, #ffffff, #f2f2f2);
  background-image: linear-gradient(to bottom, #ffffff, #f2f2f2);
  background-repeat: repeat-x;
  border: 1px solid #d4d4d4;
  -webkit-border-radius: 4px;
  -moz-border-radius: 4px;
  border-radius: 4px;
  filter:
```

```

progid:dximagetransform.microsoft.gradient(startColorstr='#ffffff',
endColorstr='#fff2f2f2', GradientType=0);
-webkit-box-shadow: 0 1px 4px rgba(0, 0, 0, 0.065);
-moz-box-shadow: 0 1px 4px rgba(0, 0, 0, 0.065);
box-shadow: 0 1px 4px rgba(0, 0, 0, 0.065);
}

```

Elle fixe le fond, les arrondis des angles, l'ombre, et la marge interne...

La classe suivante est la classe englobante que nous avons déjà rencontrée **container** et je n'y reviens pas.

Il faut ensuite garnir la barre d'items. Voici un contenu d'exemple :

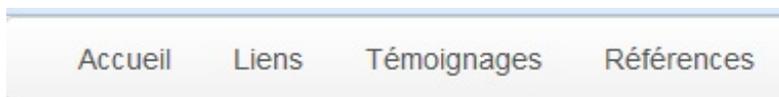
Code : HTML

```

<nav class="navbar">
  <div class="navbar-inner">
    <div class="container">
      <ul class="nav">
        <li> <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>
  </div>
</nav>

```

La classe **nav** fixe l'apparence et le positionnement de tous les éléments. Elle doit être utilisée avec une balise ****. Tous les items de navigation apparaissent alors avec des balises ****. Voici un aperçu de la barre :



Barre avec un fond noir



Avant la version 2.1 la barre avait un fond noir. Elle s'est retrouvée avec un fond blanc à partir de cette version

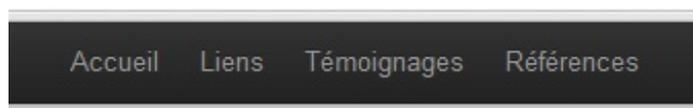
On peut facilement retrouver l'ancien look avec la classe **navbar-inverse** :

Code : HTML

```

<nav class="navbar navbar-inverse">
  <div class="navbar-inner">
    <div class="container">
      <ul class="nav">
        <li> <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>
  </div>
</nav>

```

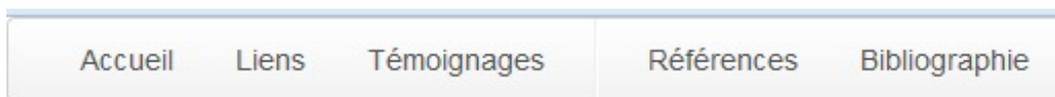


Insertion d'un séparateur

On peut insérer un séparateur pour distinguer visuellement des groupes d'items avec la classe **divider-vertical** :

Code : HTML

```
<ul class="nav">
  <li><a href="#">Accueil</a></li>
  <li><a href="#">Liens</a></li>
  <li><a href="#">Témoignages</a></li>
  <li class="divider-vertical"></li>
  <li><a href="#">Références</a></li>
  <li><a href="#">Bibliographie</a></li>
</ul>
```

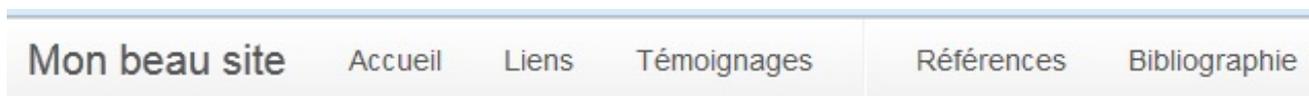


Un titre dans la barre

On peut faire apparaître un titre avec des caractères plus visibles avec la classe **brand** :

Code : HTML

```
<ul class="nav">
  <li><a class="brand" href="#">Mon beau site</a></li>
  <li><a href="#">Accueil</a></li>
  <li><a href="#">Liens</a></li>
  <li><a href="#">Témoignages</a></li>
  <li class="divider-vertical"></li>
  <li><a href="#">Références</a></li>
  <li><a href="#">Bibliographie</a></li>
</ul>
```



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>
<nav class="navbar">
  <div class="navbar-inner">
    <div class="container">
      ...
    </div>
  </div>
</nav>
<p>Un paragraphe</p>
```

Un paragraphe

Mon beau site Accueil Liens Témoignages

Un paragraphe

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

Code : HTML

```
<p>Un paragraphe</p>
<nav class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container">
      ...
    </div>
  </div>
</nav>
<p>Un paragraphe</p>
```

Mon beau site Accueil Liens
Un paragraphe



Mais où sont passés les paragraphes ?

Ils 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;
  margin-bottom: 0;
}
```

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

```
body {
  padding-top: 60px;
}
```

Voilà maintenant l'effet :



Maintenant on retrouve nos paragraphes 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.



La version 2.1 a vu l'arrivée de la classe **navbar-static-top** qui permet de fixer la barre en haut de la page en pleine largeur sans avoir à se soucier du padding

Voilà un exemple :

Code : HTML

```
<nav class="navbar navbar-static-top">
  <div class="navbar-inner">
    <div class="container">
      <ul class="nav">
        <li> <a class="brand" href="#">Mon beau site</a> </li>
        <li> <a href="#">Accueil</a> </li>
        <li> <a href="#">Liens</a> </li>
        <li> <a href="#">Témoignages</a> </li>
        <li class="divider-vertical"></li>
        <li> <a href="#">Références</a> </li>
        <li> <a href="#">Bibliographie</a> </li>
      </ul>
    </div>
  </div>
</nav>
```



Un formulaire de recherche dans la barre

Pour intégrer un formulaire il suffit de le prévoir à l'intérieur de la **div** qui a la classe **container**. Il y a une classe spéciale pour intégrer un formulaire de recherche dans la barre : **navbar-search**. On a deux classes pour envoyer ce formulaire sur l'un des côtés :

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

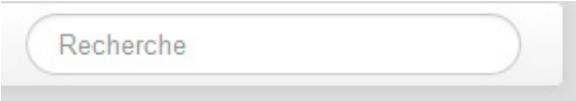
Voilà le code du formulaire :

Code : HTML

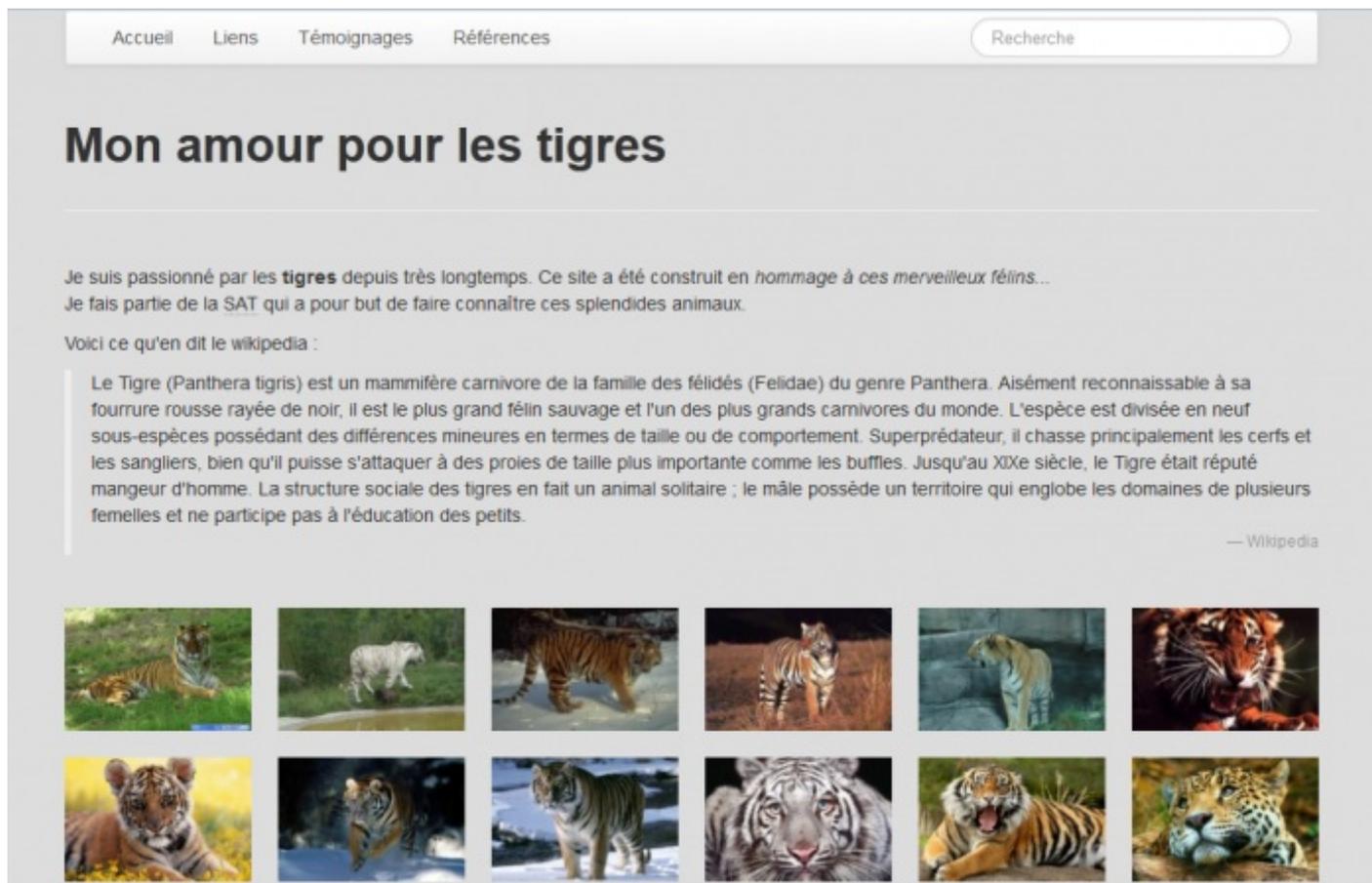
```
<form class="navbar-search pull-right" action="">
  <input class="search-query" type="text" placeholder="Recherche">
```

```
</form>
```

Le résultat est assez convaincant :



Voici la page d'exemple avec une barre de navigation et un formulaire de recherche au niveau de cette barre :



Accueil Liens Témoignages Références Recherche

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

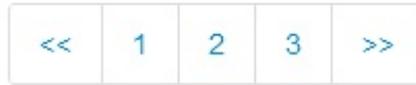
La pagination

Un exemple simple

Bootstrap propose la classe **pagination** pour réaliser d'esthétiques... paginations. Le principe est tout simple, voici un exemple de base :

Code : HTML

```
<div class="pagination">
  <ul>
    <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>
```



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 rendre les liens inactifs on a la classe **disabled** et pour les rendre actifs on a la classe **active**. Le code Javascript est inspiré d'une réponse à une question sur [ce site](#). J'ai nettoyé et arrangé un peu le code pour qu'il devienne efficace et lisible :

Code : HTML

```

<body>
<br/>
<div class="container">
  <div class="row">
    <div id="content">
      <div class="span2">  </div>
      <div class="span2">  </div>
    </div>
    <div id="page_navigation"> </div>
  </div>
</script src="../js/jquery.js"></script>
</body>
</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 = '<div class="pagination">'
    nav += '<ul>';
    nav += '<li class="previous_link">';
    nav += '<a href="javascript:previous();"><<<</a>';
    nav += '';
    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 += '';
    }
    nav += '<li class="next_link" >';
    nav += '<a href="javascript:next();">>>></a>';
    nav += '';
    nav += '</ul>';
    nav += '</div>';

    $('#page_navigation').html(nav);
    set_display(0, show_per_page);
});
</script>

```



Effets typographiques Hero Unit

La classe **hero-unit** permet de rendre le texte très visible. Prenons un exemple :

Code : HTML

```

<h1>Coucou</h1>
<p>et bonjour</p>
<div class="hero-unit">
  <h1>Coucou</h1>

```

```
<p>et bonjour</p>
</div>
```

Coucou

et bonjour



On voit bien la différence entre les éléments "normaux" et ceux qui bénéficient de la classe **hero-unit**.

Voyons cette classe :

Code : CSS

```
.hero-unit {
  padding: 60px;
  margin-bottom: 30px;
  background-color: #eeeeee;
  -webkit-border-radius: 6px;
  -moz-border-radius: 6px;
  border-radius: 6px;
}
```

Un peu de gris en fond, des coins arrondis et quelques marges.

Ensuite on trouve une stylisation de `<h1>` :

Code : CSS

```
.hero-unit h1 {
  margin-bottom: 0;
  font-size: 60px;
  line-height: 1;
  letter-spacing: -1px;
  color: inherit;
}
```

Avec un net renforcement de la police.

De même pour la balise `<p>` :

Code : CSS

```
.hero-unit p {
  font-size: 18px;
  font-weight: 200;
  line-height: 30px;
  color: inherit;
}
```

}

Des libellés colorés

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

Code : HTML

```
<p><span class="label">Valeur d'inhibition</span> 600</p>
<p><span class="label label-inverse">Valeur au repos</span> 1000</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-important">Valeur critique</span>
12000</p>
```

Valeur d'inhibition 600

Valeur au repos 1000

Valeur de signalement 4000

Valeur de fonctionnement 8000

Valeur dangereuse 11000

Valeur critique 12000

Des badges colorés

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

Code : HTML

```
<p><span class="badge">600</span> </p>
<p><span class="badge badge-inverse">1000</span> </p>
<p><span class="badge badge-info">4000</span> </p>
<p><span class="badge badge-success">8000</span> </p>
<p><span class="badge badge-warning">11000</span> </p>
<p><span class="badge badge-error">12000</span> </p>
```

600

1000

4000

8000

11000

12000

Thumbnails

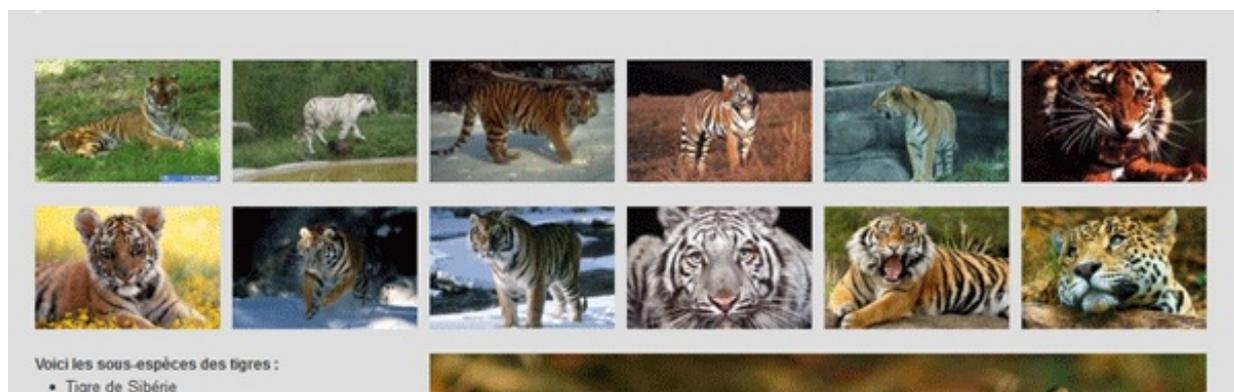
Si vous avez des images, des vidéos ou tous autres éléments à afficher dans une grille, **Bootstrap** propose le composant

Thumbnails très facile à mettre en œuvre. Je reviens sur ma page d'exemple qui comporte des petites images de tigres. Jusque-là, j'ai positionné les images en utilisant tout simplement la grille :

Code : HTML

```
<div class="row">
  <div class="span2"></div>
  <div class="span2"></div>
</div>
```

Ce qui donne une organisation bien régulière :



Voyons ce que ça donne avec le composant **Thumbnails**. Le principe en est simple, il faut utiliser la classe **thumbnails** dans une balise ****. Les éléments sont ensuite positionnés logiquement dans des balises ****. Pour obtenir l'effet graphique chaque élément doit être stylisé par la classe **thumbnail**, soit dans une balise **<a>** si on place un lien, soit dans une balise **<div>** dans le cas contraire. Voilà ce que ça donne pour la page d'exemple :

Code : HTML

```
<ul class="thumbnails">
  <li class="span2">
    <div class="thumbnail"></div>
  </li>
  <li class="span2">
    <div class="thumbnail"></div>
  </li>
  <li class="span2">
```

```

    <div class="thumbnail"></div>
  </li>
  <li class="span2">
    <div class="thumbnail"></div>
  </li>
</ul>

```

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

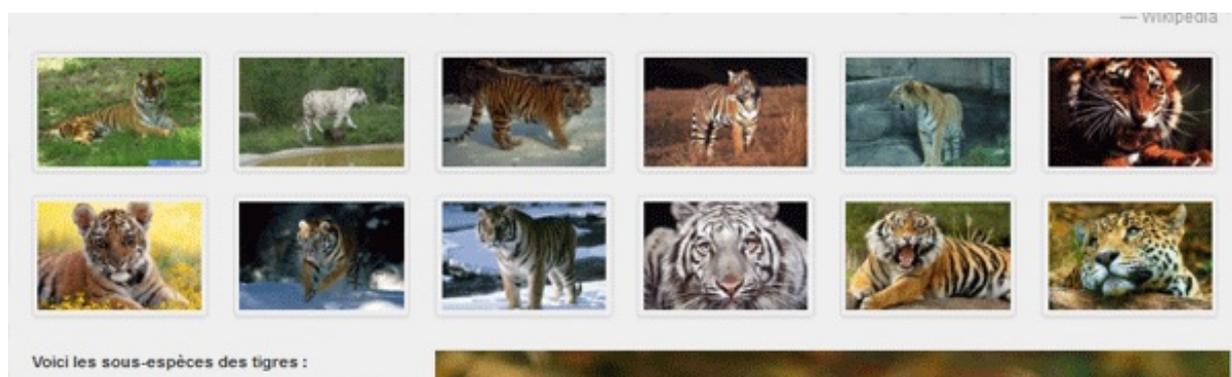
Code : HTML

```

<div class="span8">
  <div class="thumbnail"></div>
</div>

```

Le résultat à l'écran (cliquez sur l'image pour ouvrir la page correspondante, comme depuis le début de ce tuto 😊) :



J'ai dû adoucir un peu le gris du fond pour faire apparaître les bordures des diapositives. Le résultat est tout à fait sympathique. On pourrait évidemment créer des liens pour les images, auquel cas la balise `<a>` intégrerait la classe **thumbnail**.

Tout le travail de stylisation est d'ailleurs réalisé par cette classe :

Code : CSS

```
.thumbnail {
  display: block;
  padding: 4px;
  line-height: 20px;
  border: 1px solid #ddd;
  -webkit-border-radius: 4px;
  -moz-border-radius: 4px;
  border-radius: 4px;
  -webkit-box-shadow: 0 1px 3px rgba(0, 0, 0, 0.055);
  -moz-box-shadow: 0 1px 3px rgba(0, 0, 0, 0.055);
  box-shadow: 0 1px 3px rgba(0, 0, 0, 0.055);
  -webkit-transition: all 0.2s ease-in-out;
  -moz-transition: all 0.2s ease-in-out;
  -o-transition: all 0.2s ease-in-out;
  transition: all 0.2s ease-in-out;
}
```

L'ombrage est réalisé par **box-shadow** qui est aussi une propriété apparue avec le CSS3. Si vous ne vous êtes pas encore informé sur ces nouveautés, je vous conseille un petit coup d'œil à la [page du W3C](#) sur les fonds et les bordures.

Alertes et barres de progression

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

Alertes

Les alertes permettent de présenter un message visuellement significatif. Il suffit d'utiliser la classe **alert** :

Code : HTML

```
<div class="alert span3">
  <strong>Attention!</strong> Vous allez trop loin !
</div>
```

Attention! Vous allez trop loin !

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

Code : HTML

```
<div class="alert span1 alert-info">
  <strong>Infos !</strong>
</div>
<div class="alert span1 alert-success">
  <strong>Bravo!</strong>
</div>
<div class="alert span1 alert-danger">
  <strong>Danger!</strong>
</div>
```



Si le texte doit être conséquent la classe **alert-block** règle les marges internes pour aérer un peu :

Code : HTML

```
<div class="alert span5 alert-info alert-block">
  <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>
```

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...

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-block span4 hide">
    <button type="button" class="close">×</button>
    <h4>Attention!</h4> Petite information importante !
  </div>
  <div class="span3">
    <input type="button" class="btn-primary" id="afficher"
value="Afficher l'alerte">
  </div>
</div>
<script src="../js/jquery.js"></script>
<script>
$(function () {
```

```

$("#afficher").click(function () {
    $("#afficher").hide();
    $(".alert").show("slow");
});
$(".close").click(function () {
    $(".alert").hide("slow");
    $("#afficher").show();
});
});
</script>

```

Afficher l'alerte

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 **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 well">
<div class="bar"></div>
</div>
  <div class="span3">
    <input type="button" class="btn-primary" id="animer"
value="Animer">
  </div>
</div>
<script src="../js/jquery.js"></script>
<script>
function timer(n) {
  $(".bar").css("width", n + "%");
  if(n < 100) {
    setTimeout(function () {
      timer(n + 10);
    }, 200);
  }
}
$(function () {
  $("#animer").click(function () {
    timer(0);
  });
});
</script>

```

Animer

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. La classe **well** est juste utilisée pour l'esthétique.

Il existe une version rayée de la barre :

Code : HTML

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



Animer

Il existe aussi une version rayée et élégamment animée :

Code : HTML

```
<div class="progress progress-striped active well">
  <div class="bar"></div>
  <div id="pourcentage" class="pull-right"></div>
</div>
```



Animer

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 well">
  <div class="progress progress-info">
    <div class="bar"></div>
  </div>
  <div class="progress progress-success">
    <div class="bar"></div>
  </div>
  <div class="progress progress-warning">
    <div class="bar"></div>
  </div>
  <div class="progress progress-danger">
    <div class="bar"></div>
  </div>
  <div class="progress progress-info progress-striped">
    <div class="bar"></div>
  </div>
  <div class="progress progress-success progress-striped">
    <div class="bar"></div>
  </div>
  <div class="progress progress-warning progress-striped">
    <div class="bar"></div>
  </div>
  <div class="progress progress-danger progress-striped">
```

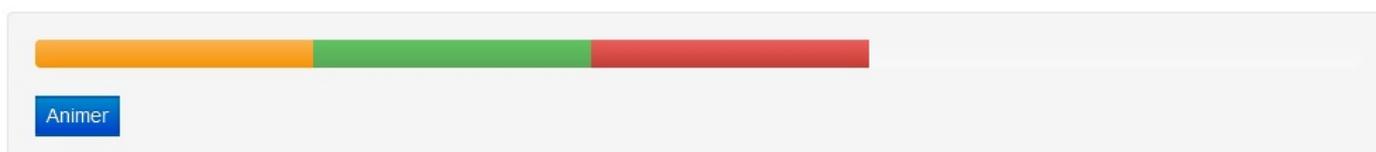
```
<div class="bar"></div>
</div>
<input type="button" class="btn-primary" id="animer"
value="Animer">
</div>
```



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

Code : HTML

```
<div class="progress">
  <div class="bar bar-warning"> </div>
  <div class="bar bar-success"> </div>
  <div class="bar bar-danger"> </div>
</div>
```



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

Nous sommes arrivés aux limites de l'utilisation de **Bootstrap** sans faire appel à ses plugins **jQuery**. Notre prochaine étape sera donc d'explorer les possibilités de **Bootstrap** boosté par ses plugins !

Partie 3 : Mise en forme avancée

Dans cette partie, nous verrons comment animer un peu le 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. Si vous ne savez pas manipuler **jQuery** vous pourrez au moins utiliser les composants proposés de façon statique. Dans le cas contraire, vous pouvez encore combler cette lacune 😊

L'ensemble des plugins se trouvent dans le fichier **bootstrap.min.js**. Vous n'avez donc besoin que de 2 références, une pour **jQuery** et l'autre pour le fichier de **Bootstrap** :

Code : HTML

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

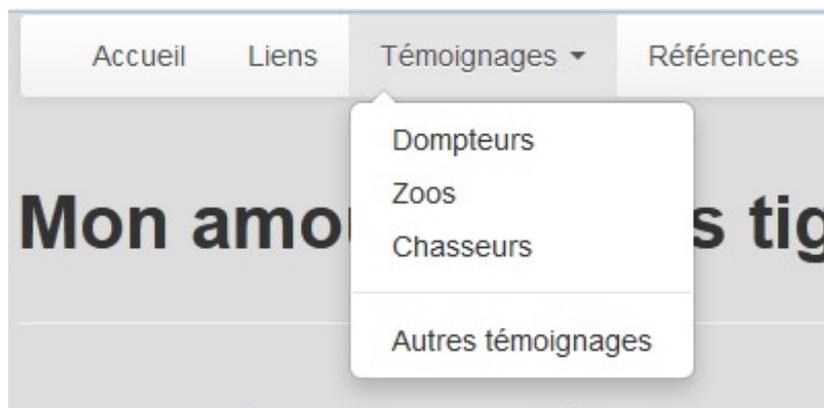
En adaptant évidemment le chemin à l'architecture de votre site. Si vous n'utilisez qu'un ou deux plugins il peut être plus intéressant d'utiliser les fichiers indépendants que vous pouvez trouver [ici](#).

Lorsqu'il est possible d'utiliser les plugins avec des data attributs, je le fais dans les exemples. Je réserve l'utilisation de **jQuery** lorsque c'est indispensable.

Menu déroulant

Menu déroulant dans la barre de navigation

Le plugin **jQuery Dropdowns** permet de faire de jolis menus déroulants de ce genre :



Il suffit de modifier la barre de navigation pour ajouter le menu déroulant :

Code : HTML

```
<div class="navbar">
  <div class="navbar-inner">
    <div class="container">
      <ul class="nav">
        <li> <a href="#">Accueil</a> </li>
        <li> <a href="#">Liens</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="#">Références</a> </li>
    </ul>
    <form class="navbar-search pull-right" action="">
      <input class="search-query" type="text"
placeholder="Recherche">
    </form>
  </div>
</div>
</div>

```

Vous pouvez voir la page correspondante [ici](#).

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.

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">
  <a class="btn" data-toggle="dropdown" href="#"> Action</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>
</div>

```

Vous remarquez l'utilisation de la classe **btn-group** nécessaire au fonctionnement du plugin mais pour le reste c'est pratiquement identique au code qu'on a vu pour la barre de navigation. Voici le résultat à l'écran (et toujours un petit clic sur l'image pour ouvrir l'exemple) :



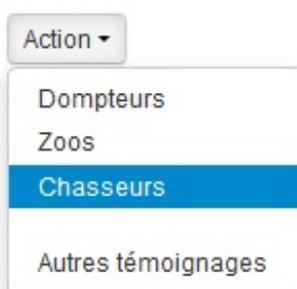
C'est pas mal, mais comme ça on comprend pas vraiment que c'est un bouton spécial, on peut avoir un repère graphique ?

On peut en ajoutant deux classes :

Code : HTML

```
<div class="btn-group"> <a class="btn dropdown-toggle" data-
toggle="dropdown" href="#"> Action <span class="caret"></span> </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>
</div>
```

J'ai ajouté les classes **dropdown-toggle** et **caret** pour le bouton et maintenant celui-ci est bien identifié :

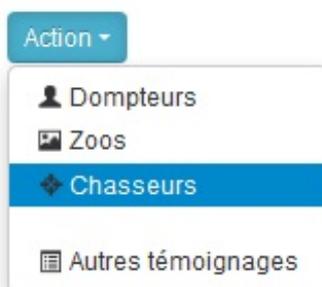


C'est plus joli, mais on peut aussi ajouter des icônes et avoir de la couleur ?

On peut... voici un exemple :

Code : HTML

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



On peut même faire en sorte que le bouton ait deux usages : en tant que bouton pour une action et en tant que déclencheur de

menu déroulant. Il est alors séparé en deux parties : à gauche pour l'action, à droite pour le menu :

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="#"><i class="icon-user"></i> Dompteurs</a></li>
    <li><a href="#"><i class="icon-picture"></i> Zoos</a></li>
    <li><a href="#"><i class="icon-screenshot"></i>
Chasseurs</a></li>
    <li class="divider"></li>
    <li><a href="#"><i class="icon-list-alt"></i> Autres
témoignages</a></li>
  </ul>
</div>
```



L'astuce consiste en fait à créer deux boutons accolés.

Sous-menu

Avec la version 2.1 on peut enfin créer un sous-menu avec la classe **dropdown-submenu**. Voici un exemple avec un bouton :

Code : HTML

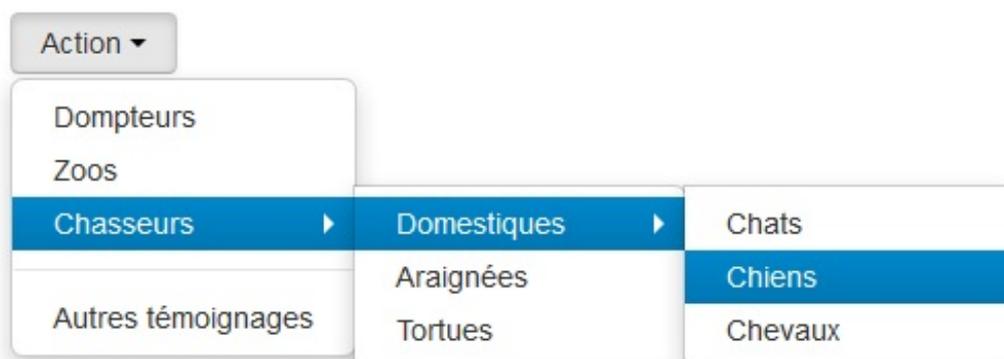
```
<div class="btn-group"> <a class="btn dropdown-toggle" data-
toggle="dropdown" href="#"> Action <span class="caret"></span> </a>
  <ul class="dropdown-menu">
    <li><a href="#">Dompteurs</a></li>
    <li><a href="#">Zoos</a></li>
  <li class="dropdown-submenu">
    <a href="#">Chasseurs</a>
    <ul class="dropdown-menu">
      <li><a href="#">Fauves</a></li>
      <li><a href="#">Araignées</a></li>
      <li><a href="#">Tortues</a></li>
    </ul>
  </li>
  <li class="divider"></li>
  <li><a href="#">Autres témoignages</a></li>
</ul>
</div>
```



J'ai eu la curiosité de voir si on peut créer un sous-sous-menu :

Code : HTML

```
<div class="btn-group"> <a class="btn dropdown-toggle" data-
toggle="dropdown" href="#"> Action <span class="caret"></span> </a>
<ul class="dropdown-menu">
<li><a href="#">Dompteurs</a></li>
<li><a href="#">Zoos</a></li>
<li class="dropdown-submenu">
<a href="#">Chasseurs</a>
<ul class="dropdown-menu">
<li class="dropdown-submenu">
<a href="#">Domestiques</a>
<ul class="dropdown-menu">
<li><a href="#">Chats</a></li>
<li><a href="#">Chiens</a></li>
<li><a href="#">Chevaux</a></li>
</ul>
</li>
<li><a href="#">Araignées</a></li>
<li><a href="#">Tortues</a></li>
</ul>
</li>
<li class="divider"></li>
<li><a href="#">Autres témoignages</a></li>
</ul>
</div>
```



Bon ça ne marche pas vraiment, le deuxième sous-menu s'ouvre tout de suite et reste ouvert 🤔.

Fenêtre modale

Le plugin **Modals** permet de créer facilement une fenêtre modale.

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

Code : HTML

```

<div class="modal hide" id="infos">
  <div class="modal-header"> <a class="close" data-
dismiss="modal">x</a>
  <h3>Plus d'informations</h3>
  </div>
  <div class="modal-body">
    <p>Le Tigre (Panthera tigris) est un mammifère carnivore de la
famille des félidés...</p>
  </div>
</div>
<a class="btn btn-primary" data-toggle="modal" href="#infos" >Plus
d'informations</a>

```

Le principe est simple : on crée un `<div>` avec la classe **modal** qui se charge de configurer et positionner la fenêtre et la classe **hide** dont le seul but est de la rendre invisible. On peut ensuite organiser la fenêtre avec 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 la classe **close** et la propriété **data-dismiss="modal"**. Il ne reste plus ensuite 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 ce bouton à l'écran :



Et quand on clique, on obtient la fenêtre modale :



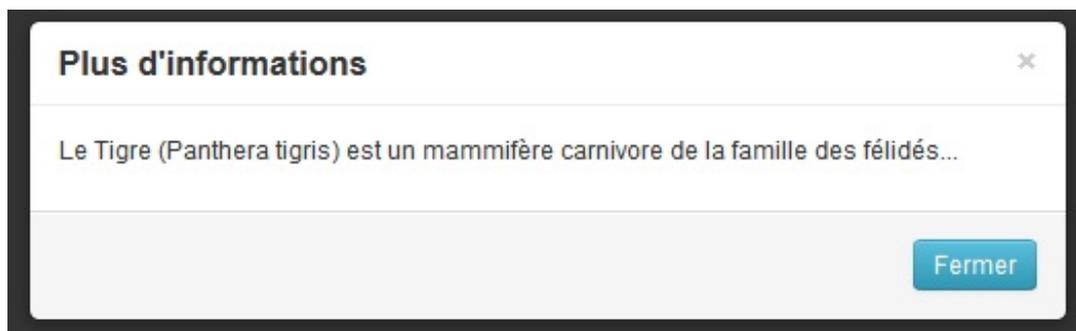
On a une petite croix pour fermer la fenêtre. On peut améliorer un peu l'esthétique en créant un pied de page avec la classe **modal-footer** et un bouton pour sortir :

Code : HTML

```

<div class="modal hide" id="infos">
  <div class="modal-header"> <a class="close" data-
dismiss="modal">x</a>
  <h3>Plus d'informations</h3>
  </div>
  <div class="modal-body">
    <p>Le Tigre (Panthera tigris) est un mammifère carnivore de la
famille des félidés...</p>
  </div>
  <div class="modal-footer"> <a class="btn btn-info" data-
dismiss="modal">Fermer</a> </div>
</div>
<a class="btn btn-primary" data-toggle="modal" href="#infos" >Plus
d'informations</a>

```



Il est possible également d'obtenir un effet de transition (pensez à référencer le plugin **transition** si vous utilisez des fichiers individuels). Il suffit d'ajouter la classe **fade** :

Code : HTML

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

Testez!



On peut aussi depuis la version 2.1 injecter une page externe dans le corps de la fenêtre modale

Cette possibilité peut s'avérer très pratique. Voici un exemple en injectant une page d'exemple sur les tigres :

Code : HTML

```
<div class="modal hide fade" id="infos">
  <div class="modal-header"> <a class="close" data-
dismiss="modal">x</a>
  <h3>Plus d'informations</h3>
</div>
<div class="modal-body">
</div>
<div class="modal-footer"> <a class="btn btn-info" data-
dismiss="modal">Fermer</a> </div>
</div>
<a class="btn btn-primary" data-toggle="modal" href="tuto02_01.html"
data-target="#infos">Plus d'informations</a>
```



On pointe la fenêtre modale avec **data-target** et la page à injecter avec **href**.

Onglets

Le plugin **Togglable tabs** permet de créer facilement des onglets.

Voici un premier exemple simple :

Code : HTML

```
<div class="tabbable">
  <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="#témoignages" 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="témoignages">Tous les
témoignages</div>
  </div>
</div>
```



On utilise une liste et quelques classes, rien de bien compliqué. Le plugin est activé par `data-toggle="tab"`.

Si cette apparence tabulaire ne vous convient pas vous pouvez changer pour un autre aspect avec une modification minimale :

Code : HTML

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



Si ça ne vous convient encore pas (mais vous êtes difficile là 😊) vous pouvez empiler les liens :

Code : HTML

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



Vous en voulez encore plus ! Un menu déroulant pour un lien ? Là vous exagérez vraiment, mais bon il faut un peu modifier le code pour utiliser le plugin `dropdown` que nous avons déjà vu :

Code : HTML

```
<div class="tabbable">
  <ul class="nav nav-tabs">
    <li class="dropdown"><a href="#accueil" data-
toggle="tab">Accueil</a></li>
    <li class="dropdown"> <a class="dropdown-toggle"
data-toggle="dropdown"
href="#"> Livres <b class="caret"></b> </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="#témoignages" 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="témoignages">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>
</div>

```



Bon, ça ne vous va encore pas et vous préféreriez les liens empilés à gauche ? Encadrez le code d'un `<div>` avec ces classes :

Code : HTML

```

<div class="tabbable tabs-left">
  ...
</div>

```



La version 2.1 permet d'aller plus loin dans l'organisation des onglets comme par exemple les placer au-dessous du contenu avec la classe `tabs-below` :

Code : HTML

```

<div class="tabbable tabs-below">
  <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="témoignages">Tous les
témoignages</div>
  </div>
  <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="#témoignages" data-
toggle="tab">Témoignages</a></li>
  </ul>
</div>

```



Ou encore à droite avec la classe **tabs-right** :

Code : HTML

```
<div class="tabbable tabs-right">
```



Carousel

Le plugin **Carousel** permet de créer facilement un carousel.

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="span9">
  <div class="carousel slide">
    <div class="carousel-inner">
      <div class="item active"> 
      <div class="carousel-caption">
        <p>Une présentation</p>
      </div>
    </div>
    <div class="item"> 
      <div class="carousel-caption">
        <p>Un autre présentation</p>
      </div>
    </div>
    <div class="item"> 
      <div class="carousel-caption">
        <p>Et encore une autre !</p>
      </div>
    </div>
  </div>
</div>
```

```

        </div>
    </div>
</div>
</div>
</div>

```

Pour chaque item on place une balise `` pour référencer une image, on associe une légende avec une balise `<div>` avec la classe `carousel-caption`. On utilise la classe `active` pour désigner l'image qui doit apparaître en premier. On a utilisé une classe `span9` pour positionner le carousel. On a dimensionné les 3 images aux mêmes côtes.



La version 2.1 impose l'utilisation de la classe "slide" pour obtenir un effet de glissement

Il nous faut aussi initialiser le carousel avec **jQuery** :

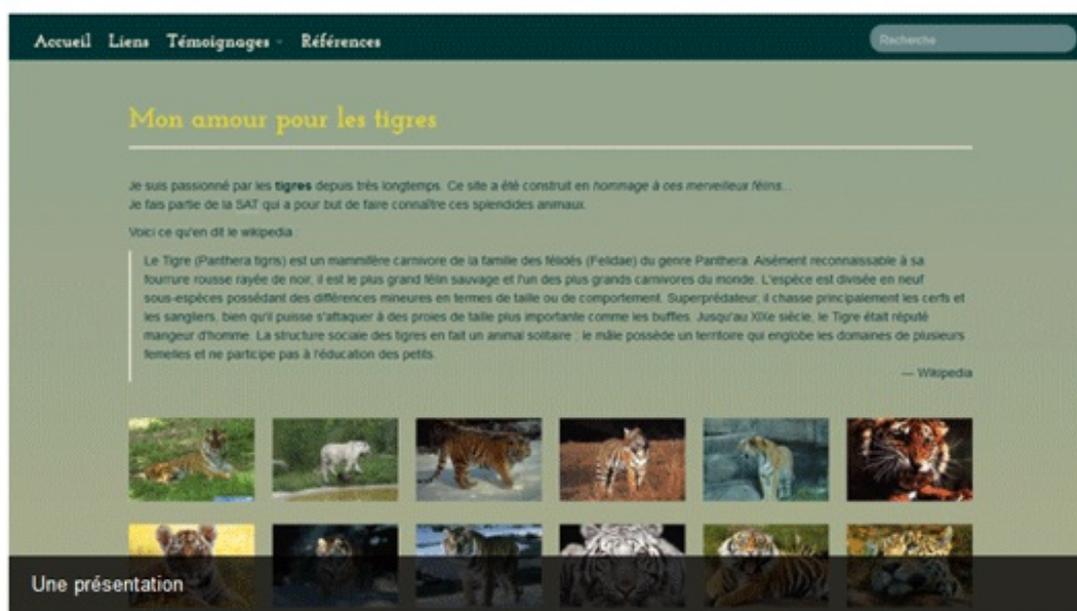
Code : JavaScript

```

$(function () {
    $('.carousel').carousel();
});

```

Voici le résultat :



On peut ajouter des boutons pour faire défiler manuellement ?

Il suffit de compléter le code avec ça :

Code : HTML

```

<div id="myCarousel" class="carousel slide">
  <div class="carousel-inner">
    <div class="item active"> 
      <div class="carousel-caption">
        <p>Une présentation</p>
      </div>
    </div>
  </div>

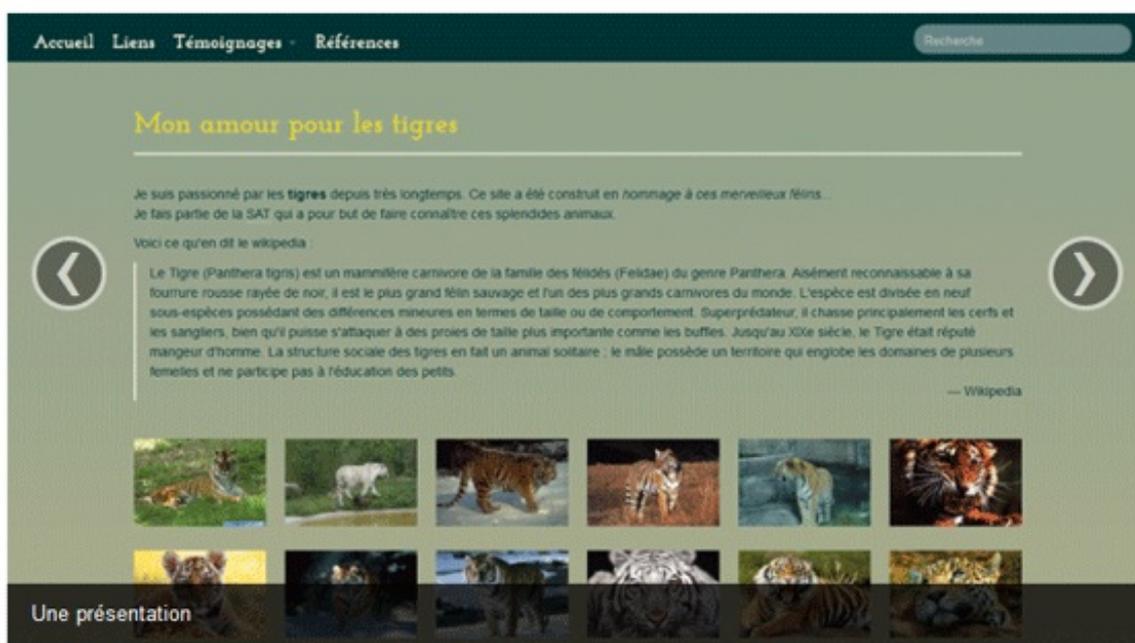
```

```

<div class="item"> 
  <div class="carousel-caption">
    <p>Un autre présentation</p>
  </div>
</div>
<div class="item"> 
  <div class="carousel-caption">
    <p>Et encore une autre !</p>
  </div>
</div>
</div>
<a class="carousel-control left" data-slide="prev"
href="#myCarousel"></a>
<a class="carousel-control right" data-slide="next"
href="#myCarousel"></a> </div>
</div>

```

On a identifié le carousel avec un l'id **myCarousel**. On a créé deux liens qui pointent sur le carousel en spécifiant la bonne classe et le bon **data-slide** :



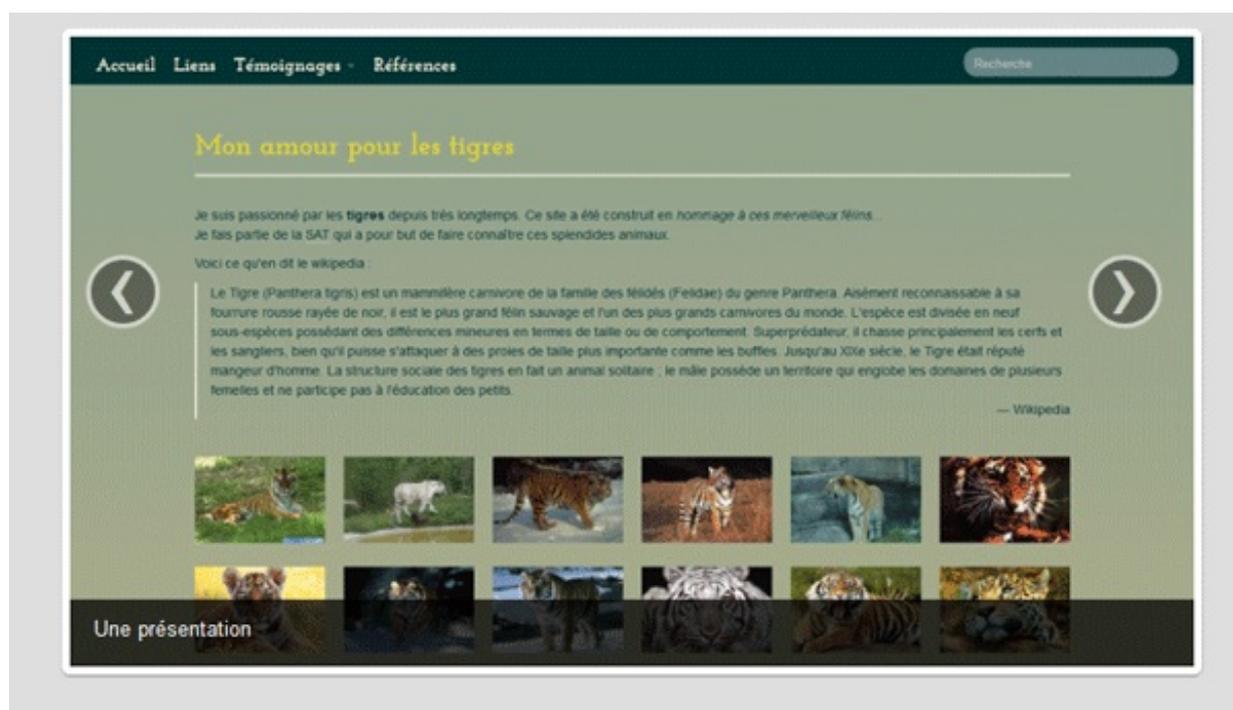
On peut améliorer la présentation en ajoutant du style :

Code : CSS

```

body {
  background-color: #ddd;
}
.carousel-inner {
  border-radius: 5px 5px 5px 5px;
  border: 5px solid white;
  box-shadow: 0 1px 1px rgba(0, 0, 0, 0.3);
}

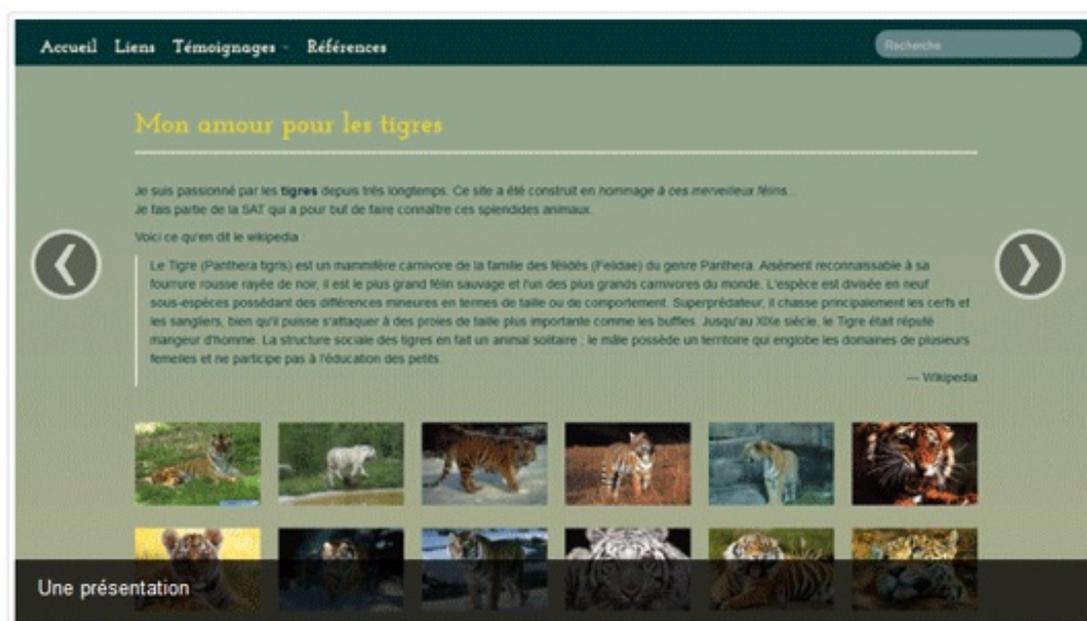
```



Mais on peut aussi détourner une classe de **Bootstrap** pour obtenir un effet sympathique sans ajouter de style :

Code : HTML

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



Le carousel comporte quelques options, par exemple le délai de défilement :

Code : JavaScript

```
$('.carousel').carousel({
  interval: 2000
});
```

Testez !

Pour terminer un petit TP. On veut obtenir ce résultat (comme depuis le début un petit clic sur l'image pour ouvrir l'exemple 😊):



Un défilement avec un délai de 2 secondes, un bouton pour arrêter le défilement et un autre pour le lancer, et en prime une étiquette qui nous indique l'état. Essayez de le faire en vous référant à [la documentation](#) pour les fonctions à utiliser.

Secret (cliquez pour afficher)

Code : HTML

```
<body>
<div class="container"> <br/>
  <div class="row">
    <div class="span9">
      <div id="myCarousel" class="carousel">
        <div class="carousel-inner thumbnail">
          <div class="item active"> 
          <div class="carousel-caption">
            <p>Une présentation</p>
          </div>
        </div>
        <div class="item"> 
        <div class="carousel-caption">
          <p>Un autre présentation</p>
        </div>
        <div class="item"> 
        <div class="carousel-caption">
          <p>Et encore une autre !</p>
        </div>
      </div>
      <a class="carousel-control left" data-slide="prev"
href="#myCarousel"></a> <a class="carousel-control right" data-
slide="next" href="#myCarousel"></a> </div>
    </div>
    <div class="span3 btn-group">
      <button id="pause" class="btn btn-warning pull-right"
>Pause</button>
      <button id="reprise" class="btn btn-info pull-right"
>Reprise</button>
    </div>
```

```

    </div>
    <span class="label label-info">Mode cycle</span> </div>
  </body>
</body>
<script src="../js/jquery.js"></script>
<script src="../bootstrap/js/bootstrap-carousel.js"></script>
<script src="../bootstrap/js/bootstrap-transition.js"></script>
<script>
$(function () {
    $('.carousel').carousel({
        interval: 2000
    });
    $("#pause").click(function() {
        $('.carousel').carousel('pause');
        $('.label').text('Mode pause').removeClass("label-
info").addClass("label-warning");
    });
    $("#reprise").click(function() {
        $('.carousel').carousel('cycle');
        $('.label').text('Mode cycle').removeClass("label-
warning").addClass("label-info");
    });
});
</script>

```

Je ne commente encore pas le code **jQuery** utilisé qui ne fait pas l'objet de ce cours.

Info-bulles et Popover

Ces deux plugins partagent une bibliothèque et ont un fonctionnement analogue, nous allons donc les traiter ensemble.

Info-bulle

Le but est d'obtenir de jolies info-bulles de ce genre :

Classe de vertébrés

ra tigris) est un [mammifère](#) carnivo
; grand félin sauvage et l'un des plu
lle ou de comportement.

Voici un premier exemple de mise en œuvre au niveau **HTML** :

Code : HTML

```

<p>Le Tigre (Panthera tigris) est un <a rel="tooltip" href="#" data-
original-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 rel="tooltip" href="#" data-original-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.</p>

```

On utilise des balises `<a>` avec l'attribut `rel` égal à `tooltip`.

Le contenu de l'info-bulle se place dans l'attribut **data-original-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.

Voici le résultat (vous pouvez tester en cliquant sur l'image, mais bon depuis le début vous avez compris ça 😊) :



Sympa non ?

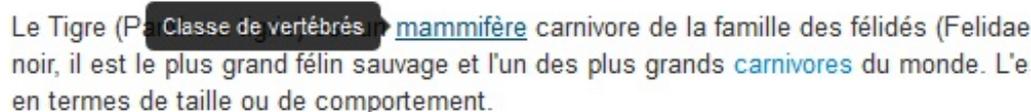
Il existe quelques options dont vous pouvez trouver [la liste complète sur le site de l'éditeur](#).

On peut par exemple changer le positionnement de l'info-bulle :

Code : JavaScript

```
$(function () {  
    $('a:first-child').tooltip({placement:'left'});  
    $('a:last-child').tooltip();  
});
```

Ici on utilise l'option **placement** en la définissant à **left** pour le premier lien et sans rien préciser pour le second (ce qui donne le positionnement par défaut au-dessus) :



Popover

Le but est d'obtenir une grosse info-bulle de ce genre :



Voici un premier exemple de mise en œuvre au niveau **HTML** :

Code : HTML

```
<div class="well"> <a href="#" id="pop" class="btn btn-info"
rel="popover" data-content="C'est tout simple à faire !"
data-original-title="Test du Popover">
Cliquez sur moi pour le Popover</a> </div>
```

On utilise des balises `<a>` avec l'attribut **rel** égal à **popover**.

Le contenu de l'info-bulle se place dans l'attribut **data-content** et le titre dans l'attribut **data-original-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**.

Voici le résultat :



Le comportement par défaut a changé avec la version 2.1. Auparavant le déclenchement par défaut était au passage du curseur de la souris. Maintenant elle se fait au clic !

Comme pour l'info-bulle on peut modifier le positionnement avec l'option **placement** :

Code : JavaScript

```
$("#pop").popover({placement: 'left'});
```



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 **focus** sur l'élément. On montre aussi qu'on peut cumuler plusieurs options puisqu'on a aussi défini le positionnement en bas avec **bottom**.



On peut aussi définir un délai d'apparition et de disparition :

Code : JavaScript

```
$("#pop").popover({delay: { show: 800, hide: 300 }});
```

[Tester !](#)

On peut aussi 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

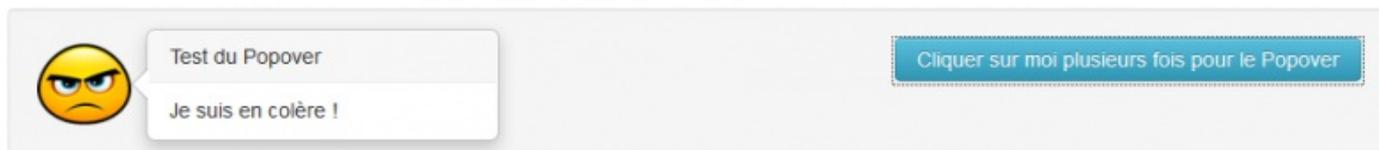
```
$("#img").popover({trigger: 'manual'});
$("#pop").click(function() {
    $('#img').popover('toggle');
});
```

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');
```

Et voici le résultat :



J'espère que je vous ai donné envie de vous amuser avec ces composants 😊.

Boutons

Un plugin est consacré aux boutons. Il permet quelques effets et groupements.

Bouton bascule

On a parfois besoin de boutons à deux états stabilisés : repos et appuyé. Le plugin permet de réaliser cela facilement. Il suffit de créer le bouton en prévoyant `data-toggle="button"` :

Code : HTML

```
<button type="button" class="btn btn-success" data-toggle="button">Simple Bascule</button>
```



Personnellement je ne trouve pas le changement très net, comme j'aime les effets tranchés je vous propose de changer aussi la classe du bouton lors du clic avec un peu de **JavaScript**. Cette fois on se contente de placer un bouton sans artifice :

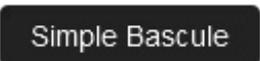
Code : HTML

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

Et on réalise tout avec **JavaScript**. C'est la méthode `toggle` qui génère l'effet, le reste du code est consacré au changement de la classe :

Code : JavaScript

```
var state;
$(function () {
  $('button').click(function () {
    with($(this)) {
      button('toggle');
      if(state = !state) removeClass('btn-primary').addClass('btn-inverse');
      else removeClass('btn-inverse').addClass('btn-primary');
    }
  });
});
```



Boutons effet "checkbox"

On peut grouper des boutons avec la classe `btn-group` et les faire fonctionner comme des "checkbox" avec `data-toggle="buttons-checkbox"`. Les boutons restent indépendants dans leur fonctionnement mais ont l'avantage d'être groupés :

Code : HTML

```
<div class="btn-group" data-toggle="buttons-checkbox">
  <button type="button" class="btn btn-success">Un</button>
  <button type="button" class="btn btn-success">Deux</button>
  <button type="button" class="btn btn-success">trois</button>
</div>
```



Ici encore si l'effet visuel ne vous convainc pas vous pouvez accentuer le résultat avec un peu de **JavaScript** :

Code : JavaScript

```
var state;
$(function () {
  $('button').click(function () {
    with($(this)) {
      if(hasClass('btn-success')) removeClass('btn-
success').addClass('btn-inverse');
      else removeClass('btn-inverse').addClass('btn-success');
    }
  })
});
```



Bon là j'ai encore fait dans la caricature visuelle mais c'est juste à visée pédagogique 😊.

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-radio"**. Les boutons sont maintenant liés dans leur fonctionnement, un seul peut être enfoncé :

Code : HTML

```
<div class="btn-group" data-toggle="buttons-radio">
  <button type="button" class="btn btn-success">Un</button>
  <button type="button" class="btn btn-success">Deux</button>
  <button type="button" class="btn btn-success">trois</button>
</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 :

Code : JavaScript

```
$(function () {  
  $('button').click(function () {  
    with($(this)) {  
      button('loading');  
      setTimeout(function () {  
        button('reset');  
      }, 4000);  
    }  
  })  
});
```



On peut aussi améliorer l'aspect visuel en jouant avec la classe du bouton comme nous l'avons fait précédemment :

Code : JavaScript

```
$(function () {  
  $('button').click(function () {  
    with($(this)) {  
      removeClass('btn-primary').addClass('btn-  
danger').button('loading');  
      setTimeout(function () {  
        removeClass('btn-danger').addClass('btn-  
primary').button('reset');  
      }, 4000);  
    }  
  })  
});
```



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

```
<button type="button" class="btn btn-primary" data-loading-  
text="Chargement en cours...">Chargez l'image !</button>  
<img id="mon_image">
```

Et voici le code **JavaScript** pour gérer l'effet :

Code : JavaScript

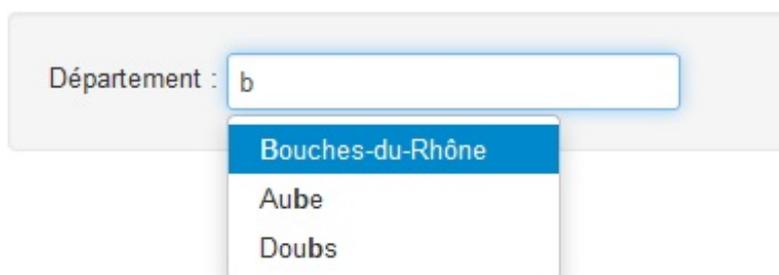
```
$(function () {  
  $('button').click(function () {  
    with($(this)) {  
      if(hasClass('btn-primary')) {  
        removeClass('btn-primary').addClass('btn-  
danger').button('loading');  
        var image = new Image();  
        image.onload = function () {  
          removeClass('btn-danger').addClass('btn-  
primary').button('reset');  
          $("#mon_image").attr({src: "images/legumes.jpg"});  
        };  
        image.src = "images/legumes.jpg";  
      }  
    }  
  })  
});
```

Chargez l'image !

Avec cet exemple vous ne verrez certainement l'effet qu'une fois dans votre navigateur étant donné qu'ensuite l'image sera en cache.

Typeahead

Le plugin **jQuery Typeahead** permet de créer un effet de menu déroulant d'aide au remplissage d'une zone de texte :



Voici un exemple de mise en œuvre :

Code : HTML

```
<div class="container"> <br/>  
  <div class="span5">  
    <form class="form-inline well">  
      <label for="départements">Département :</label>  
      <input id="départements" placeholder="Entrez un département" data-  
provide="typeahead">  
    </form>  
  </div>  
</div>  
</body>  
<script src="../js/jquery.js"></script>
```

```
<script src="../../bootstrap/js/bootstrap-typeahead.js"></script>
<script>
$(function () {
  var deps = ['Ain', 'Aisne', 'Allier', 'Alpes de-Htes', 'Alpes de-Htes
Provence', 'Hautes-Alpes',
  'Alpes-
Maritimes', 'Ardèche', 'Ardennes', 'Ariege', 'Aube', 'Aude', 'Aveyron', 'Bouches-
du-Rhône', 'Calvados',
  'Cantal', 'Charente', 'Charente-Maritime', 'Cher', 'Correze', 'Côte d\'Or',
  'Côtes d\'Armor', 'Creuse',
  'Dordogne', 'Doubs', 'Drome', 'Eure', 'Eure-et-Loire', 'Finistère'];
  $('#départements').typeahead({source: deps});
});
</script>
```

Rien de bien compliqué, il faut prévoir l'attribut **data-provide** dans la balise `<input>`.

Au niveau du script il faut créer un tableau avec les valeurs de référence, ici j'ai créé une liste de départements.

Il faut ensuite utiliser la méthode **typeahead** en déclarant la source de données.

Une option qui peut s'avérer intéressante est **items**, elle permet de définir le nombre maximal de propositions affichées. Par exemple :

Code : JavaScript

```
$('#départements').typeahead({source: deps, items:3});
```

Ici on a limité à 3 le nombre de propositions. [Tester ici](#).

Effet accordéon

Le plugin **jQuery collapse** permet de faire de jolis effets d'apparition et de disparition de zones (un clic sur l'image pour voir) :



On va commencer par un exemple simple :

Code : HTML

```
<div class="container"> </br>
  <div class="span6 well">
    <h4> Les plugins de Bootstrap</h4>
    <br>
    <div class="accordion-heading"> <a class="accordion-toggle"
href="#item1" data-toggle="collapse"> Accordéon </a> </div>
    <div id="item1" class="collapse">
```

```

    <div class="accordion-inner"> Ce plugin permet de créer des
    effets "accordéon" totalement paramétrables</div>
  </div>
  <div class="accordion-heading"> <a class="accordion-toggle"
href="#item2" data-toggle="collapse"> Fenêtre modale </a> </div>
  <div id="item2" class="collapse">
    <div class="accordion-inner"> Ce plugin permet de créer des
    fenêtres modales élégantes avec une grande simplicité. </div>
  </div>
  <div class="accordion-heading"> <a class="accordion-toggle"
href="#item3" data-toggle="collapse"> Carousel </a> </div>
  <div id="item3" class="collapse">
    <div class="accordion-inner"> 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>

```

C'est le code qui correspond à l'image située au-dessus. 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** et comporter la propriété **data-toggle="collapse"**
- La **zone** à masquer ou 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.

Il y a deux classes dévouées à l'esthétique du contrôle : **accordion-heading** pour le déclencheur et **accordion-inner** pour la zone. Si on les enlève, la mise en forme est moins convaincante mais ça fonctionne encore :



On peut encore améliorer l'esthétique en créant des bordures pour bien délimiter les éléments :

Code : HTML

```

<div class="container"> </br>
  <div class="span6 well">
    <h4> Les plugins de Bootstrap</h4>
    <br>
    <div class="accordion-group">
      <div class="accordion-heading"> <a class="accordion-toggle"
href="#item1" data-toggle="collapse"> Accordéon </a> </div>
      <div id="item1" class="collapse in">
        <div class="accordion-inner"> Ce plugin permet de créer des
        effets "accordéon" totalement paramétrables</div>
      </div>
    </div>
    <div class="accordion-group">
      <div class="accordion-heading"> <a class="accordion-toggle"
href="#item2" data-toggle="collapse"> Fenêtre modale </a> </div>
      <div id="item2" class="collapse">
        <div class="accordion-inner"> Ce plugin permet de créer des

```

```
fenêtres modales élégantes avec une grande simplicité. </div>
</div>
</div>
<div class="accordion-group">
  <div class="accordion-heading"> <a class="accordion-toggle"
href="#item3" data-toggle="collapse"> Carousel </a> </div>
  <div id="item3" class="collapse">
    <div class="accordion-inner"> 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>
</div>
```

Les plugins de Bootstrap

Accordéon

Ce plugin permet de créer des effets "accordéon" totalement paramétrables

Fenêtre modale

Carousel

Je me suis contenté d'ajouter la classe **accordion-group** à chaque bloc déclencheur-zone. Les plus attentifs auront aussi remarqué que j'ai ajouté la classe **in** à la première zone pour qu'elle soit visible dès le départ.



Oui mais j'aimerais que la zone affichée s'efface quand j'en ouvre une nouvelle...

Alors là il faut encore compléter le code :

Code : HTML

```
<div class="span6 well">
  <h4> Les plugins de Bootstrap</h4>
  <br>
  <div id="monaccordeon">
    <div class="accordion-group">
      <div class="accordion-heading"> <a class="accordion-toggle"
href="#item1" data-parent="#monaccordeon" data-toggle="collapse">
Accordéon </a> </div>
      <div id="item1" class="collapse in">
        <div class="accordion-inner"> Ce plugin permet de créer
des effets "accordéon" totalement paramétrables</div>
      </div>
    </div>
    <div class="accordion-group">
      <div class="accordion-heading"> <a class="accordion-toggle"
href="#item2" data-parent="#monaccordeon" data-toggle="collapse">
Fenêtre modale </a> </div>
      <div id="item2" class="collapse">
        <div class="accordion-inner"> Ce plugin permet de créer
des fenêtres modales élégantes avec une grande simplicité. </div>
      </div>
    </div>
  </div>
```

```

    <div class="accordion-group">
<div class="accordion-heading"> <a class="accordion-toggle"
href="#item3" data-parent="#monaccordeon" data-toggle="collapse">
Carousel </a> </div>
    <div id="item3" class="collapse">
        <div class="accordion-inner"> 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>
</div>
</div>

```

Tester !

Il a fallu englober d'un `<div>` avec un identifiant (`#monaccordeon`) et faire une référence dans les déclencheurs avec `data-parent="#monaccordeon"`.



Est-ce qu'on peut déclencher avec des boutons ?

Oui on peut mais il faut adapter le code :

Code : HTML

```

<div class="span6 well">
    <h4> Les plugins de Bootstrap</h4>
    <br>
    <div id="monaccordeon">
        <div class="accordion-group">
<button class="btn btn-primary accordion-heading" data-
toggle="collapse" data-parent="#monaccordeon" data-
target="#item1">Accordéon</button>
            <div id="item1" class="collapse accordion-group in">
                <div class="accordion-inner"> Ce plugin permet de créer
des effets "accordéon" totalement paramétrables</div>
            </div>
        </div>
        <div class="accordion-group">
<button class="btn btn-primary accordion-heading" data-
toggle="collapse" data-parent="#monaccordeon" data-
target="#item2">Fenêtre modale</button>
            <div id="item2" class="collapse accordion-group">
                <div class="accordion-inner"> Ce plugin permet de créer
des fenêtres modales élégantes avec une grande simplicité. </div>
            </div>
        </div>
        <div class="accordion-group">
<button class="btn btn-primary accordion-heading" data-
toggle="collapse" data-parent="#monaccordeon" data-
target="#item3">Carousel</button>
            <div id="item3" class="collapse accordion-group">
                <div class="accordion-inner"> 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>
</div>
</div>

```



Cette fois on fait la référence aux zones avec `data-target="#itemx"`.



Est-ce qu'on peut savoir quand une zone apparaît et l'utiliser dans du code ?

Voici un exemple illustratif avec utilisation de jQuery :

Code : HTML

```
<div class="container"> <br>
  <div class="span6 well">
    <h4> Les plugins de Bootstrap</h4>
    </br>
    <div id="monaccordeon">
      <div class="accordion-group">
        <button class="btn btn-primary accordion-heading" data-
toggle="collapse" data-parent="#monaccordeon" data-
target="#item1">Accordéon</button>
        <div id="item1" class="collapse accordion-group in">
          <div class="accordion-inner"> Ce plugin permet de créer des
effets "accordéon" totalement paramétrables</div>
        </div>
      </div>
      <div class="accordion-group">
        <button class="btn btn-primary accordion-heading" data-
toggle="collapse" data-parent="#monaccordeon" data-
target="#item2">Fenêtre modale</button>
        <div id="item2" class="collapse accordion-group">
          <div class="accordion-inner"> Ce plugin permet de créer des
fenêtres modales élégantes avec une grande simplicité. </div>
        </div>
      </div>
      <div class="accordion-group">
        <button class="btn btn-primary accordion-heading" data-
toggle="collapse" data-parent="#monaccordeon" data-
target="#item3">Carousel</button>
        <div id="item3" class="collapse accordion-group">
          <div class="accordion-inner"> 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>
    <div id="affichage"><span class="label label-
warning">"Accordéon" a été affiché !</span></div>
  </div>
</div>
<script src="../js/jquery.js"></script>
<script src="../bootstrap/js/bootstrap.min.js"></script>
```

```
<script>
$(function () {
  $("#item1").on("shown", function () {
    $("#affichage").html('<span class="label label-
warning">"Accordéon" a été affiché !</span>');
  })
  $("#item2").on("shown", function () {
    $("#affichage").html('<span class="label label-
warning">"Fenêtre modale" a été affiché !</span>');
  })
  $("#item3").on("shown", function () {
    $("#affichage").html('<span class="label label-
warning">"Carousel" a été affiché !</span>');
  })
});
</script>
```

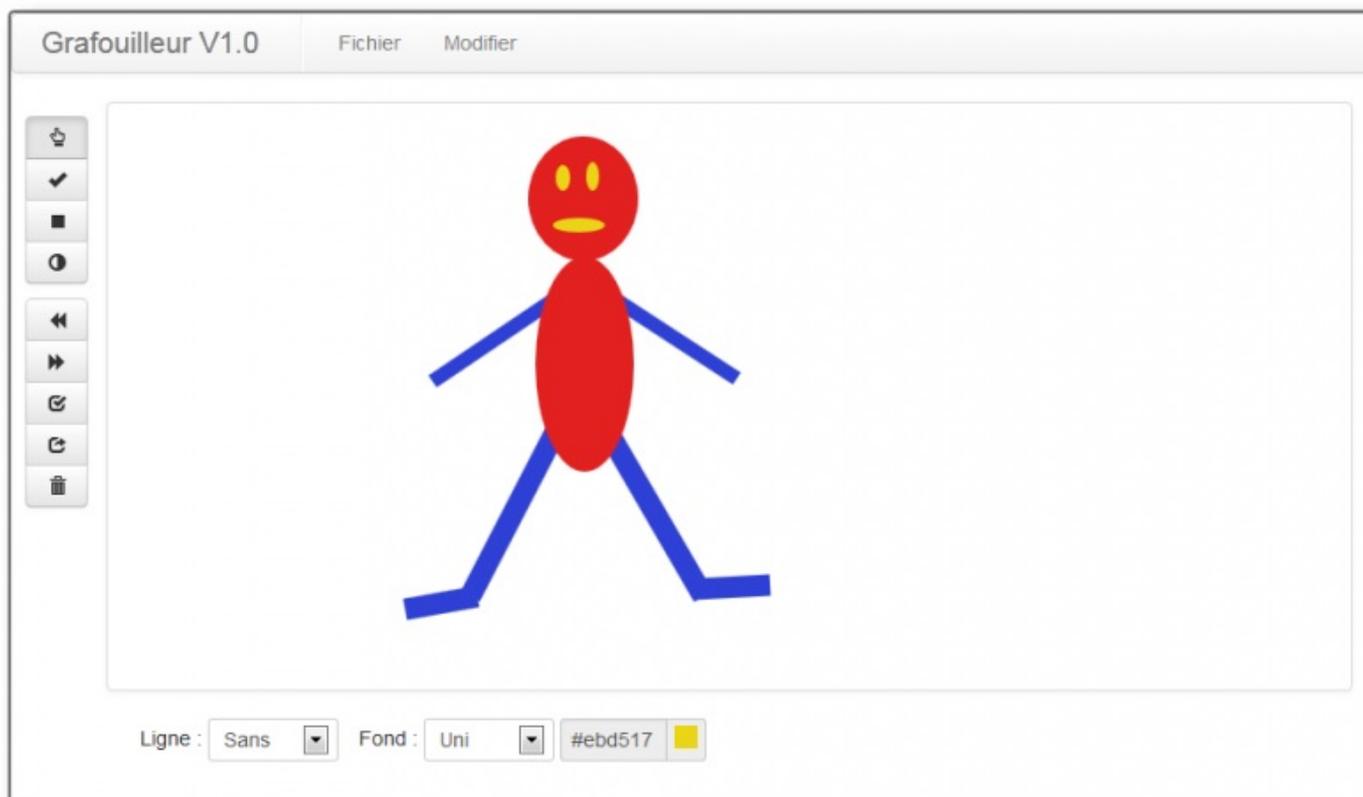


Un exemple de réalisation

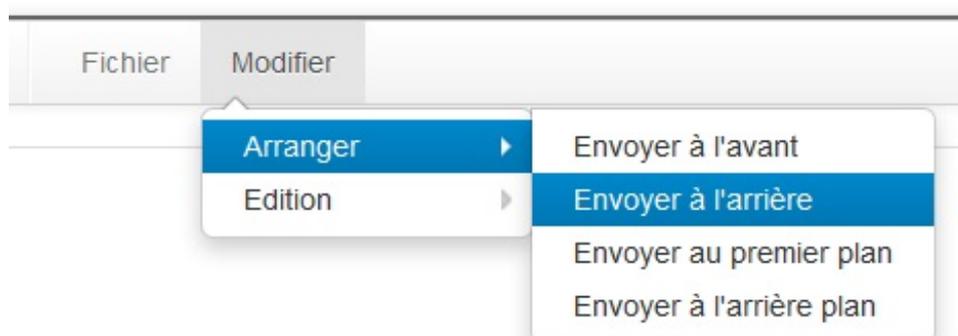
Maintenant que nous avons vu les principales possibilités de **Bootstrap** nous allons considérer une application pratique qui consiste à réaliser un modeste éditeur de dessin en ligne. Nous allons nous intéresser essentiellement à la conception de l'interface puisque c'est le sujet de ce tutoriel, mais plutôt que d'avoir une coquille vide nous allons aussi équiper cet éditeur de quelques fonctionnalités. On utilise le [canvas du HTML5](#) pour les manipulations graphiques avec l'assistance de la librairie [KineticJS](#). Cette application me sert aussi dans son aspect **Javascript** pour un tutoriel en cours de rédaction sur **KineticJS**. C'est dans ce cours là que j'explique la partie **Javascript**.

Le résultat final à l'œuvre

Voici ce que nous devons obtenir (du moins pour l'interface, pour le dessin j'espère que vous ferez mieux 🤖):



L'interface présente un menu avec des sous-menus :



Au niveau de ces menus nous avons ces fonctionnalités :

- Nouveau dessin
- Enregistrement en mode image
- Enregistrement en mode JSON
- Positionnement de la forme sélectionnée : avant, arrière, premier plan, arrière plan
- Modification : annulation dernière forme, rétablissement de la dernière forme, copier-coller, suppression d'une forme

Deux barres de boutons sur la partie gauche :



Les boutons de la première barre sont du type "radio" avec un seul bouton actif. Une aide sous forme d'info-bulle indique le nom de la commande. On peut soit être en mode sélection, soit en mode de dessin avec le choix entre ligne, rectangle et ellipse. Les boutons de la deuxième barre servent à annuler ou rétablir la dernière forme dessinée, copier-coller des formes ou les supprimer.

Et enfin une barre inférieure rassemble la sélection de la largeur et de la couleur des lignes ainsi que la couleur de remplissage :

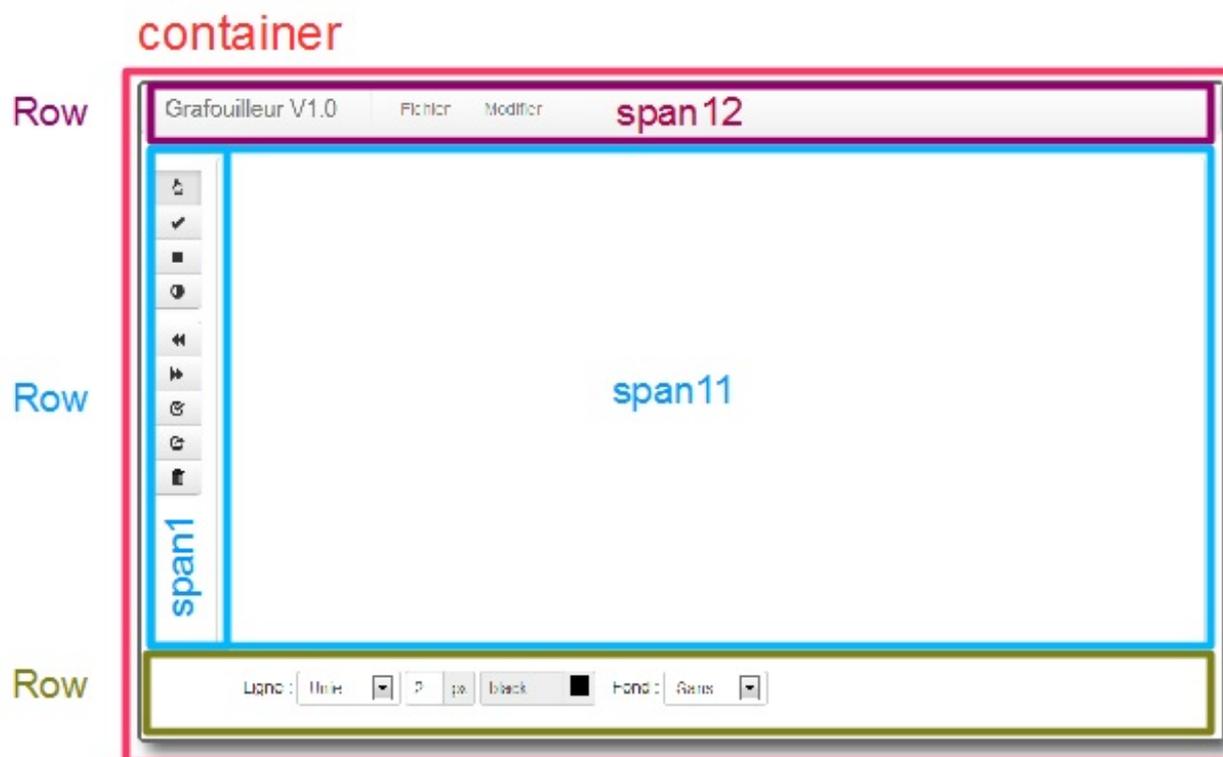


Une assistance visuelle est prévue pour le choix de la couleur.

Vous pouvez [tester l'éditeur en ligne](#) pour voir son fonctionnement.

Organisation de la page

Organisation générale



L'application de la grille de **Bootstrap** est efficace aussi pour ce genre d'application un peu éloignée d'une page web. Un **container** contient trois **row**. Dans le premier **row** un **span12** héberge la barre de menu. Le second **row** est divisé en deux avec un **span1** pour les barres de boutons et un **span11** pour accueillir les **canvas**. Enfin un dernier **row** contient les contrôles inférieurs.

Un peu de style

Pour l'esthétique j'ai traité un peu la bordure du **container** :

Code : CSS

```
.container {
  -moz-box-shadow: 8px 8px 12px #aaa;
  -webkit-box-shadow: 8px 8px 12px #aaa;
  box-shadow: 8px 8px 12px #555;
  border: 2px solid #666;
  border-radius: 4px;
  margin-top: 10px;
  height: 550px;
}
```

Donc des coins arrondis, de l'ombre et une fixation de la hauteur.

La barre de boutons nécessitait également quelques réglages :

Code : CSS

```
.btn-group {
  padding-top: 10px;
}
.btn-group-vertical .btn {
  width: 30%;
}
.btn-group + .btn-group-vertical {
  margin-left: 0;
}
.span1 {
  padding-left: 8px;
}
```

```
}
```

Juste pour réduire un peu la largeur des boutons et les positionner correctement. Pour assurer aussi l'alignement correct des deux barres de boutons.

Enfin au niveau de la barre inférieure aussi quelques petits réglages de positionnement des éléments de formulaire et une réduction de la zone de saisie de la largeur de ligne :

Code : CSS

```
form {
  margin-top: 10px;
  margin-left: 90px;
}
#width {
  max-width: 20px;
}
label {
  margin-left: 10px;
}
```

Pour terminer une annulation de la bordure interne supérieure des **canvas** pour lesquels on détourne la classe **thumbnail** pour réaliser à bon compte une bordure élégante :

Code : CSS

```
.thumbnail {
  padding-bottom: 0;
}
```

Au final on se retrouve avec assez peu de code **CSS** pour cette mise en page dont l'essentiel est réalisé par **Bootstrap**.

La barre de menu

La barre de menu, placée dans une balise **nav** affublée d'une classe **span12** est du pur **Bootstrap** avec utilisation du plugin **dropdown** pour les menus déroulants et les sous-menus :

Code : HTML

```
<nav class="span12">
  <div class="navbar">
    <div class="navbar-inner"> <a class="brand"
href="#">Grafouilleur V1.0</a>
    <ul class="nav">
      <li class="divider-vertical"></li>
      <li id="fichier" class="dropdown"> <a class="dropdown-
toggle" data-toggle="dropdown" href="#"> Fichier</a>
        <ul class="dropdown-menu">
          <li> <a href="#" id="nouveau">Nouveau...</a> </li>
          <li class="divider"></li>
          <li> <a href="#" id="enregistrer_image">Enregistrer
l'image</a> </li>
          <li> <a href="#" id="enregistrer_json">Enregistrer en
JSON</a> </li>
        </ul>
      </li>
      <li id="modifier" class="dropdown"> <a class="dropdown-
toggle" data-toggle="dropdown" href="#">Modifier</a>
        <ul class="dropdown-menu">
```

```

        <li class="dropdown-submenu"> <a tabindex="-1"
href="#">Arranger</a>
        <ul class="dropdown-menu">
            <li> <a href="#" id="go_up">Envoyer à l'avant</a>
        </li>
            <li> <a href="#" id="go_down">Envoyer à
l'arrière</a> </li>
            <li> <a href="#" id="go_front">Envoyer au premier
plan</a> </li>
            <li> <a href="#" id="go_back">Envoyer à l'arrière
plan</a> </li>
        </ul>
    </li>
    <li class="dropdown-submenu"> <a tabindex="-1"
href="#">Edition</a>
    <ul class="dropdown-menu">
        <li> <a href="#" id="annuler">Annuler</a> </li>
        <li> <a href="#" id="repetier">Répéter</a> </li>
        <li> <a href="#" id="supprimer">Supprimer</a> </li>
        <li class="divider"></li>
        <li> <a href="#" id="copier">Copier</a> </li>
        <li> <a href="#" id="coller">Coller</a> </li>
    </ul>
</li>
</ul>
</li>
</ul>
</div>
</div>
</nav>

```

Chaque commande a reçu un identifiant pour être plus facilement repérable par le **Javascript**.

Les barres de boutons et les canvas

Les barres de boutons

Les barres de boutons sont enveloppées par une balise **aside** équipée de la classe **span1** :

Code : HTML

```

<aside class="span1">
    <div id="mode" class="btn-group btn-group-vertical" data-
toggle="buttons-radio">
        <a href="#" class="btn active" rel="tooltip" data-original-
title="Sélection"><i class="icon-hand-up"></i></a>
        <a href="#" class="btn" rel="tooltip" data-original-
title="Ligne"><i class="icon-ok"></i></a>
        <a href="#" class="btn" rel="tooltip" data-original-
title="Rectangle"><i class="icon-stop"></i></a>
        <a href="#" class="btn" rel="tooltip" data-original-
title="Ellipse"><i class="icon-adjust"></i></a>
    </div>
    <div id="edition" class="btn-group btn-group-vertical">
        <a href="#" class="btn" rel="tooltip" data-original-
title="Annuler"><i class="icon-backward"></i></a>
        <a href="#" class="btn" rel="tooltip" data-original-
title="Répéter"><i class="icon-forward"></i></a>
        <a href="#" class="btn" rel="tooltip" data-original-
title="Copier"><i class="icon-check"></i></a>
        <a href="#" class="btn" rel="tooltip" data-original-
title="Coller"><i class="icon-share"></i></a>
        <a href="#" class="btn" rel="tooltip" data-original-
title="Supprimer"><i class="icon-trash"></i></a>
    </div>
</aside>

```

Les boutons sont regroupés en barre avec la classe **btn-group** et en position verticale avec **btn-group-vertical**. Le plugin **tooltip** est mis en œuvre au niveau de chaque bouton pour faire apparaître l'info-bulle d'aide. J'ai utilisé les icônes de la collection de base pour ne pas sortir de **Bootstrap**, même s'il m'a fallu un peu fouiller pour faire cadrer avec les fonctionnalités. Je n'ai prévu qu'un identifiant par groupe, les boutons étant faciles à repérer avec **jQuery**.

Les canvas

Je vous parle des **canvas** depuis le début mais vous ne voyez pas cette balise dans le code, juste ça :

Code : HTML

```
<section class="span11">
  <div id="kinetic" class="thumbnail"></div>
</section>
```

Une balise **section** équipée de la classe **span11** pour la mise en page et à l'intérieur pas grand chose, juste un div avec un identifiant et la classe **thumbnail** pour avoir une jolie bordure. En fait les **canvas** sont générés dynamiquement par **KineticJS**. Si vous poussez la curiosité jusqu'à regarder le code généré :

Code : HTML

```
<section class="span11">
  <div id="kinetic" class="thumbnail">
    <div class="kineticjs-content" style="position: relative;
display: inline-block; width: 850px; height: 400px;">
      <canvas style="position: absolute; cursor: default;"
height="400" width="850"></canvas>
      <canvas style="position: absolute; cursor: default;"
height="400" width="850"></canvas>
    </div>
  </div>
</section>
```

Vous voyez maintenant ces fameux canvas 😊.

Le pied de page

Le pied de page contient dans une balise **footer** équipée d'une classe **row** des contrôles pour la saisie des valeurs d'épaisseur et couleur de ligne ainsi que de couleur de remplissage. On trouve également des possibilités de sélection (avec ou sans bordure, avec ou sans remplissage) :

Code : HTML

```
<footer class="row">
  <form class="form-inline well-small">
    <label>Ligne : </label>
    <select id="typeligne" class="input-small">
      <option>Unie</option>
      <option>Sans</option>
    </select>
    <div id="widthline" class="input-append">
      <input id="width" type="text">
      <span class="add-on">px</span> </div>
    <div id="couleurligne" class="input-append color" data-
color="#000">
      <input id="colorligne" class="input-mini" type="text"
readonly value="">
      <span class="add-on"> <i style="background-color:#000"></i>
</span> </div>
  <label>Fond : </label>
```

```

    <select id="typefond" class="input-small">
      <option>Sans</option>
      <option>Uni</option>
    </select>
    <div id="couleurfond" class="input-append color hide" data-
color="#fff">
      <input id="colorremplissage" class="input-mini" type="text"
readonly value="">
      <span class="add-on"> <i style="background-color:#fff"></i>
</span> </div>
  </form>
</footer>

```

J'ai utilisé un formulaire en ligne avec la classe **form-inline** agrémenté pour l'esthétique avec la classe **well-small**. Pour la sélection des couleurs j'ai utilisé deux fois le plugin **colorpicker** que je décris plus loin dans les extensions de **Bootstrap**.

Les pages modales

J'utilise des pages modale, avec le plugin associé, pour la confirmation d'un nouveau dessin, et aussi pour l'enregistrement au format **JSON** :

Code : HTML

```

<div class="modal hide" id="modal_new" tabindex="-1">
  <div class="modal-header">
    <h3 id="myModalLabel">Nouveau dessin</h3>
  </div>
  <div class="modal-body">
    <p>Vous êtes sûr de vouloir un nouveau dessin ?</p>
  </div>
  <div class="modal-footer">
    <button id="nouveaucanvas" class="btn btn-danger" data-
dismiss="modal">Oui</button>
    <button class="btn" data-dismiss="modal">Annuler</button>
  </div>
</div>

<div class="modal hide" id="modal_save" tabindex="-1">
  <div class="modal-header">
    <h3 id="myModalLabel">Enregistrement au format JSON</h3>
  </div>
  <div class="modal-body">
    <p id="text_json"></p>
  </div>
  <div class="modal-footer">
    <button class="btn" data-dismiss="modal">Quitter</button>
  </div>
</div>

```

Nous avons vu presque tous les plugins proposés par **Bootstrap**. La dernière étape sera de voir comment on peut modifier **Bootstrap** pour l'adapter plus finement à nos besoins.

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** commencez par lire ce chapitre 🤪

Configuration simplifiée Configurer en ligne sur le site de Bootstrap

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 :

1. Choose components

[Toggle all](#)

Scaffolding

- Normalize and reset
- Body type and links
- Grid system
- Layouts

Base CSS

- Headings, body, etc
- Code and pre
- Labels and badges
- Tables
- Forms
- Buttons
- Icons

Components

- Button groups and dropdowns
- Navs, tabs, and pills
- Navbar
- Breadcrumbs
- Pagination
- Pager
- Thumbnails
- Alerts
- Progress bars
- Hero unit

JS Components

- Tooltips
- Popovers
- Modals
- Dropdowns
- Collapse
- Carousel

Miscellaneous

- Wells
- Close icon
- Utilities
- Component animations

Responsive

- Visible/hidden classes
- Narrow tablets and below (<767px)
- Tablets to desktops (767-979px)
- Large desktops (>1200px)
- Responsive navbar

Vous sélectionnez les plugins **jQuery** que vous utilisez :

2. Select jQuery plugins

 Transitions (required for any animation)

 Modals

 Dropdowns

 Scrollspy

 Toggable tabs

 Tooltips

 Popovers (requires Tooltips)

 Affix

 Alert messages

 Buttons

 Collapse

 Carousel

 Typeahead

Heads up!

All checked plugins will be compiled into a single file, `bootstrap.js`. All plugins require the latest version of [jQuery](#) to be included.

Vous adaptez les variables LESS à votre goût :

3. Customize variables

Scaffolding

@bodyBackground

@textColor

Links

@linkColor

@linkColorHover

Colors

@blue

@green

@red

@yellow

Typography

@sansFontFamily

@serifFontFamily

@monoFontFamily

@baseFontSize

@baseFontFamily

@baseLineHeight

@altFontFamily

@headingsFontFamily

@headingsFontWeight

@headingsColor

Form states & alerts

@warningText

@warningBackground

@errorText

@errorBackground

@successText

@successBackground

@infoText

@infoBackground

Navbar

@navbarHeight

Vous n'avez plus qu'à cliquer sur le bouton pour obtenir votre Bootstrap personnalisé :

4. Download

Customize and Download

Configurer en ligne sur le site Stylebootstrap

Ici c'est encore plus pratique parce qu'on peut régler visuellement tous les changements :

StyleBootstrap.info

Create unique web design with Twitter Bootstrap **2.1** Kw: themes, templates, styles. generator. customize.

New version came out, please report bugs if you find any. Thank you.

Tweet 747

Patinka 232

Donate

Navbar Body Typography Buttons Forms Miscellaneous Example Get this style  WrapBootstrap   I am a Web/Software Developer. Will work for bacon. 

Navbar

Project name Home **Miracle** Link Link Dropdown ▾ Link Dropdown ▾

Background: Fonts: Divider:

Brand padding: Button padding: Divider height:

Brand font-size: Button font-size:

Varier les styles

Si vous n'êtes pas trop inspiré pour faire vos propres variations j'ai trouvé [un site](#) qui propose des thèmes gratuits.

Voici quelques variations de la page de test avec leurs propositions (cliquez sur les images pour ouvrir les pages dans votre navigateur) :

[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



[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 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



Accueil
Liens
Témoignages ▾
Références

Recherche

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



Un générateur de thèmes

BootTheme va encore plus loin en offrant un générateur de thèmes :

Customize

what you change is what you see

Basic **Advanced** Colors

Fonts

Scaffolding

@bodyBackground

@textColor

Links

Typography

Preview

what you see is what you get

Layout ▾ <http://twitter.github.com/bootstrap> Go

☰ 📱 📺 🖥️

↺ ↻ ↷

Project name

SIDEBAR

Link

Link

Link

Link

SIDEBAR

Link

Link

Link

Username ▾ ☰

Get Code

[CSS](#) [LESS](#)

Le générateur est très complet et à jour pour la version 2.1 😊.

Utilisation de LESS

C'est quoi LESS ?

Bootstrap est totalement paramétrable parce qu'il est basé sur **LESS**. C'est un langage qui permet d'étendre les possibilités de **CSS** avec des variables, des opérations, des fonctions. Pour en savoir plus sur ce langage, je vous renvoie au [tuto de Iso](#). On trouve les sources **LESS** de **Bootstrap** [ici](#). On trouve de nombreux fichiers, par exemple **variables.less** qui définit la valeur de toutes les variables utilisées dans les autres fichiers. En voici un extrait :

Code : CSS

```
...
// Grays
// -----
@black: #000;
@grayDarker: #222;
@grayDark: #333;
@gray: #555;
@grayLight: #999;
@grayLighter: #eee;
@white: #fff;
...
```

Il suffit de changer la valeur d'une variable ici pour que les valeurs changent pour toutes les propriétés qui utilisent cette variable. Par exemple on retrouve du gris utilisé pour les boutons :

Code : CSS

```
// Hover state
.btn:hover {
  color: @grayDark;
  text-decoration: none;
  background-color: darken(@white, 10%);
  background-position: 0 -15px;
  ...
}
```

Mais les possibilités de **LESS** sont loin de s'arrêter à la définition de variables. On peut inclure un fichier. Par exemple le fichier **bootstrap.less** qui ne fait que des inclusions :

Code : CSS

```
...
// CSS Reset
@import "reset.less";

// Core variables and mixins
@import "variables.less"; // Modify this for custom colors, font-
sizes, etc
@import "mixins.less";
...
```

On peut aussi inclure des classes dans d'autres classes comme ici dans le fichier **code.less** :

Code : CSS

```
// Inline and block code styles
code,
pre {
  padding: 0 3px 2px;
  #font > #family > .monospace;
}
```

```
font-size: @baseFontSize - 1;  
color: @grayDark;  
.border-radius(3px);  
}
```

On peut donc faire des factorisations bien pratiques 😊.

On peut même inclure des sélecteurs dans une définition comme ici dans le fichier **breadcrumbs.less** :

Code : CSS

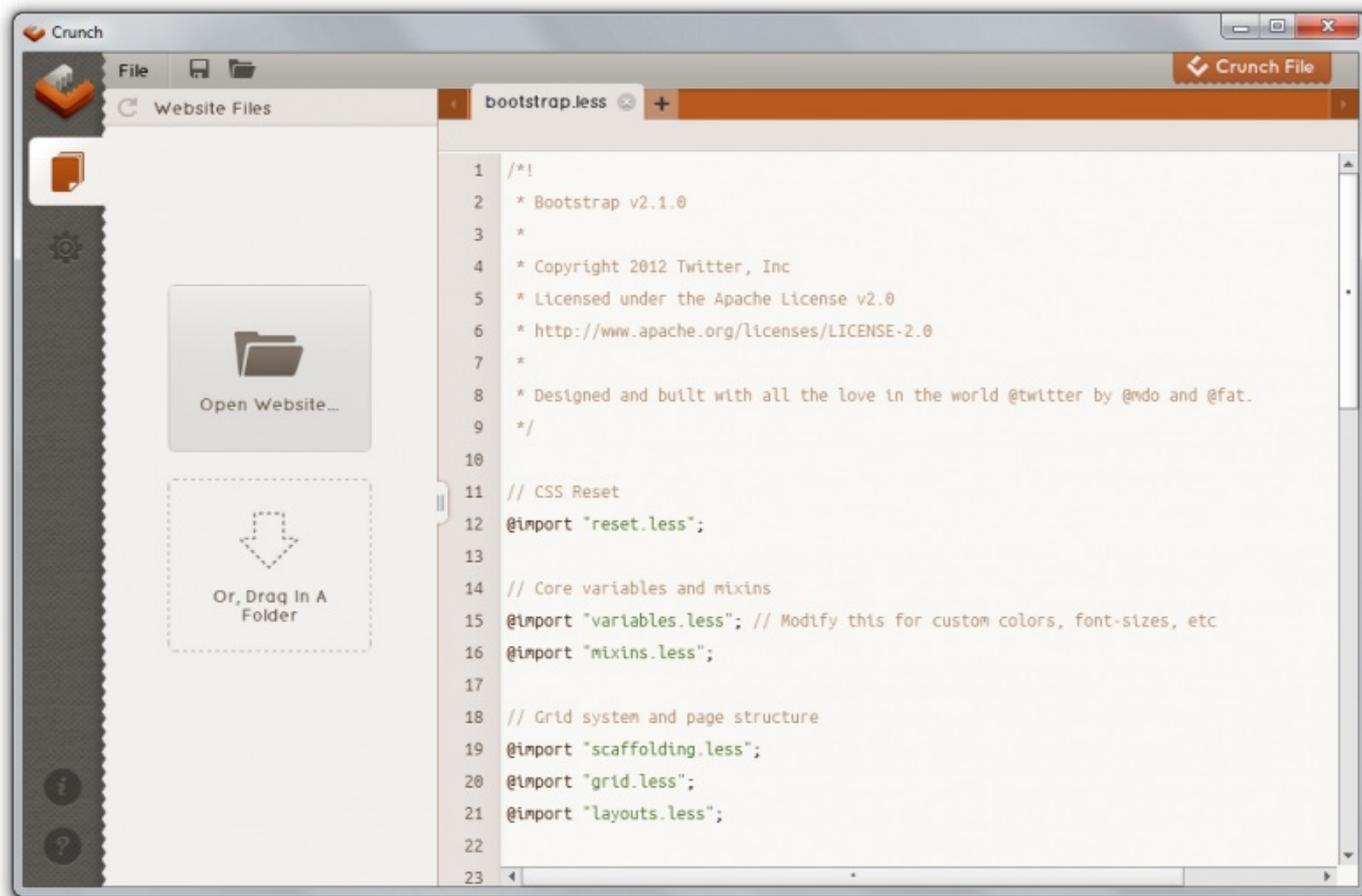
```
.breadcrumb {  
...  
li {  
display: inline-block;  
.ie7-inline-block();  
text-shadow: 0 1px 0 @white;  
}  
...  
}
```

On peut aussi effectuer des opérations comme ici dans le fichier **forms.less** :

Code : CSS

```
legend {  
...  
margin-bottom: @baseLineHeight * 1.5;  
font-size: @baseFontSize * 1.5;  
line-height: @baseLineHeight * 2;  
...  
}
```

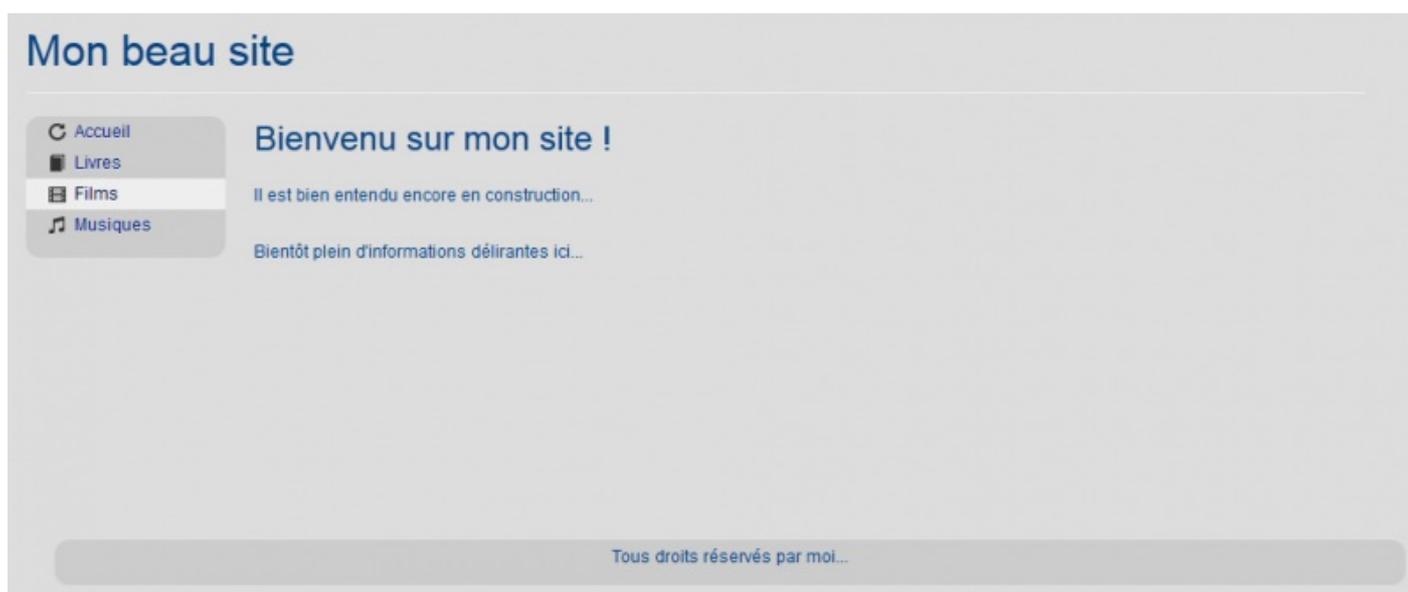
Vous pouvez modifier **Bootstrap** en tripatouillant les variables et en compilant avec les outils proposés... Pour ce tuto je vous propose d'utiliser un programme très bien fait : [crunch](#). Au moment où j'écris ce tuto, j'ai trouvé la version 1.5.3 :



Le programme est simple et efficace, vous pouvez créer ou modifier des fichiers **LESS** et les compiler en **CSS**.

Exemple simple de configuration

Supposons que nous voulons créer un site avec une certaine présentation. Quelque chose de simple parce que je veux juste vous montrer le principe de la configuration (un petit clic sur l'image pour voir en action...).



Voici le code complet de cette page :

Code : HTML

```

<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>config01</title>
<link href="../bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="../bootstrap/css/bootstrap.responsive.min.css"
rel="stylesheet" type="text/css">
<style>
body {
background-color:#DDD;
color:#048
}
a {
color: #028;
}
p {
font-size: 12px;
line-height: 16px;
}
h1, h2, h3, h4, h5, h6 {
font-weight:lighter;
}
#contenu {
height: 300px;
}
#footer {
text-align:center;
background-color:#CCC;
border-radius: 10px;
padding: 4px;
}
#menu {
height: 100px;
background-color:#CCC;
border-radius: 10px;
}
</style>
</head>
<div class="container">
<body>
<div class="container">
<div class="row">
<div class="span12" id="bandeau">
<div class="page-header">
<h1>Mon beau site</h1>
</div>
</div>
</div>
<div class="row">
<div class="span2" id="menu">
<ul class="nav nav-list">
<li> <a href="#"> <i class="icon-repeat"></i> Accueil </a>
</li>
<li> <a href="#"> <i class="icon-book"></i> Livres </a>
</li>
<li> <a href="#"> <i class="icon-film"></i> Films </a> </li>
<li> <a href="#"> <i class="icon-music"></i> Musiques </a>
</li>
</ul>
</div>
<div class="span10" id="contenu">
<h2>Bienvenu sur mon site !</h2>
<br/>
<p>Il est bien entendu encore en construction...</p>
<br/>
<p>Bientôt plein d'informations délirantes ici...</p>
</div>

```

```
</div>
<div class="span12" id="footer">
  <p>Tous droits réservés par moi...</p>
</div>
</div>
</body>
</html>
```

La page est organisée en 4 parties :

- bannière supérieure
- menu à gauche
- contenu à droite
- pied de page en bas

Bootstrap permet de réaliser cette mise en page avec facilité.

J'ai profité de l'occasion pour montrer comment créer facilement un menu vertical élégant avec **Bootstrap**. Mais la partie qui va nous intéresser concerne les styles. Parce que tout une partie de la stylisation est en fait une modification des paramètres de base de **Bootstrap** :

Code : CSS

```
body {
  background-color:#DDD;
  color:#048
}
a {
  color: #028;
}
p {
  font-size: 12px;
  line-height: 16px;
}
h1, h2, h3, h4, h5, h6 {
  font-weight:lighter;
}
```

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**...

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**

```

34 // Scaffolding
35 // -----
36 @bodyBackground: #DDD;
37 @textColor: #048;
38
39
40 // Links
41 // -----
42 @linkColor: #028;
43 @linkColorHover: darken(@linkColor, 15%);
44
45
46 // Typography
47 // -----
48 @baseFontSize: 12px;
49 @baseFontFamily: "Helvetica Neue", Helvetica, Arial, sans-serif;
50 @baseLineHeight: 16px;
51 @altFontFamily: Georgia, "Times New Roman", Times, serif;
52

```

4. Enregistrer la nouvelle version du fichier **variables.Jess**
5. Ouvrir le fichier **bootstrap.less**
6. Compiler le fichier **bootstrap.less**



7. Enregistrer le fichier **bootstrap.css** généré par **Crunch**
8. Renommer le fichier **bootstrap.css** en **bootstrap.minxxx.css**, "xxx" représentant un indice pour distinguer cette version modifiée du Bootstrap original.
9. Modifier le lien des styles dans la page web :

Code : HTML

```

<link href="../../bootstrap/css/bootstrap.min4.css"
rel="stylesheet" type="text/css">

```

10. Supprimer les styles devenus inutiles dans la page web :

Code : CSS

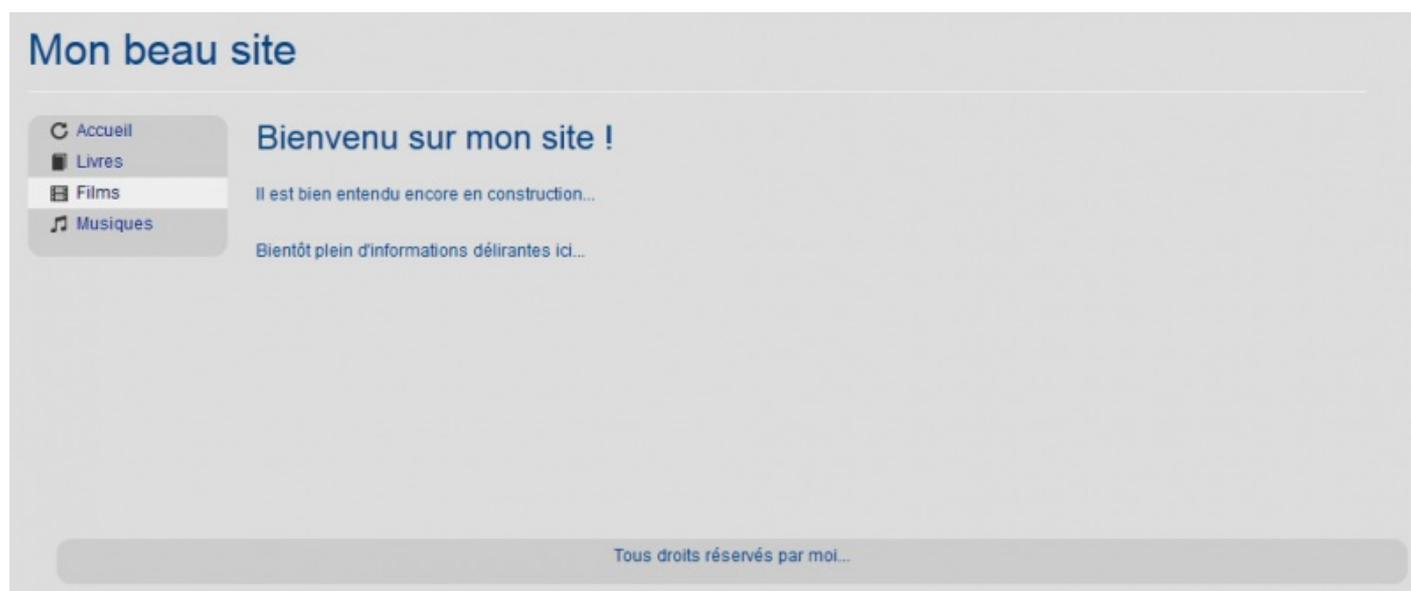
```

<style>
#contenu {
height: 300px;
}
#footer {
text-align:center;

```

```
background-color:#CCC;
border-radius: 10px;
padding: 4px;
}
#menu {
height: 100px;
background-color:#CCC;
border-radius: 10px;
}
</style>
```

Le résultat est exactement la même présentation de la page :



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 avec cette fois cette procédure. Voici la différence au niveau de l'en-tête :

Code : HTML

```
<link rel="stylesheet/less" type="text/css"
href="../bootstrap/less/bootstrap.less">
<script src="../js/less-1.3.0.min.js"></script>
```

Tester !

Évidemment ça 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 je change la couleur de fond dans le fichier **variables.less** :

```
// Scaffolding
// -----
@bodyBackground: #FFF;
@textColor: #048;
```

Il suffit de recharger la page pour voir le résultat immédiatement :

Mon beau site



Si on regarde le code généré par le navigateur on constate que le CSS a bien été construit :

```
4 <link rel="stylesheet/less" type="text/css" href="../bootstrap/less/bootstrap.less">
5 <script src="../js/less-1.3.0.min.js"></script><style id="less:bootstrap-less-bootstrap" media="screen" type="text/css">/*!
6 * Bootstrap v2.1.0
7 *
8 * Copyright 2012 Twitter, Inc
9 * Licensed under the Apache License v2.0
10 * http://www.apache.org/licenses/LICENSE-2.0
11 *
12 * Designed and built with all the love in the world @twitter by @mdo and @fat.
13 */
14 article,
15 aside,
16 details,
17 figcaption,
18 figure,
19 footer,
20 header,
21 hgroup,
22 nav,
23 section {
24   display: block;
25 }
26 audio,
27 canvas,
28 video {
29   display: inline-block;
30   *display: inline;
31   *zoom: 1;
32 }
33 audio:not([controls]) {
34   display: none;
35 }
36 html {
37   font-size: 100%;
38   -webkit-text-size-adjust: 100%;
39   -ms-text-size-adjust: 100%;
40 }
41 a:focus {
42   outline: thin dotted #333;
43   outline: 5px auto -webkit-focus-ring-color;
44   outline-offset: -2px;
45 }
```

Partie 4 : Bootstrap et son environnement

Dans cette dernière partie, nous allons voir comment optimiser son site pour la consultation sur différents supports et comment enrichir le framework avec des plugins.

Le responsive design

De nos jours on consulte de moins en moins le web sur un écran d'ordinateur. La prolifération de nouveaux supports (iPad, iPhone, iPod...) oblige les designers web à prendre en considération de nombreuses contraintes, la première étant la dimension différente de l'écran de tous ces supports. L'idéal est de créer des pages web qui s'adaptent automatiquement au support. On avait bien quelques possibilités d'adaptation avec **CSS2** et **HTML4** mais plutôt limités.

L'arrivée de **CSS3** change la donne en apportant beaucoup plus de souplesse dans la spécialisation du style avec les **Media Queries**.

Comme ce sont des spécifications nouvelles, je vais faire une petite introduction informative avant de montrer en quoi **Bootstrap** peut nous aider aussi dans ce domaine.

Présentation Présentation

Les **Media Queries** sont destinées à simplifier la création de pages web pour les rendre consultables sur des supports variés (tablette, portable...). Ce chapitre n'est qu'une introduction rapide à ce domaine qui mériterait à lui seul un tuto.

Avec les **Media Queries** on peut cibler ainsi :

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

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 dont on peut modifier la dimension de la fenêtre. Il existe d'ailleurs une balise **META viewport** consacrée à traiter ce point particulier. Si on ne s'en occupe pas, tous les efforts que l'on fait avec les **Media Queries** sont vains.

Il faut donc prendre quelques précautions :

Code : HTML

```
<meta name="viewport" content="user-scalable=no, initial-scale = 1,
minimum-scale = 1, maximum-scale = 1, width=device-width">
```

On bloque le zoom : **user-scalable=no**

On fixe le zoom minimum : **minimum-scale = 1**

On fixe le zoom maximum : **maximum-scale = 1**

On ouvre la fenêtre à la largeur de l'écran : **width=device-width**

(merci à [maxslayer44](#) pour les informations sur ce sujet)

Les trois derniers paramètres évoqués permettent de corriger des bugs d'affichage sous iOS5 lié à l'affichage retina (double résolution)

Il y a un site intéressant pour tester l'apparition sur mobile [ici](#). Ou même [une application beaucoup plus évoluée](#), mais qui ne fonctionne que sur **Chrome** et encore en bêta avec quelques blocages intempestifs.

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 à une largeur d'affichage inférieure ou égale à 767px.



Oui mais on 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 😊

Un exemple sans Bootstrap

Je vous propose un exemple d'utilisation sur une page web. Voici le code **HTML** :

Code : HTML

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Les Tigres</title>
<link href="css/test_style2.css" rel="stylesheet" type="text/css">
<!--[if lt IE 9]>
<script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
<body>
<div class="container">
  <header>  </header>
  <nav>
    <ul>
      <li><a href="#">Accueil</a></li>
      <li><a href="#">Liens</a></li>
      <li><a href="#">Témoignages</a></li>
      <li><a href="#">Références</a></li>
    </ul>
  </nav>
  <section>
    <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></p>
    <p>Voici ce qu'en dit le wikipedia :</p>
    <p> </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'hommes. 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.</blockquote>
  </section>
  <footer>
    <p>Web design par studio des délires enchantés...</p>
  </footer>
</div>
</body>
</html>
```



LES TIGRES

Accueil

Liens

Témoignages

Références

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

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.

Web design par studio des délires enchantés...

Et oui encore des tigres 😊 (toujours un petit clic sur l'image pour visualiser..)

La mise en page est sobre avec une largeur fixée à 960px, une bannière, un menu latéral, un contenu et un pied de page. Elle est assurée par une feuille de style qui gère à la fois le positionnement des éléments et leur présentation avec une bonne partie consacrée à l'apparition correcte du menu. Voici la partie qui concerne la mise en page :

Code : CSS

```
body {
  font: medium Verdana, Arial, Helvetica, sans-serif;
  margin: 5px;
  padding: 0;
  background: rgb(160,120,140);
}

h1, h2, h3, h4, h5, h6, p {
  margin-top: 0;
  padding-right: 15px;
  padding-left: 15px;
}

.container {
  width: 960px;
  margin: 0 auto;
}

header {
  height: 90px;
  border: 2px solid;
  -moz-border-radius: 8px;
}

nav {
  float: left;
  width: 180px;
  margin-top: 5px;
}

section {
  float: left;
  width: 780px;
  padding: 10px 0;
```

```

}

footer {
padding: 10px 0;
background: rgb(26,16,17);
clear: both;
color: #a9f;
text-align:center;
border-radius:8px;
-moz-border-radius:8px;
}

footer p {
margin-bottom: 0;
}

img {
width:100%;
height: 100%
}

ul, ol, dl {
padding: 0;
margin: 0;
}

...

```

Ce genre de page s'affiche tout à fait correctement sur un écran d'ordinateur, mais qu'en est-il sur une tablette ou un mobile ? Voici déjà une réponse :



Oui ça va mais si je passe en mode portrait ?

Aie là ça se gâte :



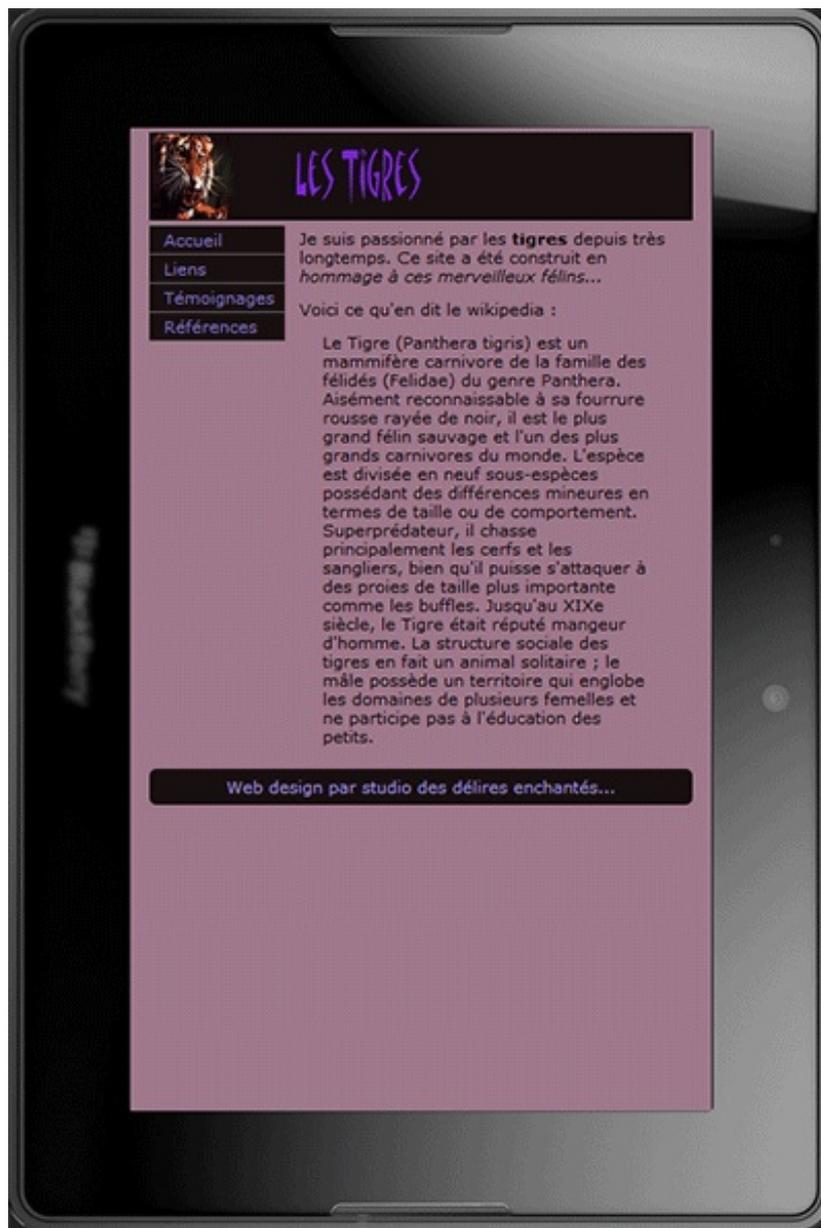
Ce n'est même pas la peine d'essayer sur un mobile 🤖

On va maintenant utiliser les **Media Queries** pour arranger un peu la situation :

Code : CSS

```
@media screen and (max-width: 980px) {  
  .container {  
    width: 95%;  
  }  
  
  section {  
    width: 75%;  
  }  
  
  nav {  
    width: 25%;  
  }  
  
  nav ul a, nav ul a:visited {  
    width: auto;  
  }  
}
```

On abandonne les valeurs fixes pour des pourcentages et on s'arrange pour ajuster le menu :



D'accord là c'est bon, mais sur un smartphone ?

On va essayer :



Bon là on a encore un problème 🤔

On va un peu améliorer notre approche en prévoyant deux cas :

Code : CSS

```
@media screen and (max-width: 980px) and (min-width: 768px) {  
  
  .container {  
    width: 95%;  
  }  
  
  section {  
    width: 75%;  
  }  
  
  nav {  
    width: 25%;  
  }  
  
  nav ul a, nav ul a:visited {  
    width: auto;  
  }  
}
```

```
    }  
  }  
  
  @media screen and (max-width: 768px) {  
  
    body {  
      font-size: large;  
    }  
  
    .container {  
      width: 95%;  
    }  
  
    nav {  
      width: 100%;  
      float: none;  
    }  
  
    nav ul a, nav ul a:visited {  
      width: auto;  
    }  
  
    section {  
      width: 95%;  
    }  
  }  
}
```

Pour une largeur inférieure à 768px on remet dans le flux et on rend les caractères plus gros, voilà le résultat :



Cette fois c'est beaucoup mieux 😊

Un exemple avec Bootstrap

On va faire presque la même page en utilisant **Bootstrap** :

Code : HTML

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Les Tigres</title>
<link href="../bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="css/test_style5.css" rel="stylesheet" type="text/css">
<!--[if lt IE 9]>
<script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
<body>
<div class="container">
  <header class="row">  </header>
  <div class="row">
<div class="span2">
```

```

<nav class="span2" >
  <ul class="nav nav-list">
    <li><a href="#">Accueil</a></li>
    <li><a href="#">Liens</a></li>
    <li><a href="#">Témoignages</a></li>
    <li><a href="#">Références</a></li>
  </ul>
</nav>
<section class="span10">
  <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></p>
  <p>Voici ce qu'en dit le wikipedia :</p>
  <p></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'hommes. 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.</blockquote>
</section>
</div>
<footer class="row">
  <p>Web design par studio des délires enchantés...</p>
</footer>
</div>
</body>
</html>

```

Avec ce code CSS qui est très léger par rapport au précédent sans utilisation de **Bootstrap** :

Code : CSS

```

body {
  margin: 5px;
  background: rgb(160,120,140);
}

header {
  border:2px solid;
  -moz-border-radius:8px;
  margin-bottom:20px;
}

footer {
  background: rgb(26,16,17);
  color: #a9f;
  text-align:center;
  border-radius:8px;
  -moz-border-radius:8px;
}

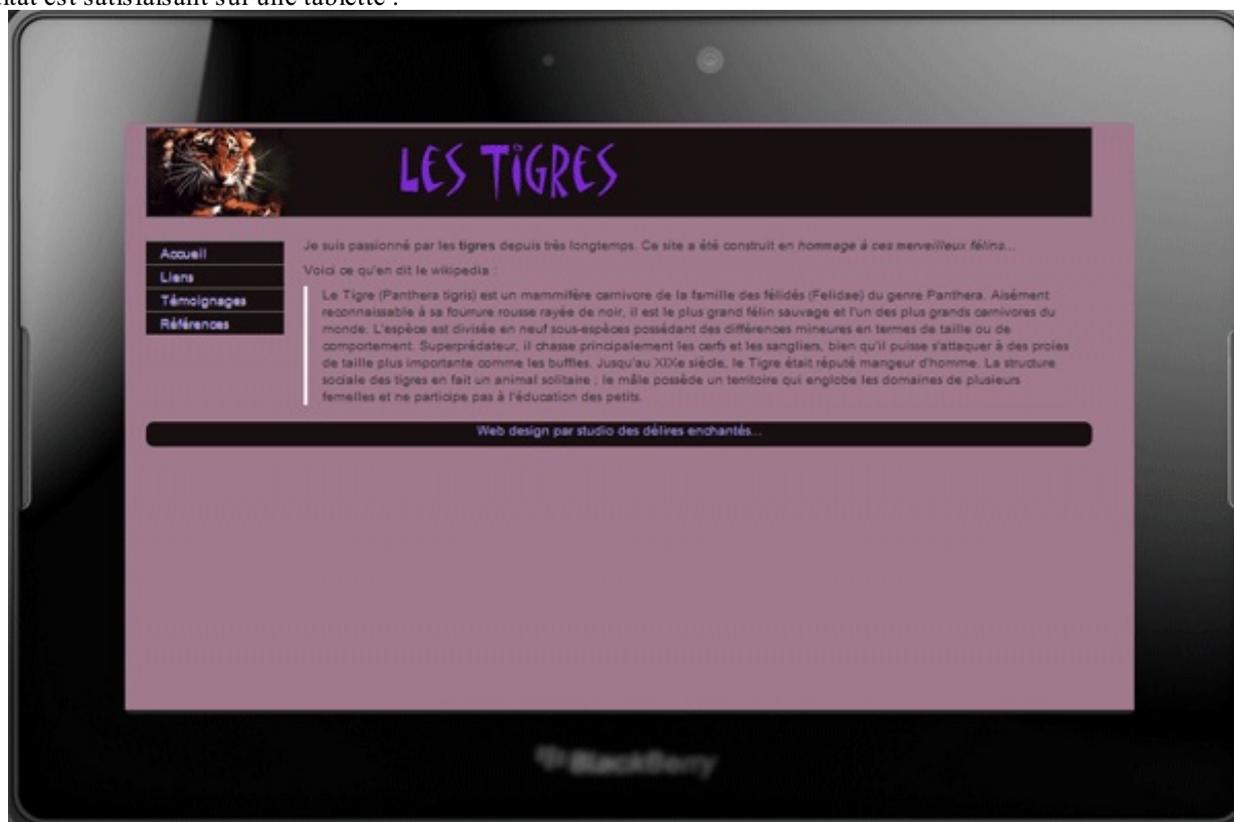
img {
  width:100%;
}

.span2 {
  margin: 5px 0 0 0;
  background: rgb(26,16,17);
}

```

```
}  
  
nav a {  
  color: #a9f;  
  border-top: 1px solid #666;  
}
```

Le résultat est satisfaisant sur une tablette :



Mais il montre aussi vite ses limites dès qu'on veut passer en portrait :



Alors on va mettre en œuvre le fichier de **Bootstrap** dévolu aux **Media Queries** :

Code : HTML

```
<link href="../../bootstrap/css/bootstrap-responsive.min.css"
rel="stylesheet" type="text/css">
```

Voilà maintenant le résultat :



Ça passe même sur un écran plus réduit :



Mais les caractères sont un peu petits pour une lecture aisée, il faudrait arranger un peu ce point. Mais **Bootstrap** gère déjà de façon très efficace la variété des affichages sans beaucoup d'effort 😊

Des classes pour masquer ou dévoiler des éléments

Sur certaines page web, il y a parfois des éléments qui trouvent leur place sur un grand écran, mais deviennent superflus ou même impossibles à caser sur des écrans réduits. Il est alors utile de pouvoir faire disparaître des éléments. **Bootstrap** est équipé de quelques classes bien pratiques pour effectuer ces apparitions/disparitions. Voici le tableau extrait de la documentation :

Class	Phones 480px and below	Tablets 767px and below	Desktops 768px and above
<code>.visible-phone</code>	Visible	Hidden	Hidden
<code>.visible-tablet</code>	Hidden	Visible	Hidden
<code>.visible-desktop</code>	Hidden	Hidden	Visible
<code>.hidden-phone</code>	Hidden	Visible	Visible
<code>.hidden-tablet</code>	Visible	Hidden	Visible
<code>.hidden-desktop</code>	Visible	Visible	Hidden

Voici une page de test :

Code : HTML

```
<!DOCTYPE HTML>
<html>
```

```

<head>
<meta name="viewport" content="user-scalable=no, initial-scale = 1,
minimum-scale = 1, maximum-scale = 1, width=device-width">
<meta charset="utf-8">
<title>Test Responsive</title>
<link href="../../bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<link href="../../bootstrap/css/bootstrap-responsive.min.css"
rel="stylesheet" type="text/css">
<!--[if lt IE 9]>
<script
src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
<style></style>
</head>
<body>
<div class="container">
  <div class="row well-small">
    <div class="span1">Visible : </div>
    <div class="span5 btn-group">
      <button class="btn btn-success visible-
phone">Téléphone</button>
      <button class="btn btn-success visible-
tablet">Tablette</button>
      <button class="btn btn-success visible-
desktop">Ordinateur</button>
    </div>
  </div>
  <div class="row well-small">
    <div class="span1">Invisible : </div>
    <div class="span5 btn-group">
      <button class="btn btn-danger hidden-phone">Téléphone</button>
      <button class="btn btn-danger hidden-tablet">Tablette</button>
      <button class="btn btn-danger hidden-
desktop">Ordinateur</button>
    </div>
  </div>
</div>
</body>
</html>

```

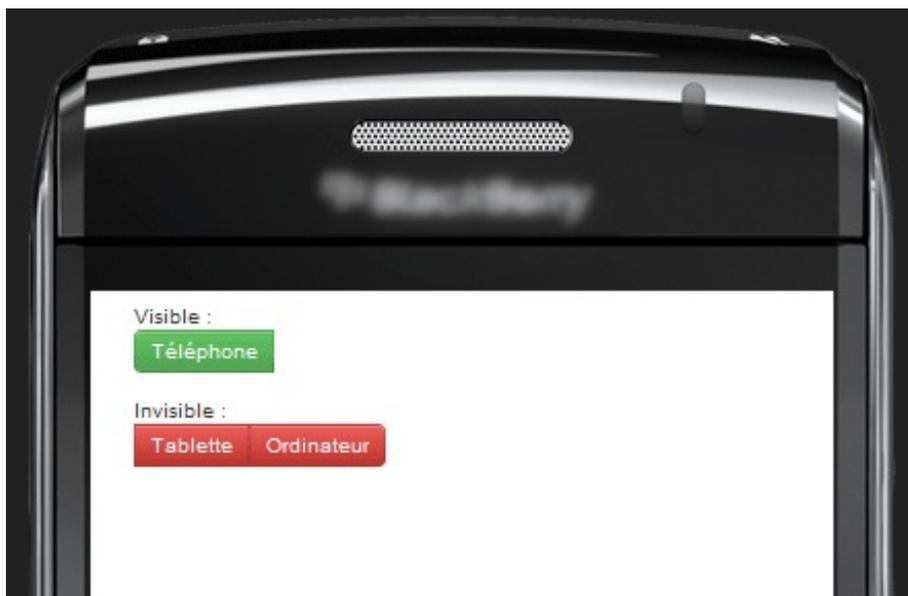
Et le résultat sur ordinateur :



Sur tablette :



Sur téléphone :



Ça a l'air de marcher 😊

📁 Plugins et extensions

Dans ce dernier chapitre je me propose de recenser les ressources disponibles pour Bootstrap : thèmes, plugins, extensions et autres...

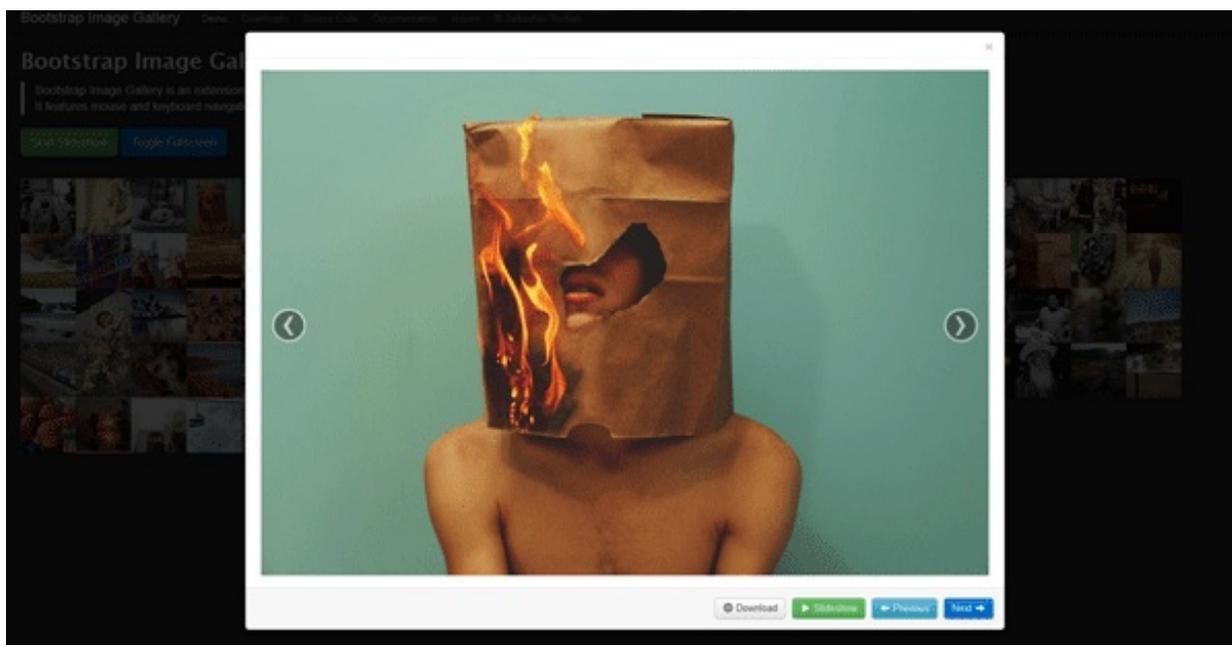
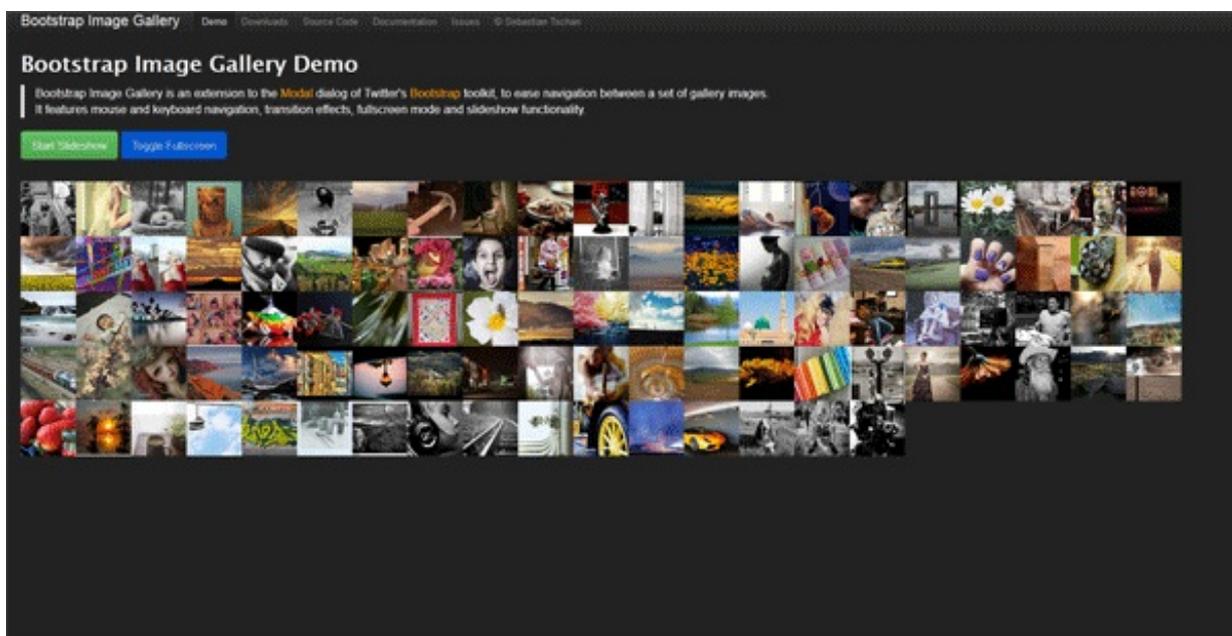
Plugins

Bootstrap Image Gallery

C'est une extension du plugin **Modals** destinée à créer de beaux effets de navigation avec des images. Au programme :

- Slideshow
- Belles transitions
- Navigation à la souris
- Navigation au clavier
- Passage en mode "plein écran"

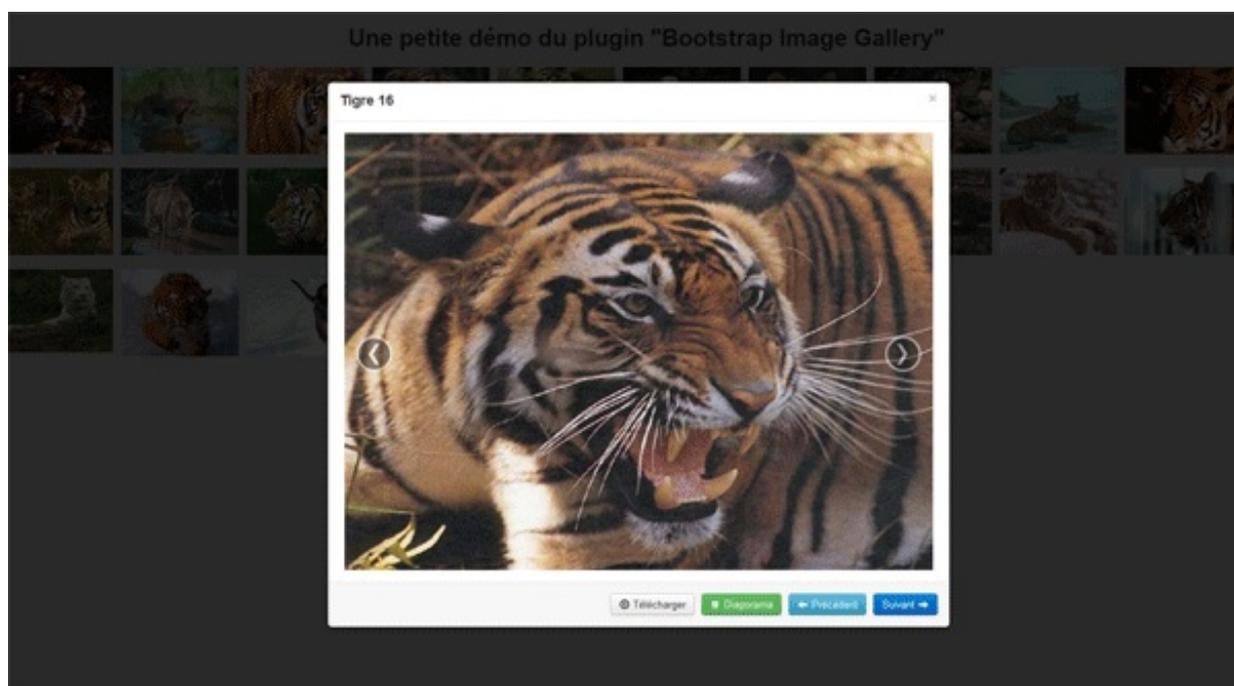
Allez vite voir la démo 😊



En fait c'est un peu un mélange entre la fenêtre modale et le carrousel.

La mise en oeuvre n'est pas bien difficile, d'autant que la [documentation](#) est bien faite et respecte les principes de celle de **Bootstrap**. Il faut aller chercher [les fichiers](#) sur le site.

Je vous propose un exemple, toujours avec les tigres (toujours un clic sur l'image) 😊



Voici le code de cette page :

Code : HTML

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<title>Les Tigres</title>
<link href="../bootstrap/css/bootstrap.min.css" rel="stylesheet"
type="text/css">
<!--[if lt IE 7]><link rel="stylesheet"
href="http://blueimp.github.com/cdn/css/bootstrap-
ie6.min.css"><![endif]-->
<link rel="stylesheet" href="../bootstrap/css/bootstrap-image-
gallery.min.css">
<style>
body {
background: rgb(204,204,204);
```

```

}
h1 {
  margin:20px 0 20px 20px;
  text-align:center;
}
</style>
</head>
<body>
<header>
  <h1>Une petite démo du plugin "Bootstrap Image Gallery"</h1>
</header>
<div class="container-fluid">
  <div id="modal-gallery" class="modal modal-gallery hide fade">
    <div class="modal-header"> <a class="close" data-
dismiss="modal">&times;</a>
    <h3 class="modal-title"></h3>
    </div>
    <div class="modal-body">
      <div class="modal-image"></div>
    </div>
    <div class="modal-footer"> <a class="btn modal-download"
target="_blank"><i class="icon-download"></i> Télécharger</a>
    <a class="btn btn-success modal-play modal-slideshow" data-
slideshow="5000"><i class="icon-play icon-white"></i> Diaporama</a>
    <a class="btn btn-info modal-prev"><i class="icon-arrow-left
icon-white"></i> Précédent</a>
    <a class="btn btn-primary modal-next">Suivant <i class="icon-
arrow-right icon-white"></i></a> </div>
    </div>
    <div id="gallery" data-target="#modal-gallery" data-toggle="modal-
gallery">
      <ul class="thumbnails">
        <li class="span2"> <a rel="gallery" href="images/tg1.jpg"
title="Tigre 1"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg2.jpg"
title="Tigre 2"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg3.jpg"
title="Tigre 3"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg4.jpg"
title="Tigre 4"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg5.jpg"
title="Tigre 5"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg6.jpg"
title="Tigre 6"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg7.jpg"
title="Tigre 7"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg8.jpg"
title="Tigre 8"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg9.jpg"
title="Tigre 9"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg10.jpg"
title="Tigre 10"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg11.jpg"
title="Tigre 11"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg12.jpg"
title="Tigre 12"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg13.jpg"
title="Tigre 13"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg14.jpg"
title="Tigre 14"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg15.jpg"
title="Tigre 15"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg16.jpg"
title="Tigre 16"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg17.jpg"
title="Tigre 17"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg18.jpg"
title="Tigre 18"></a> </li>
        <li class="span2"> <a rel="gallery" href="images/tg19.jpg"
title="Tigre 19"></a> </li>
      </ul>
    </div>
  </div>
</div>

```

```

    <li class="span2"> <a rel="gallery" href="images/tg20.jpg"
title="Tigre 20"></a> </li>
    <li class="span2"> <a rel="gallery" href="images/tg21.jpg"
title="Tigre 21"></a> </li>
    <li class="span2"> <a rel="gallery" href="images/tg22.jpg"
title="Tigre 22"></a> </li>
    <li class="span2"> <a rel="gallery" href="images/tg23.jpg"
title="Tigre 23"></a> </li>
    <li class="span2"> <a rel="gallery" href="images/tg24.jpg"
title="Tigre 24"></a> </li>
    <li class="span2"> <a rel="gallery" href="images/tg25.jpg"
title="Tigre 25"></a> </li>
    <li class="span2"> <a rel="gallery" href="images/tg26.jpg"
title="Tigre 26"></a> </li>
  </ul>
</div>
<br>
</div>
</div>
<script src="../../js/jquery.js"></script>
<script src="../../bootstrap/js/bootstrap.min.js"></script>
<script src="http://blueimp.github.com/JavaScript-Load-Image/load-
image.min.js"></script>
<script src="../../bootstrap/js/bootstrap-image-
gallery.min.js"></script>
</body>
</html>

```

Il faut déclarer la feuille de style du plugin :

Code : HTML

```

<link rel="stylesheet" href="../../bootstrap/css/bootstrap-image-
gallery.min.css">

```

Ensuite la mise en œuvre de la fenêtre modale suit exactement les principes du plugin **Modals** :

Code : HTML

```

<div id="modal-gallery" class="modal modal-gallery hide fade">
  <div class="modal-header"> <a class="close" data-
dismiss="modal">&times;</a>
  <h3 class="modal-title"></h3>
</div>
<div class="modal-body">
  <div class="modal-image"></div>
</div>
<div class="modal-footer"> <a class="btn modal-download"
target="_blank"><i class="icon-download"></i> Télécharger</a>
  <a class="btn btn-success modal-play modal-slideshow" data-
slideshow="5000"><i class="icon-play icon-white"></i> Diaporama</a>
  <a class="btn btn-info modal-prev"><i class="icon-arrow-left
icon-white"></i> Précédent</a>
  <a class="btn btn-primary modal-next">Suivant <i class="icon-
arrow-right icon-white"></i></a> </div>
</div>

```

On place les images dans des balises `<a>` avec la référence `rel="gallery"` et le lien des images (une miniature pour le lien et l'image normale en cible):

Code : HTML

```
<a rel="gallery" href="images/tgl.jpg" title="Tigre 1"></a>
```

Il faut aussi faire appel, en plus de **jQuery** et **Bootstrap** à deux libraires **Javascript** :

Code : HTML

```
<script src="http://blueimp.github.com/JavaScript-Load-Image/load-
image.min.js"></script>
<script src="../../bootstrap/js/bootstrap-image-
gallery.min.js"></script>
```

J'ai utilisé la classe **thumbnails** pour bien répartir les images.

Colorpicker and Datepicker for Twitter Bootstrap

Deux plugins intéressants qu'on peut trouver ici :

Stefan Petre

Home Contact Iubesc Brasovul Cautare

Colorpicker and Datepicker for Twitter Bootstrap

februarie 16, 2012 | Comentarii 39 | Share it



Colorpicker

Add color picker to field or to any other element.

- can be used as a component
- alpha picker
- multiple formats: hex, rgb, rgba, hsl, hsla

Download & docs



Datepicker

Add datepicker picker to field or to any other element.

- can be used as a component
- formats: dd, d, mm, m, yyyy, yy
- separators: -, /, ..

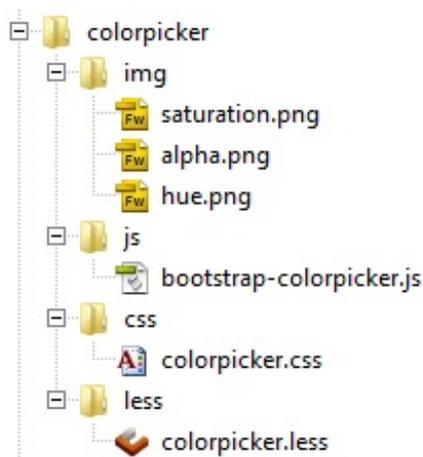
Download & docs

Colorpicker

Le Colorpicker est un plugin qui permet une saisie facilitée d'un code de couleur.

Vous trouver [la documentation](#) sur le site ainsi qu'un [bouton de téléchargement](#) des fichiers.

Voici les fichiers composant ce plugin :



Les images et les fichiers **CSS** et **JavaScript** doivent être positionnés avec ceux de **Bootstrap**. Le fichier **LESS** est optionnel pour ceux qui voudraient modifier l'aspect de l'affichage.

Il faut déclarer la feuille **CSS** spécifique du plugin en plus de celle de **Bootstrap** :

Code : HTML

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

Il faut déclarer le fichier **JavaScript** spécifique du plugin en plus de celui de **jQuery** :

Code : HTML

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

L'utilisation du plugin est facile, en voici un exemple :

Code : HTML

```
<form class="form-inline well">
  <label for="color">Couleur : </label>
  <input id="color" type="text" class="input-mini">
</form>
```

Et l'appel de **jQuery** correspondant pour initialiser le composant :

Code : JavaScript

```
$(function () {
  $('input').colorpicker()
});
```

Résultat en image (et en réel avec un petit clic...) :



On peut changer le format de définition de la couleur et déclarer une valeur initiale facilement :

Code : HTML

```
<form class="form-inline well">
  <label for="color">Couleur : </label>
  <input id="color" type="text" class="input-medium"
  value="rgb(200,10,50)" data-color-format="rgb">
</form>
```

Tester !

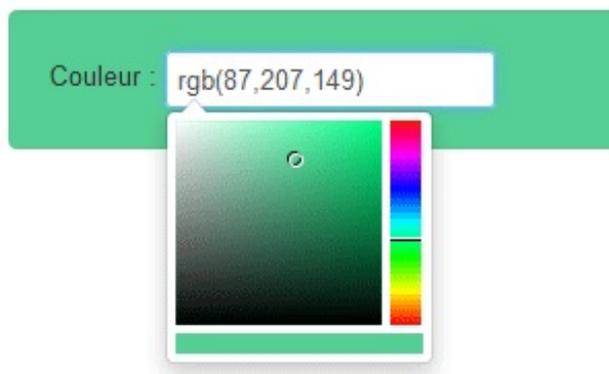
On peut aussi utiliser le format avec valeur de transparence apparue avec le CSS 3 :

Code : HTML

```
<form class="form-inline well">
  <label for="color">Couleur : </label>
  <input id="color" type="text" class="input-medium" data-color-
  format="rgba">
</form>
```



On peut aussi utiliser l'événement de changement de la valeur de la couleur, comme dans cet exemple où j'applique la couleur au fond du formulaire :



Avec ce code :

Code : JavaScript

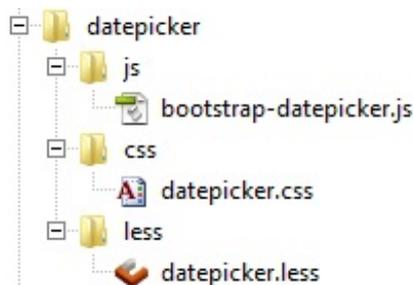
```
$('#input').colorpicker().on('changeColor', function(e) {
  $('#form').css("background-color", e.color.toHex());
});
```

Datepicker

Le **Datepicker** est un plugin qui permet une saisie facilitée d'une date.

Vous trouver [la documentation](#) sur le site ainsi qu'un [bouton de téléchargement](#) des fichiers.

Voici les fichiers composant ce plugin :



Les images et les fichiers **CSS** et **JavaScript** doivent être positionnés avec ceux de **Bootstrap**. Le fichier **LESS** est optionnel pour ceux qui voudraient modifier l'aspect de l'affichage.

Il faut déclarer la feuille **CSS** spécifique du plugin en plus de celle de **Bootstrap** :

Code : HTML

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

Il faut déclarer le fichier **JavaScript** spécifique du plugin en plus de celui de **jQuery** :

Code : HTML

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

L'utilisation du plugin est facile, en voici un exemple :

Code : HTML

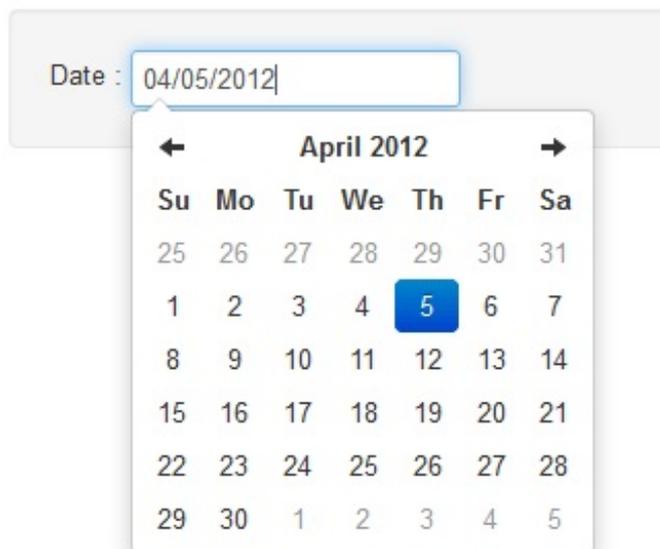
```
<form class="form-inline well">
  <label for="date">Date : </label>
  <input id="date" type="text" class="input-medium">
</form>
```

Et l'appel de **jQuery** correspondant pour initialiser le composant :

Code : JavaScript

```
$(function () {
  $('input').datepicker();
});
```

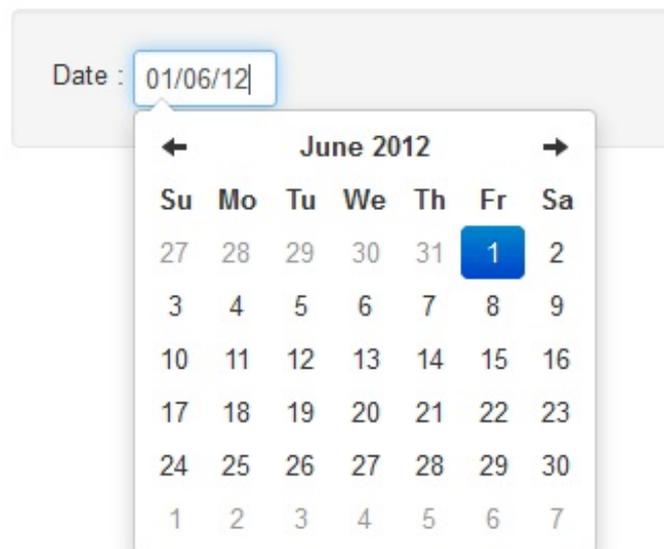
Résultat en image (et en réel avec un petit clic...) :



On peut changer le format de définition de la date et déclarer une valeur initiale facilement :

Code : HTML

```
<form class="form-inline well">
  <label for="date">Date : </label>
  <input id="date" type="text" class="input-mini" data-date-
  format="dd/mm/yy" value="01/06/12">
</form>
```



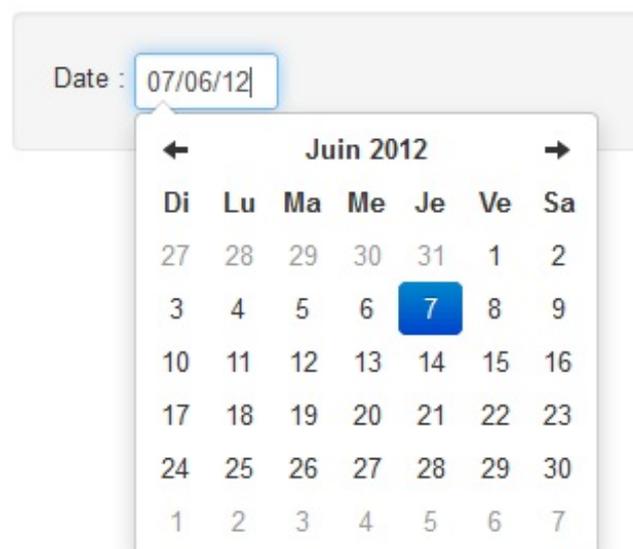
On peut aussi facilement franciser les appellations en intervenant au niveau du fichier `bootstrap-datepicker.js`:

Code : JavaScript

```

dates: {
  days: ["Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi",
    "Vendredi", "Samedi", "Dimanche"],
  daysShort: ["Dim", "Lun", "Mar", "Mer", "Jeu", "Ven", "Sam",
    "Dim"],
  daysMin: ["Di", "Lu", "Ma", "Me", "Je", "Ve", "Sa", "Di"],
  months: ["Janvier", "Fevrier", "Mars", "Avril", "Mai", "Juin",
    "Juillet", "Août", "Septembre", "Octobre", "Novembre", "Decembre"],
  monthsShort: ["Jan", "Fev", "Mar", "Avr", "Mai", "Jui", "Jui", "Aoû",
    "Sep", "Oct", "Nov", "Dec"]
},

```



Il est possible d'utiliser l'événement de changement de date :

Code : JavaScript

```

$('input').datepicker().on('changeDate', function(e) {
  $('label').text("Date changée !");
});

```

Date (changée !)

← Juin 2012 →

Di	Lu	Ma	Me	Je	Ve	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Un petit TP pour ce contrôle : on veut deux zones de saisie, une pour une date de départ et une autre pour la date de retour, avec vérification d'antériorité de la date de départ. Il y a un exemple sur le site du créateur, on peut le simplifier, je vous laisse réfléchir...

Voilà déjà l'effet souhaité :

La date de retour doit être postérieure à la date de départ !

Date de départ :

Date de retour :

← Avril 2012 →

Di	Lu	Ma	Me	Je	Ve	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Et une solution possible :

Secret (cliquez pour afficher)

Code : HTML

```
<div class="container"> <br>
  <form class="form-horizontal well">
    <div id="alerte" class="alert alert-error"
```

```
style="display:none"> <strong>La date de retour doit être
postérieure à la date de départ !</strong> </div>
  <fieldset>
    <div class="control-group">
      <label for="start" class="control-label">Date de départ :
</label>
      <div class="controls">
        <input id="start" type="text" class="input-mini" data-
date-format="dd/mm/yy">
        </div>
      </div>
      <div class="control-group">
        <label for="end" class="control-label">Date de retour :
</label>
        <div class="controls">
          <input id="end" type="text" class="input-mini" data-
date-format="dd/mm/yy">
          </div>
        </div>
      </div>
    </fieldset>

  </form>
</div>
</body>
<script src="../../js/jquery.js"></script>
<script src="../../bootstrap/js/bootstrap-datepicker_fr.js"></script>
<script>
$(function () {
  function get_date(text) {
    var d_split = text.split('/');
    var d = new Date(0);
    return d.setFullYear(d_split[2], d_split[1], d_split[0])
  }
  $('#start, #end').datepicker().on('changeDate', function(e) {
    if(get_date($('#start').val()) >= get_date($('#end').val()))
      $('#alerte').show();
    else $('#alerte').hide();
  });
});
</script>
```

Extensions

De jolis boutons

Génération du CSS

Un [site](#) propose de générer du code CSS pour améliorer l'aspect des boutons de **Bootstrap**. La mise en œuvre est facile. D'abord vous ajuster les paramètres avec des curseurs :



Ensuite vous récupérez le code **CSS** :

How to Use This CSS Generator

This is an extension to the Twitter Bootstrap framework. **It makes creating pretty buttons easy.** (Send improvements to @charliepark.)

First, monkey with the sliders on the left.

Use your arrow keys for extra precision. (Button Puffiness might not affect all browsers.)

Second, copy the CSS in the box below.

You should be able to just pop it into your CSS file. Apply the class "btn-custom" to any button (or other element) on your site that you want to have these colors.

```

background-image: -khtml-gradient(linear, left top, left bottom, from(#45c62e), to(#051003))
background-image: -moz-linear-gradient(top, #45c62e, #051003);
background-image: -ms-linear-gradient(top, #45c62e, #051003);
background-image: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #45c62e), co
background-image: -webkit-linear-gradient(top, #45c62e, #051003);
background-image: -o-linear-gradient(top, #45c62e, #051003);
background-image: linear-gradient(#45c62e, #051003);
border-color: #051003 #051003 hsl(111, 62%, -7%);
color: #fff !important;
text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.72);
-webkit-font-smoothing: antialiased;
}

```

Le code généré crée la classe **btn-custom** :

Code : CSS

```
.btn-custom {
  background-color: hsl(111, 62%, 4%) !important;
  background-repeat: repeat-x;
  filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr="#45c62e",
endColorstr="#051003");
  background-image: -khtml-gradient(linear, left top, left bottom,
from(#45c62e), to(#051003));
  background-image: -moz-linear-gradient(top, #45c62e, #051003);
  background-image: -ms-linear-gradient(top, #45c62e, #051003);
  background-image: -webkit-gradient(linear, left top, left bottom,
color-stop(0%, #45c62e), color-stop(100%, #051003));
  background-image: -webkit-linear-gradient(top, #45c62e, #051003);
  background-image: -o-linear-gradient(top, #45c62e, #051003);
  background-image: linear-gradient(#45c62e, #051003);
  border-color: #051003 #051003 hsl(111, 62%, -7%);
  color: #fff !important;
  text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.72);
  -webkit-font-smoothing: antialiased;
}
```

Il suffit d'utiliser la classe :

Code : HTML

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

Pour avoir un joli bouton :



Génération du HTML

Un site va encore plus loin en générant le code **HTML** spécialement pour **Bootstrap** :

Twitter Bootstrap Button Generator

Button text:

Button Size: Mini Small Large

Button Color: Default Primary Info Success Warning Danger Inverse

Button Type: Link Button Input Submit

Button State: None Active Disabled

Icon:

Icon Position: ← Left Right →

Icon Color: White Black

Choose Icon:

Twitter Bootstrap Button Preview

↖ [Click here to copy HTML](#)

```
<a href="#" class="btn btn-success btn-large">
<i class="icon-white icon-heart"></i> Bootstra
p Button Generator</a>
```

PlugoBrowser - TinyMCE Image & File Manager / Uploader

Vous avez juste à cliquer sur les option et un petit clic vous donne le code **HTML** directement dans le presse-papier 😊.

Des icônes vectorielles

Présentation

Le site [Font Awesome](#) propose une collection d'icônes spécialement conçues pour **Bootstrap**. La plupart sont des reproductions de celles déjà contenues dans **Bootstrap** :

Base Icons

Inspired by the Glyphicon set that comes with [Twitter Bootstrap 2.0](#), Font Awesome is designed from scratch for a full array of web-related actions.

icon-glass	icon-refresh	icon-pencil	icon-arrow-left
icon-music	icon-list-alt	icon-map-marker	icon-arrow-right
icon-search	icon-lock	icon-adjust	icon-arrow-up
icon-envelope	icon-flag	icon-tint	icon-arrow-down
icon-heart	icon-headphones	icon-edit	icon-share-alt
icon-star	icon-volume-off	icon-share	icon-resize-full
icon-star-empty	icon-volume-down	icon-check	icon-resize-small
icon-user	icon-volume-up	icon-move	icon-plus
icon-film	icon-qrcode	icon-step-backward	icon-minus
icon-th-large	icon-barcode	icon-fast-backward	icon-asterisk
icon-th	icon-tag	icon-backward	icon-exclamation-sign
icon-th-list	icon-tags	icon-play	icon-gift
icon-ok	icon-book	icon-pause	icon-leaf
icon-remove	icon-bookmark	icon-stop	icon-fire
icon-zoom-in	icon-print	icon-forward	icon-eye-open
icon-zoom-out	icon-camera	icon-fast-forward	icon-eye-close
icon-off	icon-font	icon-step-forward	icon-warning-sign
icon-signal	icon-bold	icon-eject	icon-plane
icon-cog	icon-italic	icon-chevron-left	icon-calendar
icon-trash	icon-text-height	icon-chevron-right	icon-random
icon-home	icon-text-width	icon-plus-sign	icon-comment
icon-file	icon-align-left	icon-minus-sign	icon-magnet
icon-time	icon-align-center	icon-remove-sign	icon-chevron-up
icon-road	icon-align-right	icon-ok-sign	icon-chevron-down
icon-download-alt	icon-align-justify	icon-question-sign	icon-retweet
icon-download	icon-list	icon-info-sign	icon-shopping-cart
icon-upload	icon-indent-left	icon-screenshot	icon-folder-close
icon-inbox	icon-indent-right	icon-remove-circle	icon-folder-open
icon-play-circle	icon-facetime-video	icon-ok-circle	icon-resize-vertical
icon-repeat	icon-picture	icon-ban-circle	icon-resize-horizontal

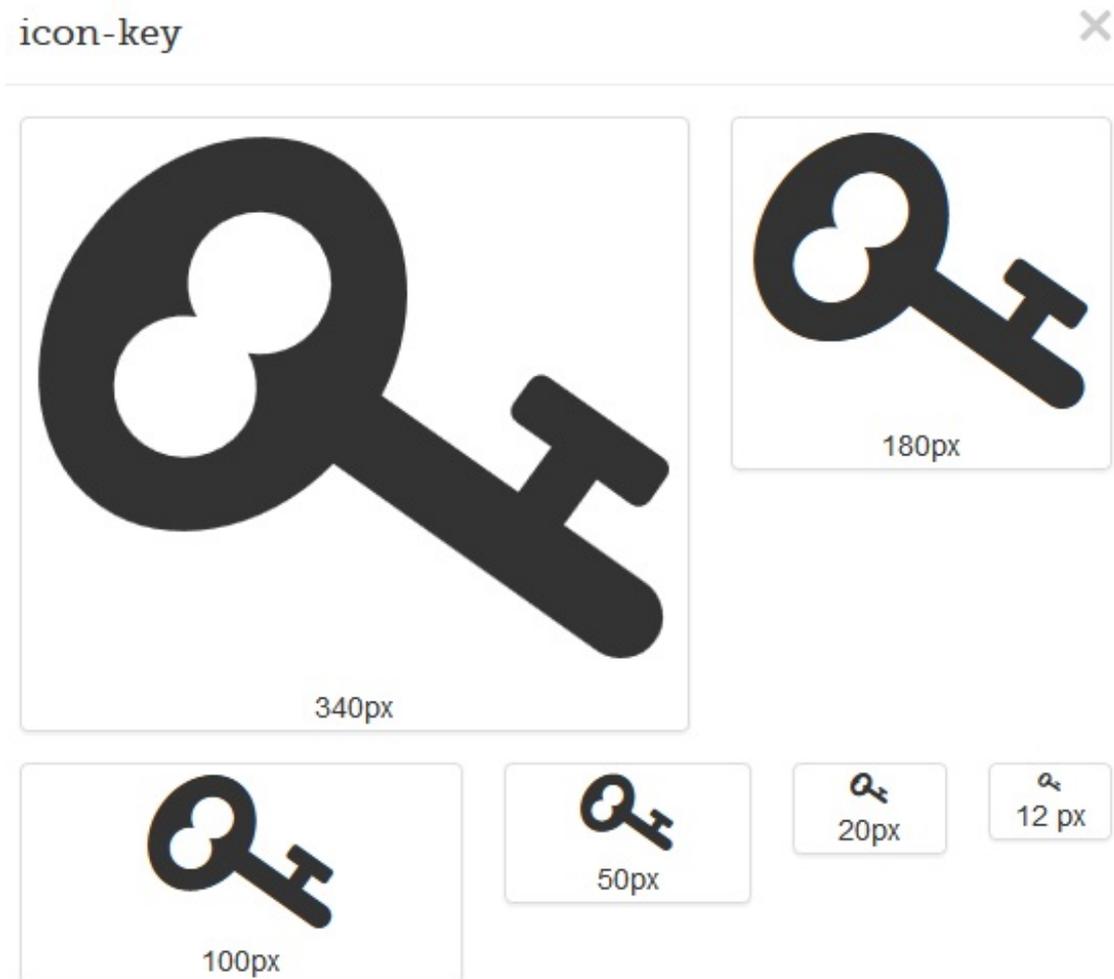
Mais il y en a des nouvelles :

Extended Icons

But wait, there's more! Font Awesome also includes this set of ever-expanding extra-awesome pictograms.

icon-camera-retro	icon-facebook-sign	icon-thumbs-up	icon-signin
icon-bar-chart	icon-twitter-sign	icon-thumbs-down	icon-signout
icon-cogs	icon-linkedin-sign	icon-comments	icon-star-half
icon-external-link	icon-github-sign	icon-trophy	icon-heart-empty
icon-pushpin	icon-key	icon-upload-alt	icon-lemon

Le gros avantage de cette collection est que les images sont vectorielles, on peut donc les agrandir sans perte de qualité :

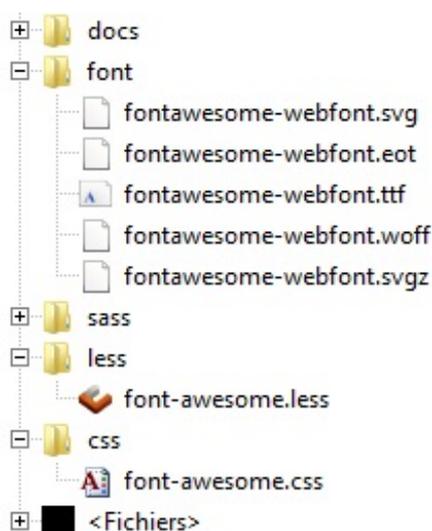


Installation

Pour mettre en œuvre ces icônes c'est tout simple. D'abord on télécharge les fichiers en cliquant sur le bouton :



Vous obtenez un certain nombre de fichiers répartis dans des répertoires. J'ai fait apparaître ci-dessous les plus importants :



Les polices vectorielles sont contenues dans le répertoire **font**.

L'appel de ces polices se fait dans la feuille de style **font-awesome.css** :

Code : CSS

```
@font-face {
  font-family: "FontAwesome";
  font-style: normal;
  font-weight: normal;
  src: url("../font/fontawesome-webfont.eot?#iefix")
  format("embedded-opentype"),
  url("../font/fontawesome-webfont.woff") format("woff"),
  url("../font/fontawesome-webfont.ttf") format("truetype"),
  url("../font/fontawesome-webfont.svgz#FontAwesomeRegular")
  format("svg"),
  url("../font/fontawesome-webfont.svg#FontAwesomeRegular")
  format("svg");
}
```

La façon la plus simple d'utiliser ces icônes est donc de placer le répertoire **font** avec ses fichiers sur son site ainsi que la feuille de style **font-awesome.css** en faisant attention à bien pointer le répertoire des polices au niveau des url.

Il est aussi possible de recompiler **Bootstrap** avec **LESS** puisqu'il est prévu un fichier **font-awesome.less** à importer.

On peut mettre des icônes partout, il y a plusieurs exemples bien conçus sur le site.

Kickstrap

Kickstrap est un projet ambitieux fondé sur **Bootstrap**. Il propose des thèmes et des plugins supplémentaires. Au moment où j'écris ce cours il n'y a pas encore de version stable.

Kickstrap

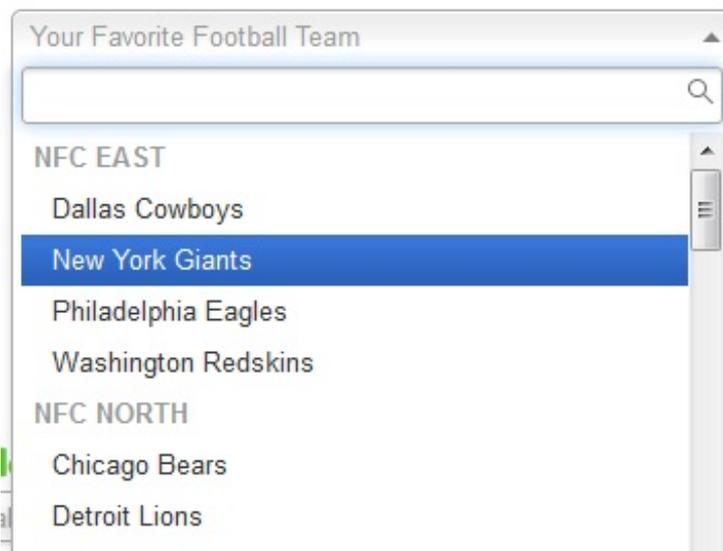
A full version of Twitter's Bootstrap with themes, enhancements, and other goodies.

Les thèmes proposés sont ceux de [bootswatch](#), j'en ai déjà parlé dans le chapitre sur la configuration.

Kickstrap intègre aussi [HTML5 Boilerplate](#), un template **HTML/CSS/JS** assez complet.

Au niveau des composants ajoutés on trouve :

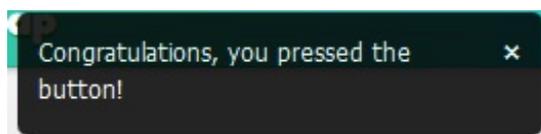
- [chosen](#), des listes comportant de nombreuses possibilités de paramétrages :



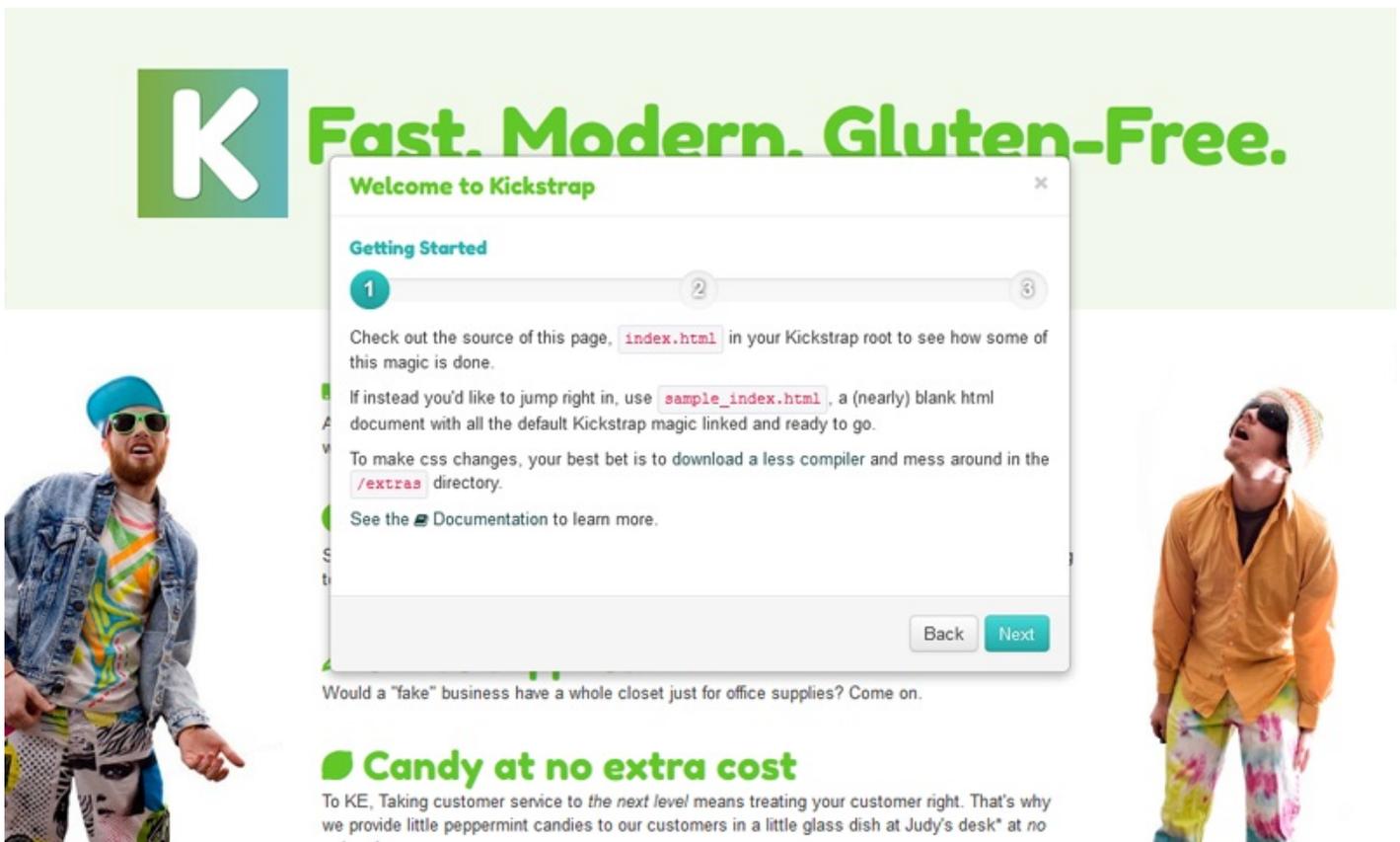
- [animate.css](#), un fichier CSS qui permet de faire de belles animations :



- [jGrowl](#), un plugin **jQuery** qui permet de faire de jolies fenêtres de notification :



Mais le plus grand intérêt de **Kickstrap** est sans doute qu'il se présente comme une solution intégrée avec un guide de démarrage plutôt bien fait :



K Fast. Modern. Gluten-Free.

Welcome to Kickstrap

Getting Started

1 2 3

Check out the source of this page, `index.html` in your Kickstrap root to see how some of this magic is done.

If instead you'd like to jump right in, use `sample_index.html`, a (nearly) blank html document with all the default Kickstrap magic linked and ready to go.

To make css changes, your best bet is to download a less compiler and mess around in the `/extras` directory.

See the [Documentation](#) to learn more.

Back Next

Would a "fake" business have a whole closet just for office supplies? Come on.

Candy at no extra cost

To KE, Taking customer service to *the next level* means treating your customer right. That's why we provide little peppermint candies to our customers in a little glass dish at Judy's desk* at no

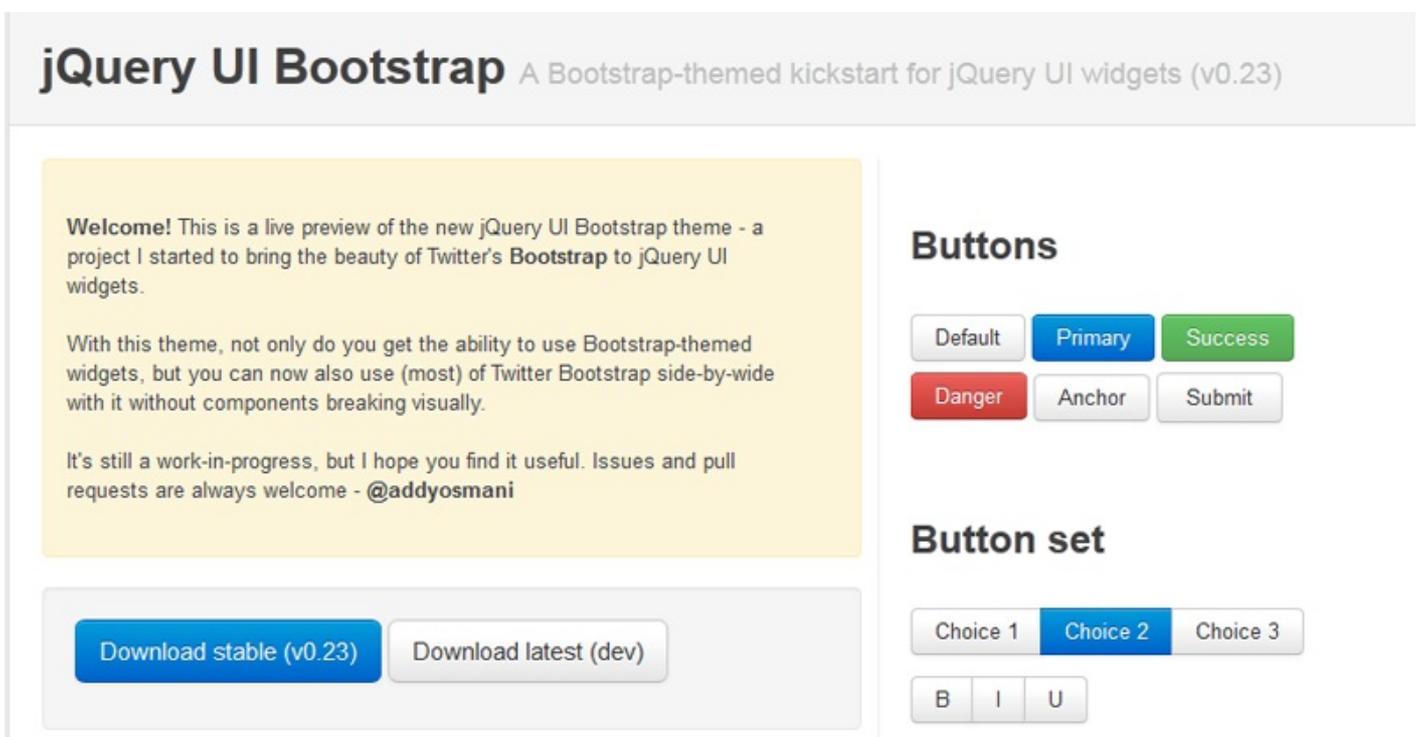
La seule question par rapport à ce projet est : sera-t-il viable dans l'avenir ?

Intégration à divers environnements

Bootstrap possède une esthétique très élégante et des efforts ont été faits pour intégrer son aspect à divers environnements ou faciliter sa mise en oeuvre.

jQuery UI Bootstrap

jQuery UI rencontre un succès croissant et un thème a été réalisé pour obtenir un aspect correspondant à celui de **Bootstrap**.



jQuery UI Bootstrap A Bootstrap-themed kickstart for jQuery UI widgets (v0.23)

Welcome! This is a live preview of the new jQuery UI Bootstrap theme - a project I started to bring the beauty of Twitter's **Bootstrap** to jQuery UI widgets.

With this theme, not only do you get the ability to use Bootstrap-themed widgets, but you can now also use (most) of Twitter Bootstrap side-by-side with it without components breaking visually.

It's still a work-in-progress, but I hope you find it useful. Issues and pull requests are always welcome - @addyosmani

Download stable (v0.23) Download latest (dev)

Buttons

Default Primary Success Danger Anchor Submit

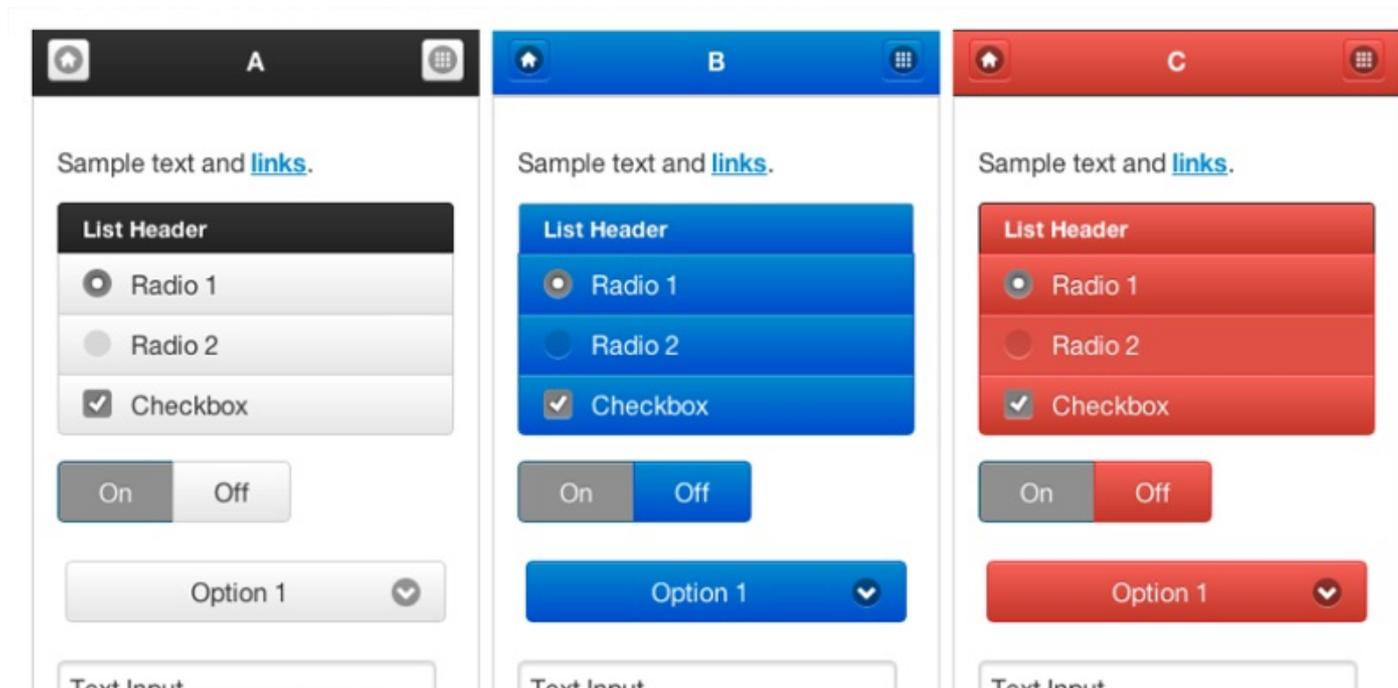
Button set

Choice 1 Choice 2 Choice 3

B I U

jQuery-Mobile-Bootstrap-Theme

On peut trouver aussi [un thème](#) pour **jQuery Mobile** :



Django-bootstrap-toolkit

Une [intégration en cours](#) pour le framework **Django** est assez prometteuse.

Thèmes pour Wordpress

On trouve beaucoup de thèmes pour **WordPress**. Voici quelques liens :

BootstrapWP

Bootstrap theme for WordPress

Bootstrap is a simple and flexible HTML, CSS, and Javascript for popular user interface components and interactions.

Now you can use it with **WordPress** as a solid base for theme development, to build custom themes quickly and easily.

Download Theme

v.86 – Release

Follow the Development on Github

v.9 – Still Baking

GitHub

Issues

RachelBaker.me

WordPress.org

Watch 216

Fork 34

Follow @rachelbaker

Roots Theme

Roots is a starting WordPress theme based on HTML5 Boilerplate & Bootstrap from Twitter.

Watch 1,699 Tweet

Roots is based on HTML5 Boilerplate, Bootstrap from Twitter, and is hNews microformat ready

You get **all** of HTML5 Boilerplate's markup (but we do place JavaScript in the header to avoid any plugin conflicts). You get most of HTML5 Boilerplate's incredibly powerful `.htaccess` directly from the theme. You get all of HTML5 Boilerplate's `style.css`.

Bootstrap from Twitter is included by default, with topbar navigation and the responsive grid ready to go.

Blog posts are based on the hNews microformat as recommended by Readability's Article Publishing Guidelines.

Extremely clean URLs for basically everything, even with WordPress

`/wp-content/themes/roots/css/` rewrites to `/css/`

`/wp-content/themes/roots/img/` rewrites to `/img/`

`/wp-content/themes/roots/js/` rewrites to `/js/`

Pretty much all WordPress generated URLs are root relative so your source isn't filled with `http://www.yoursite.com/` all over the place

Roots on GitHub

Download Roots

Last updated May 3, 2012

Merge pull request #432 from ello/disable_relative_urls

 DIGITALNATURE
THEMES
SUPPORT

HOME » THEMES » BOOTSTRAP

Bootstrap 0.9.3

Raw Atom theme built around Twitter's [Bootstrap](#) CSS framework.

This one uses the Atom 2.0 core, which is in beta stage. So back-up your theme settings before doing upgrades, because most likely you will loose them, as they are not final. **You should only use this theme for testing and experimental purposes until 1.0 is released.**

A basic demo can be seen on wp.digitalnature.eu (it's the main site of a multisite installation).

Comments, suggestions and (less obvious) bug reports are always welcome 😊

Bootstrap for Wordpress

Wordpress theme based on Twitter Bootstrap

Thème pour Joomla

J'ai aussi trouvé un thème pour Joomla ici :



Bootstrap templates, extensions, code snippets, discussions and more for Joomla!

*"Joomla! Rocks" but now with "Bootstrap Awesomeness"
Bootstrap will set you free - It's a way of life!*

Extension pour Typo

On trouve aussi une extension pour **Typo 3** :

The screenshot shows the TYPO3 website's Extension Repository. The page title is "Twitter bootstrap". The description states: "Include Twitter bootstrap in your TYPO3 installation." The version is 1.1.0, marked as "Stable", and was last uploaded on February 16, 2012. It has 56 downloads. The extension key is "mn_twitter_bootstrap" and the author is "Mattias Nilsson". A "Download" button is visible at the bottom of the extension details.

Extension pour Yii

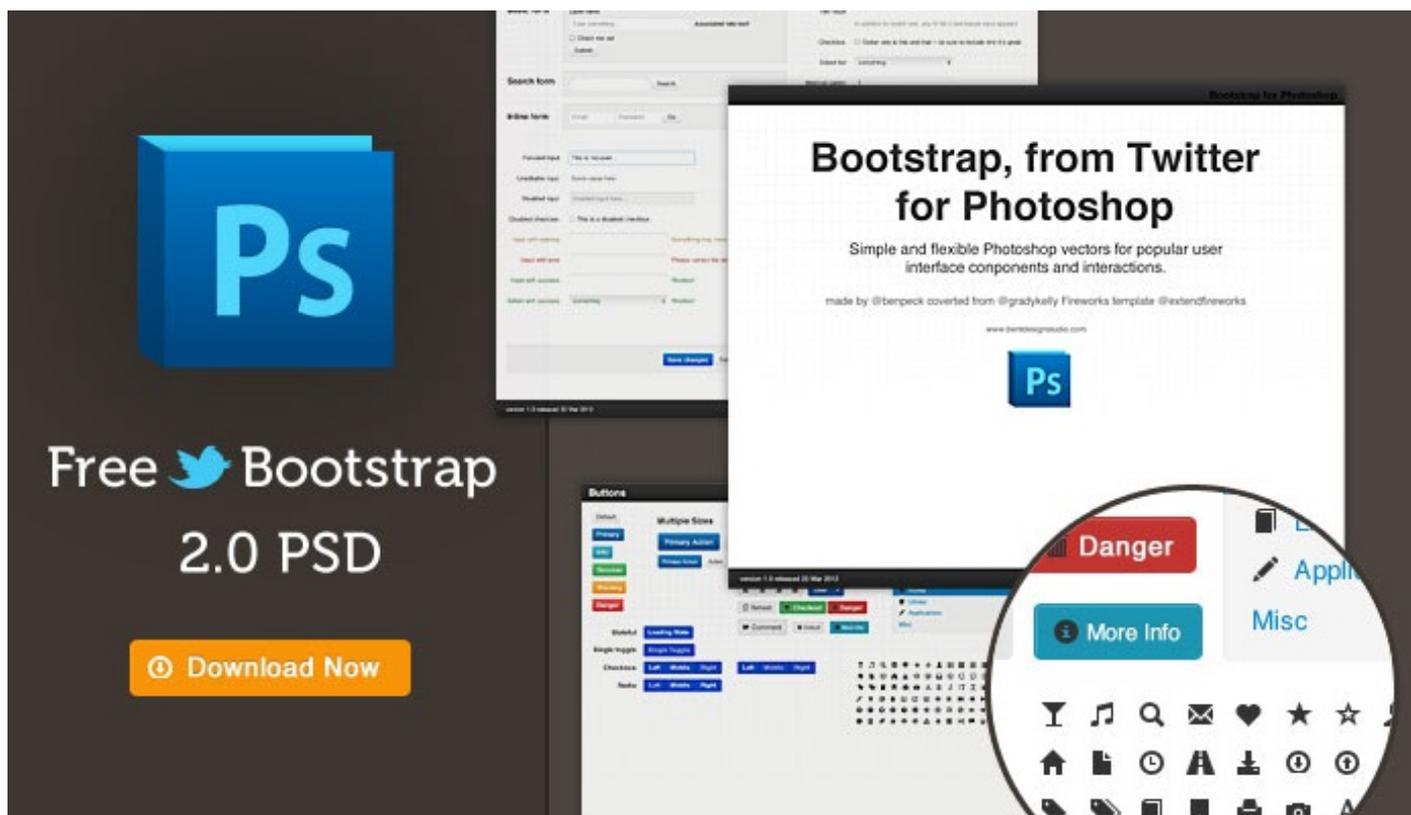
Yii est un framework **PHP** qui possède désormais un thème pour **Bootstrap** :

Yii-Bootstrap

Bringing together the **Yii PHP framework** and **Bootstrap** Twitter's new web development toolkit. Now with support for Bootstrap 2!
Yii-Bootstrap is an extension for Yii that provides a wide range of server-side widgets that allow you to easily use Bootstrap with Yii. All widgets have been developed following Yii's conventions and work seamlessly together with Bootstrap and its jQuery plugins.

Un template pour Photoshop

Un template gratuit pour photoshop ici :



Une extension pour Fireworks

Un thème pour Fireworks ici :



Une boîte à outils pour facebook

Pour développer pour facebook :

Fbootstrapp

Fbootstrapp is a toolkit designed to kickstart development of facebook iframe apps in both relevant sizes. It includes base CSS and HTML for typography, forms, buttons, tables, grids, navigation, and more, styled in the typical facebook look and feel.

[Visit the github project »](#)

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 😊.

Merci à [maxlayer44](#) pour sa relecture du chapitre sur les **responsive design** et ses conseils.