

 eBook Gratuit

# APPRENEZ

---

# twitter-bootstrap

eBook gratuit non affilié créé à partir des  
**contributeurs de Stack Overflow.**

#twitter-  
bootstrap

# Table des matières

À propos.....	1
<b>Chapitre 1: Démarrer avec twitter-bootstrap.....</b>	<b>2</b>
Remarques.....	2
Versions.....	2
Exemples.....	3
Installation / configuration.....	3
Modèle de base.....	5
Quand utiliser Bootstrap.....	6
Page Web de base utilisant des composants bootstrap.....	6
<b>Chapitre 2: Alerte.....</b>	<b>9</b>
Remarques.....	9
Exemples.....	9
Types d'alerte.....	9
Exemple de base d'alerte.....	9
Alertes Animées.....	10
Alertes Rejetables.....	10
Couleur de lien dans les alertes.....	10
<b>Chapitre 3: Badges et étiquettes Bootstrap.....</b>	<b>12</b>
Exemples.....	12
Badges.....	12
Étiquettes.....	12
<b>Chapitre 4: Bootstrap Affixe.....</b>	<b>13</b>
Exemples.....	13
Sur la barre de navigation.....	13
Affixe Exemple 2.....	13
<b>Chapitre 5: Bootstrap Dropdowns.....</b>	<b>15</b>
Paramètres.....	15
Remarques.....	15
Exemples.....	15
Comment utiliser.....	15

Exemple de base.....	16
<b>Chapitre 6: Bootstrap Navbar.....</b>	<b>17</b>
Exemples.....	17
Bootstrap Navbar.....	17
Image de marque Bootstrap.....	17
<b>Chapitre 7: Boutons.....</b>	<b>19</b>
Syntaxe.....	19
Exemples.....	19
Classes de boutons.....	19
<b>Chapitre 8: Carrousels.....</b>	<b>22</b>
Remarques.....	22
Exemples.....	22
Utilisation HTML de base.....	22
Utilisation et initialisation Javascript de base.....	23
<b>Chapitre 9: Classes d'utilité.....</b>	<b>24</b>
Exemples.....	24
Générer des classes .hidden- * pour tous les points d'arrêt - SCSS.....	24
<b>Chapitre 10: Colonnes.....</b>	<b>25</b>
Exemples.....	25
Colonnes réactives même hauteur (CSS ou SASS uniquement).....	25
<b>Chapitre 11: Composants Bootstrap.....</b>	<b>29</b>
Remarques.....	29
Exemples.....	29
Exemples de composants bootstrap.....	29
<b>Chapitre 12: Conteneurs Bootstrap.....</b>	<b>30</b>
Introduction.....	30
Exemples.....	30
Conteneurs.....	30
<b>Chapitre 13: Dialogues modaux.....</b>	<b>31</b>
Remarques.....	31
Exemples.....	31

Utilisation HTML de base.....	31
Utilisation et initialisation Javascript de base.....	31
<b>Chapitre 14: Formes.....</b>	<b>33</b>
Exemples.....	33
Forme basique.....	33
Entrées en lecture seule et désactivées.....	33
<b>Chapitre 15: Glyphicons.....</b>	<b>34</b>
Remarques.....	34
Exemples.....	34
Comment utiliser les glyphicons.....	34
<b>Chapitre 16: Groupe de liste.....</b>	<b>36</b>
Remarques.....	36
Exemples.....	36
Exemple de base.....	36
Badges.....	36
Articles liés.....	36
Éléments de bouton.....	36
Articles désactivés.....	37
Classes contextuelles.....	37
Contenu personnalisé.....	37
<b>Chapitre 17: Imbrication de la grille.....</b>	<b>38</b>
Introduction.....	38
Remarques.....	38
Exemples.....	38
Colonnes d'imbrication.....	38
<b>Chapitre 18: Impression dans Bootstrap.....</b>	<b>39</b>
Exemples.....	39
Utilisation HTML de base.....	39
<b>Chapitre 19: Info-bulle.....</b>	<b>40</b>
Remarques.....	40
Exemples.....	40
Info-bulles de positionnement.....	40

Exemple de base.....	40
<b>Chapitre 20: Jumbotron.....</b>	<b>42</b>
Introduction.....	42
Remarques.....	42
Exemples.....	42
Jumbotron de base avec deux lignes de texte et un bouton.....	42
<b>Chapitre 21: Les dropdowns.....</b>	<b>44</b>
Remarques.....	44
Exemples.....	44
Utilisation HTML de base.....	44
<b>Chapitre 22: les tables.....</b>	<b>45</b>
Exemples.....	45
Tableau simple.....	45
<b>Chapitre 23: les tables.....</b>	<b>46</b>
Remarques.....	46
Exemples.....	46
Table de base.....	46
Table avec un style avancé.....	46
<b>Lignes rayées.....</b>	<b>46</b>
<b>Table bordée.....</b>	<b>47</b>
<b>Survolez les lignes.....</b>	<b>47</b>
<b>Table condensée.....</b>	<b>47</b>
<b>Classes contextuelles.....</b>	<b>47</b>
Tables réactives.....	48
Tableau Reflow - En-têtes verticaux.....	48
<b>Chapitre 24: Menus de navigation.....</b>	<b>50</b>
Exemples.....	50
Menu Pilule Horizontale.....	50
Menu Pilule Verticale.....	50
Pilule horizontale réactive pleine largeur.....	50
<b>Chapitre 25: Migration vers Bootstrap 4.....</b>	<b>51</b>

Introduction.....	51
Remarques.....	51
Exemples.....	51
Modification de la disposition des colonnes du système de grille dans Bootstrap 4.....	51
Mise en page de grille Bootstrap 4.....	52
Modification du support du navigateur.....	52
Suppression de classe.....	52
Bootstrap 4 Navbar.....	55
Bootstrap 3 à Bootstrap 4 Changements CSS.....	58
Bootstrap 4 Alignement vertical.....	60
Bootstrap 4 Centrage.....	62
<b>Centre horizontal.....</b>	<b>62</b>
<b>Centre vertical.....</b>	<b>63</b>
Bootstrap 4 Column Order.....	63
<b>Chapitre 26: Modals.....</b>	<b>65</b>
Remarques.....	65
Exemples.....	65
HTML modal de base.....	65
<b>Chapitre 27: Navbar.....</b>	<b>66</b>
Exemples.....	66
Base Navbar (fixée en haut de la page).....	66
Sous-menu dans la barre de navigation.....	66
Diviseur de barre de navigation.....	67
Garder le lien de navigation actuel "actif".....	67
Changer le point d'arrêt Navbar (mobile vs normal).....	67
Fermez la barre de navigation réduite lorsque vous cliquez en dehors de la barre de naviga.....	68
<b>Chapitre 28: Navs.....</b>	<b>69</b>
Exemples.....	69
Navs Bootstrap.....	69
<b>Chapitre 29: Onglets.....</b>	<b>70</b>
Exemples.....	70
HTML de base.....	70

Onglets animés.....	70
<b>Chapitre 30: Pagination.....</b>	<b>72</b>
Introduction.....	72
Exemples.....	72
Un exemple simple de pagination.....	72
<b>Chapitre 31: Panneaux.....</b>	<b>73</b>
Remarques.....	73
Exemples.....	73
Exemple de base.....	73
Panneau avec cap.....	73
Panneau avec pied de page.....	73
<b>Chapitre 32: Système de grille.....</b>	<b>75</b>
Introduction.....	75
Remarques.....	75
Exemples.....	75
Requêtes médias.....	75
Niveaux de grille Bootstrap (points d'arrêt).....	75
Lignes et colonnes bootstrap.....	77
Conteneurs.....	78
Colonnes de compensation.....	79
Manipulation de l'ordre des colonnes à l'aide de push and pull.....	79
<b>Chapitre 33: Thèmes Bootstrap.....</b>	<b>81</b>
Exemples.....	81
Thèmes d'amorçage et remplacements de règles.....	81
<b>Chapitre 34: Twitter Personnalisation du style Bootstrap.....</b>	<b>82</b>
Remarques.....	82
Exemples.....	82
Remplacement du CSS par défaut.....	82
<b>Chapitre 35: Utiliser Clearfix dans les lignes et les colonnes.....</b>	<b>84</b>
Introduction.....	84
Remarques.....	84

Exemples.....	84
La première tentative naïve.....	84
Le problème de la hauteur.....	86
<b>Clearfix à la rescousse.....</b>	<b>87</b>
<b>Un tableau de bord.....</b>	<b>90</b>
2,4,6 Disposition avec Clearfixes.....	95
Pourquoi les colonnes bootstrap dépasseraient-elles 12 dans une rangée?.....	97
<b>Chapitre 36: Validation bootstrap.....</b>	<b>100</b>
Remarques.....	100
Exemples.....	100
Utilisation d'ASP.NET MVC et d'annotations de données.....	100
Exemple de saisie nécessitant une validation.....	101
Optionnel.....	101
<b>Crédits.....</b>	<b>103</b>



---

# À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [twitter-bootstrap](#)

It is an unofficial and free twitter-bootstrap ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official twitter-bootstrap.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapitre 1: Démarrer avec twitter-bootstrap

## Remarques

Bootstrap est un framework HTML, CSS et JavaScript utilisé pour créer des sites Web créés à l'aide d'un paradigme de premier mobile, ainsi que de la conception Web réactive (RWD). Il utilise une combinaison de classes CSS prédéfinies et de JavaScript pour créer diverses choses sur le Web. Il comprend des éléments tels qu'une grille personnalisée et réactive qui permet d'afficher les sites Web sur n'importe quel écran, une barre de navigation déroulante capable d'être réactive et même des éléments simples qui peuvent prendre beaucoup de temps, comme des boutons, des peu.

Bootstrap peut être utile pour les raisons suivantes:

1. **Gain de temps** : Bootstrap comporte de nombreux éléments pré-construits, qui doivent simplement être utilisés lors de l'écriture du code. Cela permet de gagner un temps considérable et de réduire considérablement le temps nécessaire pour coder un site Web.
2. **Conçu en tenant compte de la conception Web réactive** : Bootstrap permet aux développeurs Web de ne pas se soucier de créer des éléments qui évolueront avec la taille de leur écran, car Bootstrap utilise une conception réactive qui leur permet de créer des éléments qui fonctionneront sur n'importe quel écran. Taille.
3. **Simplifie le processus de conception** : Bootstrap est livré avec des éléments dotés de bonnes pratiques de conception. Cela peut être utile pour ceux dont les compétences en conception Web ne sont pas excellentes, ou pour ceux qui considèrent le design comme une tâche fastidieuse, car de nombreuses classes Bootstrap sont esthétiquement agréables et idéales à regarder.

---

Cette section devrait mentionner tous les grands sujets dans twitter-bootstrap et établir un lien avec les sujets connexes. La documentation de twitter-bootstrap étant nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

## Versions

Version	Date de sortie
4.0	2999-01-01
3.3.7	2016-07-25
3.3.6	2015-11-24
3.3.5	2015-06-15

Version	Date de sortie
3.3.4	2015-03-16
3.3.1	2014-11-12
3.3.0	2014-10-29
3.2.0	2014-06-26
3.1.0	2014-01-30
3.0	2013-08-19
2.3	2013-02-07
2.2	2012-10-29
2.1	2012-08-20
2.0	2012-02-01
1.0	2011-08-18

## Exemples

### Installation / configuration

Téléchargement:

- Télécharger Bootstrap [directement](#) ou cloner, etc. depuis le dépôt [GitHub](#)
- Téléchargez votre version personnalisée de Bootstrap à partir des [documents](#) officiels
- Installer avec bower: `bower install bootstrap`
- Installer avec npm: `npm install bootstrap`
- Installer avec compositeur: `composer require twbs/bootstrap`

### La structure de fichier

```

bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.min.css
│   ├── bootstrap-theme.css
│   └── bootstrap-theme.min.css
├── js/
│   ├── bootstrap.js
│   └── bootstrap.min.js
├── fonts/
│   ├── glyphicons-halflings-regular.eot
│   ├── glyphicons-halflings-regular.svg
│   ├── glyphicons-halflings-regular.ttf
│   └── glyphicons-halflings-regular.woff

```

### Installation:

Dans votre page HTML, incluez les CSS, JS et la dépendance de jQuery de Bootstrap (avant la version 3, au moins depuis la dernière version de Bootstrap). Veuillez noter que si vous prévoyez d'utiliser les fonctionnalités JavaScript de Bootstrap, votre référence jQuery doit figurer *avant* votre référence bootstrap.js dans votre code HTML.

Vous pouvez utiliser vos fichiers Bootstrap installés à partir de la section ci-dessus ou référencer un CDN fourni par les créateurs de Bootstrap (liens tirés de [Premiers pas avec Bootstrap](#)):

```

<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">

<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-
theme.min.css" integrity="sha384-
rHyoNliRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp" crossorigin="anonymous">

<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA71l2mCWNIPg9mGCD8wGNICPD7Txa"
crossorigin="anonymous"></script>

```

### Une page Web Bootstrap très simple:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come
*after* these tags -->

```

```

<title>Bootstrap 101 Template</title>

<!-- Bootstrap -->
<link href="css/bootstrap.min.css" rel="stylesheet">

<!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->
</head>
<body>
  <h1>Hello, world!</h1>

  <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <!-- Include all compiled plugins (below), or include individual files as needed -->
  <script src="js/bootstrap.min.js"></script>
</body>
</html>

```

## Modèle de base

```

<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must
    come *after* these tags -->

    <title>Bootstrap 101 Template</title> <!-- The title of the Website -->

    <!-- Reference to Bootstrap's CSS file -->
    <!-- This is the line to reference the bootstrap's Stylesheet -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!-- [if lt IE 9] -->
    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <!--[endif]-->

  </head>

  <body>

    <h1>Hello, world!</h1>

    <!-- Referencing jQuery (necessary for Bootstrap JavaScript plugins(bootstrap.min.js)

```

```
to work) -->
    <script src =
"https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>

    <!-- Referencing Javascript Bootstrap Plugin to Facilitate Bootstrap Animations and
functionalities. -->
    <!-- (Necessary to run Bootstrap) -->
    <script src="js/bootstrap.min.js"></script>

</body>
</html>
```

## Quand utiliser Bootstrap

Bootstrap est un framework avisé pour HTML, CSS et Javascript. Il contient un style et des fonctionnalités de base pour ce qui est devenu des éléments [Interface utilisateur] acceptés, tels que des éléments de formulaire, des boutons, des fenêtres modales et des éléments de navigation.

Bootstrap est un framework Web réactif, ce qui signifie qu'il est conçu pour adapter la mise en page et la conception à des tailles d'écran grandes et petites, telles que les appareils mobiles, les tablettes et les ordinateurs de bureau, le tout dans une base de code unique.

L'un des concepts fondamentaux de Bootstrap est le cadre de la grille. En appliquant des classes à des éléments HTML, il est possible de créer des mises en page complexes en utilisant une grille de base de douze colonnes. Par exemple, une disposition à quatre colonnes peut être adaptée à deux colonnes sur les tablettes et à une colonne sur les périphériques mobiles. La grille utilise `media queries`, une méthode CSS pour cibler des tailles d'écran spécifiques, pour y parvenir.

Bootstrap fonctionne particulièrement bien si:

- La conception personnalisée n'est pas une priorité absolue
- Vous êtes plus à l'aise pour éditer du HTML et ajouter des classes que pour créer des CSS personnalisés
- Vous êtes à l'aise avec un framework qui aura de nombreuses similitudes visuelles avec de nombreux autres sites Web.

Bootstrap peut être utilisé par les nouveaux utilisateurs de HTML, CSS et Javascript, car la [documentation](#) est excellente. Cependant, il existe une courbe d'apprentissage pour ceux qui ne sont pas entièrement à l'aise avec les trois technologies de base utilisées par Bootstrap (HTML, CSS et Javascript).

Il est possible d'acheter ou de télécharger des thèmes Bootstrap afin de modifier le style ou la fonctionnalité de Bootstrap. Il est également possible d'utiliser Bootstrap comme point de départ, avec une personnalisation de CSS et de Javascript.

## Page Web de base utilisant des composants bootstrap

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head content must come
*after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <!-- Fixed navbar -->
    <nav class="navbar navbar-default navbar-fixed-top">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target="#navbar" aria-expanded="false" aria-controls="navbar">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="#">Project name</a>
        </div>
        <div id="navbar" class="navbar-collapse collapse">
          <ul class="nav navbar-nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#about">About</a></li>
            <li><a href="#contact">Contact</a></li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button"
aria-haspopup="true" aria-expanded="false">Dropdown <span class="caret"></span></a>
              <ul class="dropdown-menu">
                <li><a href="#">Action</a></li>
                <li><a href="#">Another action</a></li>
                <li><a href="#">Something else here</a></li>
                <li role="separator" class="divider"></li>
                <li class="dropdown-header">Nav header</li>
                <li><a href="#">Separated link</a></li>
                <li><a href="#">One more separated link</a></li>
              </ul>
            </li>
          </ul>
          <ul class="nav navbar-nav navbar-right">
            <li><a href="../navbar/">Default</a></li>
            <li><a href="../navbar-static-top/">Static top</a></li>
            <li class="active"><a href="#">Fixed top <span class="sr-
only">(current)</span></a></li>
          </ul>
        </div><!--/.nav-collapse -->
      </div>
    </nav>
  </body>

```

```

</nav>

<div class="container">

  <div class="jumbotron">
    <h1>Navbar example</h1>
    <p>This example is a quick exercise to illustrate how the default, static and
fixed to top navbar work. It includes the responsive CSS and HTML, so it also adapts to your
viewport and device.</p>
    <p>To see the difference between static and fixed top navbars, just scroll.</p>
    <p>
      <a class="btn btn-lg btn-primary" href="../../components/#navbar"
role="button">View navbar docs &raquo;</a>
    </p>
  </div>

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

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

Lire Démarrer avec twitter-bootstrap en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/818/demarrer-avec-twitter-bootstrap>



---

# Chapitre 2: Alerte

## Remarques

Voir plus: <http://getbootstrap.com/components/#alerts>

## Exemples

### Types d'alerte

Contrairement à d'autres composants Bootstrap tels que les [boutons](#) , les [alertes](#) ne sont **pas** associées à un style `default` ou `primary` , car elles sont destinées à alerter l'utilisateur d'une manière spécifique.

```
<div class="alert alert-success" role="alert">
  Some action was completed successfully
</div>
<div class="alert alert-info" role="alert">
  Here is some information. Just FYI.
</div>
<div class="alert alert-warning" role="alert">
  Careful! You're about to do something dangerous.
</div>
<div class="alert alert-danger" role="alert">
  An error (or something dangerous) happened!
</div>
```

### Exemple de base d'alerte

```
<div class="container">
  <h2>Alerts</h2>
  <div class="alert alert-success">
    <strong>Success!</strong>
  </div>
  <div class="alert alert-info">
    <strong>Info!</strong>
  </div>
  <div class="alert alert-warning">
    <strong>Warning!</strong> All foelds are required
  </div>
  <div class="alert alert-danger">
    The username is required and can't be empty
  </div>
</div>
```

# Alerts

Success!

Info!

Warning! This alert box could indicate a warning that might need attention.

The username is required and can't be empty

## Alertes Animées

Les classes `.fade` et `.in` ajoutent un effet de fondu lors de la fermeture du message d'alerte.

```
<div class="alert alert-success fade in">
  <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
  <strong>Success!</strong> This is a good example!
</div>
```

## Alertes Rejetables

Pour donner une alerte à la fonctionnalité de fermeture, tout ce dont nous avons besoin est d'ajouter `data-dismiss="alert"` à notre bouton de fermeture.

```
<div class="alert alert-info alert-dismissible" role="alert">
  <button type="button" class="close" data-dismiss="alert" aria-label="Close">
    <span aria-hidden="true">&times;</span>
  </button>
  Sphinx of black quartz, judge my vow
</div>
```

Sphinx of black quartz, judge my vow



`.alert-dismissible` et `.close` sont facultatives, mais ne sont utiles que pour le style.

## Couleur de lien dans les alertes

Pour fournir rapidement une couleur correspondante pour les liens à l'intérieur d'une alerte, nous

pouvons utiliser la classe d'utilitaire `.alert-link` .

```
<div class="alert alert-success">
  You have won! Click <a href="#" class="alert-link">here</a> to claim your prize ...
</div>

<div class="alert alert-info">
  You might want to check <a href="#" class="alert-link">this</a> instead.
</div>

<div class="alert alert-warning">
  You are running out of coins. Buy more <a href="#" class="alert-link">here</a>.
</div>

<div class="alert alert-danger">
  Something went wrong. You can try <a href="#" class="alert-link">again</a> or ...
</div>
```

Congratulations! You have **WON!** Click **here** to claim your prize ...

You might want to check **this** instead.

You are running out of coins. Buy more **here**.

Something went wrong. You can try **again** or ...

Lire Alerte en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6434/alerte>

---

# Chapitre 3: Badges et étiquettes Bootstrap

## Exemples

### Badges

Les badges sont des indicateurs numériques du nombre d'éléments associés à un lien:

Utilisez la classe `.badge` dans les éléments `<span>` pour créer des badges:

```
<a href="#">News <span class="badge">5</span></a><br>
<a href="#">Comments <span class="badge">10</span></a><br>
<a href="#">Updates <span class="badge">2</span></a>
```

### Badge dans le bouton

```
<button type="button" class="btn btn-primary">Primary <span class="badge">7</span></button>
```

### Étiquettes

Les étiquettes sont utilisées pour fournir des informations supplémentaires sur quelque chose:

Utilisez la `.label` classe, suivi par l'un des six classes contextuelles `.label-default`, `.label-primary`, `.label-success`, `.label-info`, `.label-warning` ou `.label-danger`, dans un `<span>` élément pour créer une étiquette:

```
<h1>Example <span class="label label-default">New</span></h1>
<h2>Example <span class="label label-default">New</span></h2>
<h3>Example <span class="label label-default">New</span></h3>
<h4>Example <span class="label label-default">New</span></h4>
<h5>Example <span class="label label-default">New</span></h5>
<h6>Example <span class="label label-default">New</span></h6>
```

Lire Badges et étiquettes Bootstrap en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/7867/badges-et-etiquettes-bootstrap>

# Chapitre 4: Bootstrap Affixe

## Exemples

### Sur la barre de navigation

Html:

```
<nav class="navbar navbar-default" data-offset-top="120" data-spy="affix" >
  ...
</nav>
```

Css:

```
<style>
.navbar {
  background-color: red;
}
.navbar.affix {
  background-color: green;
}
</style>
```

### Affixe Exemple 2

```
<div class="container" id="con">
  <div class="row">
    <div class="span12">
      <div class="well">
        <h1> Header </h1>
      </div>
    </div>
  </div>
</div>
<div class="container" data-spy="affix" data-offset-top="400" id="nav">
  <div class="navbar">
    <div class="navbar-inner">
      <div class="container">
        <div class="span12">
          <a class="brand" href="#">Home</a>
          <ul class="nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="container">
  <div class="span3">
```

```
<p style="height:1000px;padding:10px;">

    Long scrolling text here... Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd
idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo
bar foor foo. Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo.
Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo. Ovi lispmd idr.
Blah goo bar foor foo. Ovi lispmd idr. Blah goo bar foor foo.

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

<style>

#con .well {
    height:400px;
}

#nav.affix {
    position: fixed;
    top: 0;
    width: 100%
}
</style>

<script>
    $('#nav').affix();
</script>
```

Lire Bootstrap Affixe en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6639/bootstrap-affixe>

# Chapitre 5: Bootstrap Dropdowns

## Paramètres

Les méthodes	Exemple
Appelez via Javascript	<code>\$('.dropdown-toggle').dropdown();</code>
Bascule la liste déroulante	<code>\$('.dropdown-toggle').dropdown('toggle')</code>
Type d'événement	La description
show.bs.dropdown	Cet événement se déclenche immédiatement lorsque la méthode d'instance show est appelée.
affiché.bs.dropdown	Cet événement est déclenché lorsque l'utilisateur a rendu visible la liste déroulante (attendra que les transitions CSS soient terminées).
hide.bs.dropdown	Cet événement est déclenché immédiatement lorsque la méthode d'instance hide a été appelée.
hidden.bs.dropdown	Cet événement est déclenché lorsque la liste déroulante a été masquée par l'utilisateur (attendra la fin des transitions CSS).
<b>Exemple de gestionnaire d'événement</b>	<code>\$(element).on('show.bs.dropdown', function () { // do something... })</code>

## Remarques

Lorsque vous appelez Dropdown via Javascript `$('.dropdown-toggle').dropdown()` Dropdown `$('.dropdown-toggle').dropdown()` Dropdown `$('.dropdown-toggle').dropdown()`, le data-api ie `data-toggle="dropdown"` toujours requis. [Lire la suite](#)

## Exemples

### Comment utiliser

Utilisez la classe `.dropdown` sur l'élément parent du menu déroulant.

Ajoutez la classe `.dropdown-menu` à un élément pour initialiser le plug-in du menu déroulant.

Appelez le plug-in en utilisant la classe `.dropdown-toggle` et l'attribut `data-toggle="dropdown"` sur un bouton ou un lien hypertexte.

## Exemple de base

```
<div class="dropdown">
  <button class="btn btn-primary dropdown-toggle" type="button" data-
toggle="dropdown">Dropdown Example
  <span class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#">Option One</a></li>
    <li><a href="#">Option two</a></li>
    <li><a href="#">More Options</a></li>
  </ul>
</div>
```

Lire Bootstrap Dropdowns en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6361/bootstrap-dropdowns>



# Chapitre 6: Bootstrap Navbar

## Exemples

### Bootstrap Navbar

Voici l'exemple de la version 3 de la barre de navigation de Bootstrap:

```
<nav class="navbar navbar-default" role="navigation">
<div class="container-fluid">
  <!-- Brand and toggle get grouped for better mobile display -->
  <div class="navbar-header">
    <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-ex1-collapse">
      <span class="sr-only">Toggle navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="#">Title</a>
  </div>

  <!-- Collect the nav links, forms, and other content for toggling -->
  <div class="collapse navbar-collapse navbar-ex1-collapse">
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
    <form class="navbar-form navbar-left" role="search">
      <div class="form-group">
        <input type="text" class="form-control" placeholder="Search">
      </div>
      <button type="submit" class="btn btn-default">Submit</button>
    </form>
    <ul class="nav navbar-nav navbar-right">
      <li><a href="#">Link</a></li>
      <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-toggle="dropdown">Dropdown <b
class="caret"></b></a>
        <ul class="dropdown-menu">
          <li><a href="#">Action</a></li>
          <li><a href="#">Another action</a></li>
          <li><a href="#">Something else here</a></li>
          <li><a href="#">Separated link</a></li>
        </ul>
      </li>
    </ul>
  </div><!-- /.navbar-collapse -->
</div>
```

### Image de marque Bootstrap

```
<nav class="navbar navbar-default">
  <div class="container-fluid">
```

```
<div class="navbar-header">
  <a class="navbar-brand" href="#">
    
  </a>
</div>
</div>
</nav>
```

Lire Bootstrap Navbar en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6233/bootstrap-navbar>

# Chapitre 7: Boutons

## Syntaxe

- Classes: `.btn-default` | `.btn-primary` | `.btn-success` | `.btn-info` | `.btn-warning` | `.btn-danger` | `.btn-link`;
- Tailles: `.btn-lg` | `.btn-md` | `.btn-sm` | `.btn-xs`;
- Etat: actif | disséqué.

## Exemples

### Classes de boutons

Bootstrap fournit plusieurs classes pour les boutons de style et les fait ressortir.

Les boutons d'amorçage peuvent être créés en ajoutant la classe `.btn` à un élément.

Classe Bootstrap	Rôle (couleur)
<code>.btn-default</code>	Bouton standard (blanc)
<code>.btn-primary</code>	Fournit un poids visuel supplémentaire et identifie l'action principale (bleu)
<code>.btn-success</code>	Utilisé pour indiquer une action réussie (vert)
<code>.btn-info</code>	Bouton contextuel pour fournir des informations (bleu clair)
<code>.btn-warning</code>	Indique que la prudence doit être appliquée par l'utilisateur (jaune)
<code>.btn-danger</code>	Indique une action dangereuse ou négative (rouge)
<code>.btn-link</code>	Faites en sorte que votre bouton ressemble à une balise d'ancrage.

### Taille des boutons

Vous pouvez également créer différentes tailles de boutons avec les classes de `.btn-size`

Classe Bootstrap	Résultat
<code>.btn-lg</code>	Crée un bouton de plus grande taille
<code>.btn-sm</code>	Crée un bouton de taille plus petite

Classe Bootstrap	Résultat
<code>.btn-xs</code>	Crée un petit bouton supplémentaire
<code>.btn-block</code>	Les boutons deviennent des éléments de niveau bloc et couvrent toute la largeur de leur parent

## Activer le bouton

La classe `active` fera apparaître un bouton pressé.

```
<button type="button" class="btn btn-primary active">Active Primary</button>
```

## Désactiver un bouton

L'ajout de la classe `disabled` à un bouton rendra le bouton non cliquable et affichera un curseur interdit lors du survol.

```
<button type="button" class="btn btn-primary disabled">Disabled Primary</button>
```

## Rendez les boutons horizontalement ensemble

Plusieurs boutons peuvent être rendus horizontalement avec la classe `.btn-group`. Enveloppez simplement vos boutons dans un élément container et attribuez à cet élément la classe `btn-group`.

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apples</button>
  <button type="button" class="btn btn-primary">Oranges</button>
  <button type="button" class="btn btn-primary">Pineapples</button>
</div>
```

## Rendu des boutons verticalement

Appliquer la `.btn-group-vertical` à l'élément conteneur

```
<div class="btn-group-vertical">
  <button type="button" class="btn btn-primary">Apples</button>
  <button type="button" class="btn btn-primary">Oranges</button>
  <button type="button" class="btn btn-primary">Pineapples</button>
</div>
```

## Faire groupe de boutons prendre toute la largeur

Les boutons enveloppés dans un élément `.btn-group` ne prennent que la largeur requise. Pour que le groupe `.btn-group-justified` toute la largeur de l'écran, utilisez `.btn-group-justified`.

```
<div class="btn-group btn-group-justified">
  <a href="#" class="btn btn-primary">Apples</a>
  <a href="#" class="btn btn-primary">Oranges</a>
  <a href="#" class="btn btn-primary">Pineapples</a>
</div>
```

```
</div>
```

Lire Boutons en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/4757/boutons>

# Chapitre 8: Carrousels

## Remarques

Pour plus d'informations, consultez la documentation officielle à l' [adresse](http://getbootstrap.com/javascript/#carousel) <http://getbootstrap.com/javascript/#carousel> , d'où proviennent les exemples et les informations de base sur l'utilisation de HTML et de Javascript.

Il convient de noter que les carrousels ne fonctionnent pas correctement dans IE 9 et les versions antérieures en raison de l'utilisation de transitions / animations CSS3.

## Exemples

### Utilisation HTML de base

Un carrousel Bootstrap est un composant Bootstrap qui crée un diaporama qui fait défiler les éléments du carrousel.

Voici un exemple d'utilisation basique de HTML:

```
<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>
    <li data-target="#carousel-example-generic" data-slide-to="1"></li>
    <li data-target="#carousel-example-generic" data-slide-to="2"></li>
  </ol>

  <!-- Wrapper for slides -->
  <div class="carousel-inner" role="listbox">
    <div class="item active">
      
      <div class="carousel-caption">
        ...
      </div>
    </div>
    <div class="item">
      
      <div class="carousel-caption">
        ...
      </div>
    </div>
    ...
  </div>

  <!-- Controls -->
  <a class="left carousel-control" href="#carousel-example-generic" role="button" data-
slide="prev">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="right carousel-control" href="#carousel-example-generic" role="button" data-
```

```
slide="next">
  <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
  <span class="sr-only">Next</span>
</a>
</div>
```

## Utilisation et initialisation Javascript de base

Les composants du carrousel peuvent être instanciés via jQuery avec la fonction `$('.carousel').carousel(options)`, où `$('.carousel')` est une référence de niveau supérieur au carrousel spécifique et `options` un objet Javascript spécifiant le carrousel attributs par défaut.

L'objet `options` permet de définir plusieurs propriétés qui affecteront le comportement du carrousel. Ces propriétés sont définies en tant que telles:

- La propriété `interval` accepte un type de `number` Javascript qui permet à un utilisateur de définir la durée pendant laquelle le carrousel affiche une diapositive de carrousel donnée. Si la valeur booléenne `false` est spécifiée, le carrousel ne fonctionnera pas automatiquement.
- La propriété `pause` accepte un type de `string` Javascript qui active le comportement lorsque le cycle automatique du carrousel est suspendu lorsque la souris de l'utilisateur entre dans le carrousel. La valeur par défaut (et unique) acceptée est "hover".
- La propriété `wrap` accepte un type de `boolean` Javascript qui permet à un utilisateur de définir s'il souhaite ou non que le carrousel effectue un cycle continu sans s'arrêter sur une diapositive donnée.
- La propriété `keyboard` accepte un type de `boolean` Javascript qui permet à un utilisateur de définir s'il veut ou non que le carrousel réponde aux événements du clavier.

Voici un exemple d'utilisation de Javascript de base:

```
$('#carCarousel').carousel({ interval: 2500, pause: "hover", wrap: false, keyboard: true });
```

Comme pour les autres composants Bootstrap, les options du carrousel peuvent également être spécifiées en HTML via des attributs de données.

Lire Carrousels en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/1568/carrousels>

---

# Chapitre 9: Classes d'utilité

## Exemples

### Générer des classes `.hidden-*` pour tous les points d'arrêt - SCSS

```
// Mixin to generate hidden classes
@mixin generate-hidden-classes {
  @each $bp in map-keys($grid-breakpoints) {
    .hidden-#{$bp} {
      @include media-breakpoint-only($bp) {
        display: none !important;
      }
    }
  }
}

// Call to the mixin
@include generate-hidden-classes();
```

Lire Classes d'utilité en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6217/classes-d-utilite>



# Chapitre 10: Colonnes

## Exemples

### Colonnes réactives même hauteur (CSS ou SASS uniquement)

Vous devez ajouter une div avec la classe `.row-height` à l'intérieur de la ligne et ajouter également `.col-height` aux colonnes. Si vous souhaitez limiter l'effet à une requête multimédia `.row-height`, utilisez simplement les `.row-height responsive` `.row-height` et `.col-height` : par exemple `.row-sm-height` avec `.col-sm-height`.

Version CSS:

```
.row-height {
  display: table;
  table-layout: fixed;
  height: 100%;
  width: calc(100% + 30px);
}
.col-height {
  display: table-cell;
  float: none;
  height: 100%;
}
.col-top {
  vertical-align: top;
}
.col-middle {
  vertical-align: middle;
}
.col-bottom {
  vertical-align: bottom;
}

@media (min-width: 480px) {
  .row-xs-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: 100%;
  }
  .col-xs-height {
    display: table-cell;
    float: none;
    height: 100%;
  }
  .col-xs-top {
    vertical-align: top;
  }
  .col-xs-middle {
    vertical-align: middle;
  }
  .col-xs-bottom {
    vertical-align: bottom;
  }
}
```

```
}

@media (min-width: 768px) {
  .row-sm-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: 100%;
  }
  .col-sm-height {
    display: table-cell;
    float: none;
    height: 100%;
  }
  .col-sm-top {
    vertical-align: top;
  }
  .col-sm-middle {
    vertical-align: middle;
  }
  .col-sm-bottom {
    vertical-align: bottom;
  }
}

@media (min-width: 992px) {
  .row-md-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: calc(100% + 30px);
  }
  .col-md-height {
    display: table-cell;
    float: none;
    height: 100%;
  }
  .col-md-top {
    vertical-align: top;
  }
  .col-md-middle {
    vertical-align: middle;
  }
  .col-md-bottom {
    vertical-align: bottom;
  }
  .row-md-height .col-md-3 {
    width: 25%;
    min-width: 25%;
    max-width: 25%;
  }
}

@media (min-width: 1200px) {
  .row-lg-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: 100%;
  }
  .col-lg-height {
```

```

    display: table-cell;
    float: none;
    height: 100%;
}
.col-lg-top {
    vertical-align: top;
}
.col-lg-middle {
    vertical-align: middle;
}
.col-lg-bottom {
    vertical-align: bottom;
}
}

```

## Version SASS (bootstrap nécessaire \_variables.scss):

```

@import "../bootstrap/variables.scss";
$sizes: xs sm md lg;
$screens: $screen-xs-min $screen-sm-min $screen-md-min $screen-lg-min;

//general
.row-height {
    display: table;
    table-layout: fixed;
    height: 100%;
    width: calc(100% + $grid-gutter-width);
}
.col-height {
    display: table-cell;
    float: none;
    height: 100%;
}
.col-top {
    vertical-align: top;
}
.col-middle {
    vertical-align: middle;
}
.col-bottom {
    vertical-align: bottom;
}

//different sizes
@for $i from 1 through length($sizes) {
    $size: nth($sizes, $i);
    $screen: nth($screens, $i);

    @media (min-width: #{$screen}) {
        .row-#{$size}-height {
            display: table;
            table-layout: fixed;
            height: 100%;
            width: 100%;
        }
        .col-#{$size}-height {
            display: table-cell;
            float: none;
            height: 100%;
        }
    }
}

```

```
.col-#{$size}-top {  
  vertical-align: top;  
}  
.col-#{$size}-middle {  
  vertical-align: middle;  
}  
.col-#{$size}-bottom {  
  vertical-align: bottom;  
}  
}  
  
}
```

Lire Colonnes en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6469/colonnes>

---

# Chapitre 11: Composants Bootstrap

## Remarques

Pour plus d'informations, consultez la documentation officielle à l' [adresse](http://getbootstrap.com/javascript/) <http://getbootstrap.com/javascript/> , d'où provient la liste des composants.

## Exemples

### Exemples de composants bootstrap

Les composants Bootstrap sont une collection de plug-ins jQuery facultatifs fournis avec Bootstrap.

Le but des composants Bootstrap est de fournir des fonctionnalités et des capacités étendues qui seraient difficiles (ou impossibles) à accomplir sans utiliser Javascript. Certains composants fournis sont purement fonctionnels, alors que certains composants sont utilisés pour définir des fonctionnalités pour certains des widgets frontaux spéciaux de Bootstrap.

Les exemples incluent des [effets de transition](#) , les [boîtes de dialogue modales](#) , [menus déroulants](#) , [scrollspy](#) , [onglets](#) , [infobulles](#) , [alertes](#) , [popovers](#) , [boutons](#) , [effondrement](#) , [carrousels](#) et [affixes](#) .

Lire Composants Bootstrap en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6054/composants-bootstrap>

---

# Chapitre 12: Conteneurs Bootstrap

## Introduction

Utilisez `.container` pour un conteneur de largeur fixe sensible.

Utilisez `.container-fluid` pour un conteneur pleine largeur couvrant toute la largeur de la fenêtre.

## Exemples

### Conteneurs

`.container` a une largeur fixe pour chaque taille d'écran dans bootstrap (xs, sm, md, lg);

`.container-fluid` se dilate pour remplir la largeur disponible.

```
@media (min-width: 568px) {  
  .container {  
    width: 550px;  
  }  
}  
  
@media (min-width: 992px) {  
  .container {  
    width: 970px;  
  }  
}  
  
@media (min-width: 1200px) {  
  .container {  
    width: 1170px;  
  }  
}
```

Selon la largeur de la fenêtre d'affichage de la page Web, la classe de conteneur lui attribue une largeur fixe spécifique.

Votre élément `.container-fluid`, quant à lui, sera redimensionné en permanence au fur et à mesure que vous apporterez les plus petites modifications à la largeur de votre navigateur.

Lire Conteneurs Bootstrap en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/10908/conteneurs-bootstrap>

# Chapitre 13: Dialogues modaux

## Remarques

Pour plus d'informations, consultez la documentation officielle à l' [adresse](http://getbootstrap.com/javascript/#modals) <http://getbootstrap.com/javascript/#modals> , où est dérivé l'exemple «Utilisation de base HTML».

## Exemples

### Utilisation HTML de base

Une boîte de dialogue modale Bootstrap est un composant Bootstrap qui crée une fenêtre de dialogue modale qui flotte sur le contenu au niveau de la page.

Voici un exemple d'utilisation basique d'un dialogue modal Bootstrap en HTML:

```
<div class="modal fade" tabindex="-1" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
        <h4 class="modal-title">Modal title</h4>
      </div>
      <div class="modal-body">
        <p>One fine body&hellip;</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div><!-- /.modal-content -->
  </div><!-- /.modal-dialog -->
</div><!-- /.modal -->
```

### Utilisation et initialisation Javascript de base

Les composants du dialogue modal peuvent être instanciés via jQuery avec la fonction `$('#myModal').modal(options)` , où `$('#myModal')` est une référence de haut niveau au dialogue modal spécifique et `options` est un objet Javascript spécifiant les attributs par défaut du dialogue modal.

L'objet `options` permet de définir plusieurs propriétés qui affecteront le comportement de la boîte de dialogue modale. Ces propriétés sont définies en tant que telles:

- La propriété `backdrop` permet à un utilisateur de définir si une superposition d'arrière-plan grise doit apparaître ou non derrière la boîte de dialogue modale. Les deux valeurs booléennes et la chaîne "static" sont reconnues. Si "static" est spécifié, la boîte de dialogue modale ne sera pas fermée lorsqu'un utilisateur clique sur la superposition d'arrière-plan.

- La propriété `keyboard` permet à un utilisateur de définir si la boîte de dialogue modale doit être fermée ou non lorsque la touche d'échappement est enfoncée sur le clavier.
- La propriété `show` permet à un utilisateur de définir si oui ou non la boîte de dialogue modale doit apparaître lorsque le modal est initialisé.

Voici un exemple d'utilisation de Javascript de base:

```
$('#carModal').modal({ backdrop: false, keyboard: true, show: false });
```

Comme avec les autres composants Bootstrap, les options du modal peuvent également être spécifiées en HTML via des attributs de données.

Lire Dialogues modaux en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/5927/dialogues-modaux>



# Chapitre 14: Formes

## Exemples

### Forme basique

Les contrôles de formulaire ont un style par défaut sans utiliser de classes spéciales.

Toutefois, les étiquettes et les contrôles peuvent être `.form-group` dans des `.form-group` pour un espacement optimal.

```
<form>
  <div class="form-group">
    <label for="input-email">Email address</label>
    <input type="email" class="form-control" id="input-email" placeholder="Email">
  </div>
  <div class="form-group">
    <label for="input-password">Password</label>
    <input type="password" class="form-control" id="input-password" placeholder="Password">
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

### Entrées en lecture seule et désactivées

Ajoutez l'attribut `readonly` pour empêcher l'entrée de l'utilisateur. Un champ en lecture seule ne peut être modifié

```
<input class="form-control" type="text" placeholder="Readonly input here..." readonly>
```

Ajouter l'attribut `disabled` pour désactiver un champ de saisie. Un champ désactivé ne peut pas non plus être modifié. Le curseur change pour le rendre plus visible.

```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled input here..." disabled>
```

Lire Formes en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6251/formes>

---

# Chapitre 15: Glyphicons

## Remarques

Cette section fournit une vue d'ensemble des glyphicons Bootstrap et explique comment utiliser les glyphicons.

## Exemples

### Comment utiliser les glyphicons

Twitter Bootstrap prend en charge les icônes appelées glyphicons et elles peuvent être utilisées avec toutes les balises HTML.

Toutes les icônes nécessitent une classe de base et une classe d'icônes individuelles.

N'oubliez pas que les classes d'icônes ne peuvent pas être combinées directement avec d'autres composants. Utilisez donc toujours la `<span></span>` interne.

Si votre code HTML contient des éléments enfants internes, vous ne pouvez pas utiliser de classes d'icônes pour cette balise particulière.

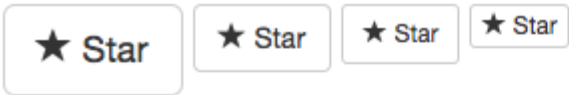
### Exemples

Par exemple, vous créez un bouton bootstrap, la syntaxe de ce bouton doit être la suivante:

```
<button type="button" class="btn btn-default btn-lg">
  Star
</button>
```

Ainsi, dans l'exemple ci-dessus, un simple bouton bootstrap est créé, mais vous souhaitez maintenant ajouter un glypicon dans ce bouton, car il suffit d'ajouter un élément `<span>` dans une `<button>`. Comme ça:

```
<button type="button" class="btn btn-default btn-lg">
  <span class="glyphicon glyphicon-star" aria-hidden="true"></span>Star
</button>
```



```
<button type="button" class="btn btn-default" aria-label="Left Align">
  <span class="glyphicon glyphicon-align-left" aria-hidden="true"></span>
</button>

<button type="button" class="btn btn-default btn-lg">
  <span class="glyphicon glyphicon-star" aria-hidden="true"></span> Star
</button>
```

Lire Glyphicons en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6098/glyphicons>

---

# Chapitre 16: Groupe de liste

## Remarques

Vous devez savoir comment utiliser les [boutons](#) bootstrap et peu d'informations sur les classes [contextuelles](#) .

## Exemples

### Exemple de base

```
<ul class="list-group">
  <li class="list-group-item">Cras justo odio</li>
  <li class="list-group-item">Dapibus ac facilisis in</li>
  <li class="list-group-item">Morbi leo risus</li>
  <li class="list-group-item">Porta ac consectetur ac</li>
  <li class="list-group-item">Vestibulum at eros</li>
</ul>
```

### Badges

```
<ul class="list-group">
  <li class="list-group-item">
    <span class="badge">14</span>
    Cras justo odio
  </li>
</ul>
```

### Articles liés

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    Cras justo odio
  </a>
  <a href="#" class="list-group-item">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item">Morbi leo risus</a>
  <a href="#" class="list-group-item">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item">Vestibulum at eros</a>
</div>
```

### Éléments de bouton

```
<div class="list-group">
  <button type="button" class="list-group-item">Cras justo odio</button>
  <button type="button" class="list-group-item">Dapibus ac facilisis in</button>
  <button type="button" class="list-group-item">Morbi leo risus</button>
  <button type="button" class="list-group-item">Porta ac consectetur ac</button>
  <button type="button" class="list-group-item">Vestibulum at eros</button>
</div>
```

## Articles désactivés

```
<div class="list-group">
  <a href="#" class="list-group-item disabled">
    Cras justo odio
  </a>
  <a href="#" class="list-group-item">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item">Morbi leo risus</a>
  <a href="#" class="list-group-item">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item">Vestibulum at eros</a>
</div>
```

## Classes contextuelles

```
<ul class="list-group">
  <li class="list-group-item list-group-item-success">Dapibus ac facilisis in</li>
  <li class="list-group-item list-group-item-info">Cras sit amet nibh libero</li>
  <li class="list-group-item list-group-item-warning">Porta ac consectetur ac</li>
  <li class="list-group-item list-group-item-danger">Vestibulum at eros</li>
</ul>
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-success">Dapibus ac facilisis in</a>
  <a href="#" class="list-group-item list-group-item-info">Cras sit amet nibh libero</a>
  <a href="#" class="list-group-item list-group-item-warning">Porta ac consectetur ac</a>
  <a href="#" class="list-group-item list-group-item-danger">Vestibulum at eros</a>
</div>
```

## Contenu personnalisé

```
<div class="list-group">
  <a href="#" class="list-group-item active">
    <h4 class="list-group-item-heading">List group item heading</h4>
    <p class="list-group-item-text">...</p>
  </a>
</div>
```

Lire Groupe de liste en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6347/groupe-de-liste>

# Chapitre 17: Imbrication de la grille

## Introduction

Dans Bootstrap, il est possible d'utiliser des colonnes de grille à l' *intérieur d'* autres colonnes. Cela est utile lors de la création de mises en page réactives avancées utilisant [plusieurs niveaux de grille](#) .

## Remarques

Nous pouvons avoir autant de colonnes que possible de la manière mentionnée ci-dessus.

## Exemples

### Colonnes d'imbrication

```
<div class="row">
  <div class="col-sm-9">
    Level 1: .col-sm-9
    <div class="row">
      <div class="col-xs-8 col-sm-6">
        Level 2: .col-xs-8 .col-sm-6
      </div>
      <div class="col-xs-4 col-sm-6">
        Level 2: .col-xs-4 .col-sm-6
      </div>
    </div>
  </div>
</div>
```

L'exemple est tiré de [ <http://getbootstrap.com/css/#grid-nesting>][1]

comme le suggère la documentation du site officiel

Pour imbriquer votre contenu avec la grille par défaut, ajoutez un nouveau `.row` et un ensemble de colonnes `.col-sm-*` dans une colonne existante `.col-sm-*`. Les lignes imbriquées doivent inclure un ensemble de colonnes de 12 au maximum (il n'est pas nécessaire d'utiliser les 12 colonnes disponibles).

Level 1: .col-sm-9	
Level 2: .col-xs-8 .col-sm-6	Level 2: .col-xs-4 .col-sm-6

Lire Imbrication de la grille en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/9088/imbrication-de-la-grille>

---

# Chapitre 18: Impression dans Bootstrap.

## Exemples

### Utilisation HTML de base

Les éléments d'impression de Bootstrap vous permettent de désigner quels éléments doivent être visibles lors de l'impression et lesquels doivent être masqués.

Pour rendre quelque chose de visible, utilisez l'une des méthodes suivantes en fonction de l'élément et de la manière dont il doit apparaître lors de l'impression:

```
.visible-print-block  
.visible-print-inline  
.visible-print-inline-block
```

Pour masquer quelque chose d'impression, utilisez ce qui suit:

```
.hidden-print
```

Lire Impression dans Bootstrap. en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6707/impression-dans-bootstrap->

# Chapitre 19: Info-bulle

## Remarques

L'info-bulle est un élément d'interface utilisateur qui ressemble à une petite fenêtre contextuelle. Il est généralement déclenché lorsqu'un utilisateur passe son pointeur sur un autre élément, sans cliquer dessus.

Pour des raisons de performances, les info-bulles doivent être initialisées avec jQuery. Le code suivant activera toutes les info-bulles dans le DOM:

```
<script>
  $(document).ready(function() {
    $('[data-toggle="tooltip"]').tooltip();
  });
</script>
```

## Exemples

### Info-bulles de positionnement

Par défaut, l'info-bulle apparaîtra au-dessus de l'élément. Nous pouvons utiliser `data-placement` attribut de `data-placement` pour définir la position de l'info-bulle en haut, en bas, à gauche ou à droite de l'élément.

```
<a href="#" data-toggle="tooltip" data-placement="top" title="Top tooltip">Hover</a>
<a href="#" data-toggle="tooltip" data-placement="bottom" title="Bottom tooltip">Hover</a>
<a href="#" data-toggle="tooltip" data-placement="left" title="Left tooltip">Hover</a>
<a href="#" data-toggle="tooltip" data-placement="right" title="Right tooltip">Hover</a>
```

Hover Hover Hover Hover 

Nous pouvons également utiliser `data-placement="auto"` pour réorienter dynamiquement l'info-bulle. L'info-bulle dans l'exemple suivant, l'info-bulle s'affichera à gauche si possible, sinon elle s'affichera correctement.

```
<a href="#" data-toggle="tooltip" data-placement="auto left" title="To the left?">Hover</a>
```

### Exemple de base

Pour créer une info-bulle, il suffit d'ajouter l'attribut `data-toggle="tooltip"` et un `title` à l'élément HTML qui aura l'infobulle. L'attribut `Title` est utilisé pour spécifier le texte affiché dans l'info-bulle.

```
<span data-toggle="tooltip" title="Hello world!">Hover over me</span>
```



Hello world!

Hover over me

Lire Info-bulle en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/3731/info-bulle>

---

# Chapitre 20: Jumbotron

## Introduction

Jumbotron est un composant standard de Bootstrap pour afficher certains contenus importants sur votre site Web. Il est généralement utilisé juste sous la barre de navigation, avant le contenu.

## Remarques

A l'intérieur du jumbotron, tous les systèmes de grille, classe de conteneur et classe de ligne fonctionnent également.

## Exemples

### Jumbotron de base avec deux lignes de texte et un bouton

Ceci est un jumbotron avec un titre, un contenu et un bouton.

#### Code

```
<div class="jumbotron">
  <h1>Title text</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec tortor ipsum, convallis sit.</p>
  <p><a class="btn btn-default" href="#" role="button">A button</a></p>
</div>
```

#### Résultat

# Title text

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Donec tortor ipsum, convallis sit.

A button

Lire Jumbotron en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/9188/jumbotron>

---

# Chapitre 21: Les dropdowns

## Remarques

Pour plus d'informations, consultez la documentation officielle de Bootstrap située à l' [adresse http://getbootstrap.com/javascript/#dropdowns](http://getbootstrap.com/javascript/#dropdowns) , d'où provient l'exemple d'utilisation HTML de base.

## Exemples

### Utilisation HTML de base

Un menu déroulant Bootstrap est un composant Bootstrap qui permet à un élément HTML de déclencher l'affichage d'une liste déroulante de sous-menu lorsque l'élément est cliqué.

Voici un exemple d'utilisation basique de HTML:

```
<div class="dropdown">
  <button id="dLabel" type="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    Dropdown trigger
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu" aria-labelledby="dLabel">
    . . .
  </ul>
</div>
```

Les éléments de sous-menus déroulants peuvent être spécifiés en insérant `li` element dans l'élément `ul` avec la classe `.dropdown-menu` .

Lire Les dropdowns en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6033/les-dropdowns>

---

# Chapitre 22: les tables

## Exemples

### Tableau simple

Bien que les effets de style puissent varier en fonction du thème, la classe `.table` est utilisée pour créer une apparence uniforme et cohérente pour les tables d'une application:

```
<table class="table">
  <tr>
    <th>Season</th>
    <th>Doctor</th>
    <th>Companion</th>
  </tr>
  <tr>
    <td>1</td>
    <td>Christopher Eccleston</td>
    <td>Rose Tyler</td>
  </tr>
</table>
```

Lire les tables en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6299/les-tables>

---

# Chapitre 23: les tables

## Remarques

**Ordre de contenu et tableaux complexes** Attention: le style de redistribution de table modifie l'ordre visuel du contenu. Assurez-vous de n'appliquer ce style qu'aux tables de données simples et **bien formées** (et en particulier, ne l'utilisez pas pour les tableaux de disposition) avec les cellules d'en-tête de table appropriées pour chaque ligne et chaque colonne.

En outre, cette classe ne fonctionnera pas correctement pour les tables comportant des cellules couvrant plusieurs lignes ou colonnes (en utilisant les attributs `rowspan` ou `colspan`).

## Exemples

### Table de base

Bootstrap définit un style personnalisé pour la table à l'aide de la classe `.table`. Ajoutez simplement la classe `.table` à n'importe quelle `<table>` pour voir les séparateurs horizontaux et le remplissage:

```
<table class="table">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

### Table avec un style avancé

Bootstrap fournit quelques classes pour un style de tableau avancé.

---

## Lignes rayées

Vous aurez une table avec des lignes rayées, si vous ajoutez `.table-striped` classe `.table-striped` :

```
<table class="table table-striped">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

Notez que:

Les tableaux rayés sont stylés via le sélecteur CSS `:nth-child`, qui n'est pas disponible dans Internet Explorer 8.

---

## Table bordée

Vous aurez une table avec des bordures sur tous les côtés de la table et des cellules, si vous ajoutez la classe `.table-bordered` :

```
<table class="table table-bordered">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

---

## Survolez les lignes

Si vous ajoutez la classe `.table-hover`, vous aurez une table avec des lignes en surbrillance lorsque l'utilisateur survolera une ligne:

```
<table class="table table-hover">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

---

## Table condensée

Si vous ajoutez `.table-condensed` classe `.table-condensed`, le remplissage de la cellule par défaut sera réduit de moitié, vous aurez donc une table plus compacte:

```
<table class="table table-condensed">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td>Bloggs</td></tr>
  </tbody>
</table>
```

---

## Classes contextuelles

Les tables bootstrap prennent en charge les couleurs contextuelles. Pour modifier la couleur d'arrière-plan d'une ligne ou d'une cellule de tableau, il vous suffit d'ajouter l'une des classes

contextuelles suivantes: `.active` , `.success` , `.info` , `.warning` , `.danger`

```
<table class="table">
  <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
  <tbody>
    <tr class="success"><td>John</td><td>Doe</td></tr>
    <tr><td>Fred</td><td class="info">Bloggs</td></tr>
  </tbody>
</table>
```

## Tables réactives

Vous devez envelopper toute `.table` dans un conteneur HTML avec la classe `.table-responsive` pour créer des tables réactives:

```
<div class="table-responsive">
  <table class="table">
    <thead><tr><th>First Name</th><th>Last name</th></tr></thead>
    <tbody>
      <tr><td>John</td><td>Doe</td></tr>
      <tr><td>Fred</td><td>Bloggs</td></tr>
    </tbody>
  </table>
</div>
```

Les tableaux réactifs défilent horizontalement sur les petits appareils (<768px). Il n'y aura pas de différences pour les écrans de plus de 768 pixels de large.

## Tableau Reflow - En-têtes verticaux

### Obtenir un tableau avec des en-têtes verticaux.

Le bootstrap Twitter prend désormais en charge l'en-tête vertical sur une table normale bien formatée. Pour ce faire, utilisez `.table-reflow` classe `.table-reflow`

Utilisez la classe twitter bootstrap `.table-reflow` sur une table bien formée pour obtenir une table avec des en-têtes verticales. De plus, vous pouvez combiner avec l'utilisation de `.table-striped` et `.table-hover` pour le `.table-hover` colonnes cette fois.

```
<table class="table table-striped table-hover table-reflow">
  <thead>
    <tr>
      <th ><strong> First Name: </strong></th>
      <th ><strong> Last Name: </strong></th>
      <th ><strong> Email: </strong></th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td> John </td>
      <td> Doe </td>
      <td> john.doe@email.com </td>
    </tr>
  </tbody>
</table>
```



```
        <td> Joane </td>
        <td> Donald </td>
        <td> jane@email.com </td>
    </tr>
</tbody>
</table>
```

Vous devriez vérifier les documents Alpha v4 ici: [twitter-bootstrap .table-reflow](#)

Lire les tables en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6360/les-tables>

---

# Chapitre 24: Menus de navigation

## Exemples

### Menu Pilule Horizontale

```
<ul class="nav nav-pills">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

### Menu Pilule Verticale

```
<ul class="nav nav-pills nav-stacked">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

### Pilule horizontale réactive pleine largeur

```
<ul class="nav nav-tabs nav-justified">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

Lire Menus de navigation en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6404/menus-de-navigation>

---

# Chapitre 25: Migration vers Bootstrap 4

## Introduction

Bootstrap 4 est une [réécriture majeure](#) et de nombreuses modifications sont à prendre en compte lors de la mise à niveau depuis Bootstrap 3. Voici les changements de noms de classes, des astuces et des exemples de migration de votre code Bootstrap 3.x vers Bootstrap 4.x.

## Remarques

Ceci est juste un petit exemple des exemples plus détaillés à suivre.

## Exemples

### Modification de la disposition des colonnes du système de grille dans Bootstrap 4

Le premier bloc de code est écrit dans Bootstrap 3. Dans Bootstrap 3, il existe 4 types de spécifications de colonne, à savoir `col-md-*` `col-lg-*` `col-sm-*` `col-xs-*`. Une disposition entièrement réactive ressemblera à ceci dans Bootstrap 3:

```
<div class="row">
  <div class="col-lg-4 col-md-8 col-sm-8 col-xs-8">
    contents
  </div>
  <div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
    contents
  </div>
</div>
```

Dans Bootstrap 4, ils ont ajouté un nouveau niveau de grille inférieure à 768 pixels pour un contrôle plus précis. Donc, Bootstrap 4 a `col-*` (`xs`), `col-sm-*`, `col-md-*`, `col-lg-*`, and `col-xl-*`. Donc, ce qui était `.col-md-6` en v3 est maintenant `.col-lg-6` en v4. Notez que l'infixe `-xs` a été supprimé et que `.col-6` représente 6 unités de colonne au point d'arrêt supplémentaire (par défaut).

Donc, si nous voulons maintenant écrire le même exemple dans Bootstrap 4, cela ressemblerait à ceci:

```
<div class="row">
  <div class="col-xl-8 col-lg-8 col-md-8 col-sm-8 col-8">
    contents
  </div>
  <div class="col-xl-8 col-lg-8 col-md-4 col-sm-4 col-4">
    contents
  </div>
</div>
```

## Mise en page de grille Bootstrap 4

	<b>Extra small</b> <576px	<b>Small</b> ≥576px	<b>Medium</b> ≥768px	<b>Large</b> ≥992px	<b>Extra large</b> ≥1200px
<b>Grid behavior</b>	Horizontal at all times	Collapsed to start, horizontal above breakpoints			
<b>Max container width</b>	None (auto)	540px	720px	960px	1140px
<b>Class prefix</b>	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
<b># of columns</b>	12				
<b>Gutter width</b>	30px (15px on each side of a column)				
<b>Nestable</b>	Yes				
<b>Offsets</b>	Yes				
<b>Column ordering</b>	Yes				

### Modification du support du navigateur

Dans twitter-bootstrap 4, le support pour `IE8`, `IE9`, and `iOS 6` a été supprimé. v4 est maintenant seulement `IE10+` and `iOS 7+`. Pour les sites qui en ont besoin, utilisez v3.

Dans twitter-bootstrap 4, le support officiel `Android v5.0 Lollipop's Browser and WebView` d' `Android v5.0 Lollipop's Browser and WebView` a été ajouté. Les versions antérieures du navigateur Android et de `WebView` restent uniquement `unofficially supported`.

### Suppression de classe

`Affix` est retiré de Bootstrap 4.

Il est recommandé d'utiliser une `position: sticky` `polyfill` `position: sticky` place.

Si vous utilisiez `Affix` pour appliquer des styles supplémentaires, les `polyfills` pourraient ne pas prendre en charge votre cas d'utilisation. Une option pour de telles utilisations est la bibliothèque `ScrollPos-Styler` tierce.

[Selon la documentation Bootstrap](#)

Suppression du plug-in `Affix` jQuery. Nous vous recommandons d'utiliser une `position:`

polyfill collant à la place. Voir le HTML5 S'il vous plaît entrer pour plus de détails et des recommandations spécifiques polyfill.

Si vous utilisiez Affix pour appliquer des styles supplémentaires, les polyfills pourraient ne pas prendre en charge votre cas d'utilisation. Une option pour de telles utilisations est la bibliothèque ScrollPos-Styler tierce.

Si quelqu'un passe de Bootstrap v3 à Bootstrap v4 l'approche de secours est donnée ci-dessous--

## HTML

```
<header>

</header>
<nav class="navbar navbar-light bg-faded" data-toggle="affix">
  <button class="navbar-toggler hidden-sm-up pull-xs-right" type="button" data-
toggle="collapse" data-target="#collapsingNavbar">
    ≡
  </button>
  <a class="navbar-brand" href="#">Brand</a>
  <div class="collapse navbar-toggleable-xs" id="collapsingNavbar">

    <ul class="nav navbar-nav pull-xs-right">
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#"
role="button" aria-haspopup="true" aria-expanded="false">
          Menu
        </a>
        <div class="dropdown-menu" aria-labelledby="Preview">
          <a class="dropdown-item" href="">Logout</a>

        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item ">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item ">
        <a class="nav-link" href="#">Link</a>
      </li>
    </ul>
  </div>
</nav>

<div class="container" id="main">
  <h2>Hello Bootstrap 4.</h2>
  <div class="row">
    <div class="col-xs-12 col-sm-6 col-md-9">
      <p>3 wolf moon retro jean shorts chambray sustainable roof party. Shoreditch vegan
artisan Helvetica. Tattooed Codeply Echo Park Godard kogi, next level irony ennui twee squid
fap selvage. Meggings flannel Brooklyn literally small batch, mumblecore
PBR try-hard kale chips. Brooklyn vinyl lumbersexual bicycle rights, viral fap
cronut leggings squid chillwave pickled gentrify mustache. 3 wolf moon hashtag church-key Odd
Future. Austin messenger bag normcore, Helvetica Williamsburg
sartorial tote bag distillery Portland before they sold out gastropub
taxidermy Vice.</p>
    </div>
```

```

    <div class="col-xs-6 col-md-3">
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra codeply
varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt codeply condimentum
vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi.
      Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan.
Aliquam in felis sit amet augue.
      </p>
      <hr>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis pharetra codeply
varius quam sit amet vulputate. Quisque mauris augue, molestie tincidunt codeply condimentum
vitae, gravida a libero. Aenean sit amet felis dolor, in sagittis nisi.
      Sed ac orci quis tortor imperdiet venenatis. Duis elementum auctor accumsan.
Aliquam in felis sit amet augue.
      </p>
    </div>
  </div>
  <div class="row">
    <div class="col-xs-6 col-sm-4">
      <div class="card card-outline-primary">
        <div class="card-block">
          <h3 class="card-title">Card</h3>
          <p class="card-text">With supporting text below as a natural lead-in to
additional content.</p>
          <a href="#" class="btn btn-outline-secondary">Outline</a>
        </div>
      </div>
    </div>
    <div class="col-xs-6 col-sm-4">
      <div class="card card-outline-primary">
        <div class="card-block">
          <h3 class="card-title">Card</h3>
          <p class="card-text">With supporting text below as a natural lead-in to
additional content.</p>
          <a href="#" class="btn btn-outline-secondary">Outline</a>
        </div>
      </div>
    </div>
    <div class="col-xs-6 col-sm-4">
      <div class="card card-outline-primary">
        <div class="card-block">
          <h3 class="card-title">Card</h3>
          <p class="card-text">With supporting text below as a natural lead-in to
additional content.</p>
          <a href="#" class="btn btn-outline-secondary">Outline</a>
        </div>
      </div>
    </div>
  </div>
</div>

```

## CSS

```

header {
  height: 220px;
  background: #ccc;
}

```

## JAVASCRIPT

```

$(document).ready(function() {

    var toggleAffix = function(affixElement, scrollElement, wrapper) {

        var height = affixElement.outerHeight(),
            top = wrapper.offset().top;

        if (scrollElement.scrollTop() >= top){
            wrapper.height(height);
            affixElement.addClass("affix");
        }
        else {
            affixElement.removeClass("affix");
            wrapper.height('auto');
        }

    };

    $('[data-toggle="affix"]').each(function() {
        var ele = $(this),
            wrapper = $('<div></div>');

        ele.before(wrapper);
        $(window).on('scroll resize', function() {
            toggleAffix(ele, $(this), wrapper);
        });

        // init
        toggleAffix(ele, $(window), wrapper);
    });

});

```

## Bootstrap 4 Navbar

Le nouveau composant Navar Bootstrap 4 est amélioré par rapport à son prédécesseur Bootstrap 3.x. Dans Bootstrap 4, la Navbar est **réactive par défaut** et utilise la **flexbox** pour faciliter l'alignement du contenu **Navbar** . Il `navbar-toggableable-*` également d'utiliser les nouvelles `navbar-toggableable-*` pour modifier le point d'arrêt Navbar. Maintenant, la barre de navigation a 6 tailles ou «états» de points d'arrêt, de sorte que vous pouvez facilement avoir l'une des options de barre de navigation suivantes.

- La barre de navigation *ne se réduit jamais* dans la vue mobile verticale et est toujours horizontale.
- La barre de navigation est *toujours* repliée dans la vue verticale et basculée via le hamburger.
- La barre de navigation se réduit en vue verticale à l' *un des 4 points d'arrêt sensibles* .

## Bootstrap de base 4 Navbar

```

<nav class="navbar navbar-toggableable-md">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
  data-target="#navbar1">
    <span class="navbar-toggler-icon"></span>

```

```

</button>
<a class="navbar-brand" href="#">Navbar</a>
<div class="collapse navbar-collapse" id="navbar1">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item active">
      <a class="nav-link" href="#">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
  </ul>
</div>
</nav>

```

Comme vous pouvez le voir à partir du code ci-dessus, la classe d' `navbar-header` a été supprimée de Bootstrap 4 et le `container-fluid` n'est plus requis pour une barre de navigation pleine largeur.

## Changer le point d'arrêt de la barre de navigation

La `navbar-togglerable-md` fait que la `navbar-togglerable-md` navigation ci-dessus s'effondre verticalement (et affiche l'icône du bascule) au point d'arrêt moyen (md) de 992px. Pour changer cela à un point d'arrêt différent, il suffit de remplacer `navbar-togglerable-md` par l'un de ceux-ci.

- `navbar-togglerable` = réduire sur des largeurs `xs` <576px
- `navbar-togglerable-sm` = réduire sur des largeurs de `sm` <768px
- `navbar-togglerable-lg` = réduire sur des largeurs de `lg` <1200px

## Démo Bootstrap 4 Breakpoint Navbar

### Modification de l'alignement de la barre de navigation

Flexbox nous permet de modifier facilement l'alignement de la barre de navigation et de son contenu (marque, liens, formulaires ou texte). Le contenu de la barre de navigation par défaut est aligné. Bien sûr, il existe de nombreux autres scénarios d'alignement ...

- Marque à gauche (par défaut), centre des liens et à droite
- Centre de marques, liens gauche et droit
- Marque à gauche et liens à droite
- Marque, liens et entrée de formulaire de largeur de remplissage
- Aucune marque, centre de liens et droite
- Marque à gauche, liens à l'intérieur du conteneur
- Liens justifiés (largeur de remplissage) centrés

### Bootstrap 4 Navbar avec marque centrée et liens gauche / droite

```

<nav class="navbar navbar-togglerable-sm navbar-inverse bg-primary">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
  data-target=".dual-collapse">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="navbar-collapse collapse dual-collapse">
    <ul class="navbar-nav mr-auto">

```



```

    <li class="nav-item active">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#features">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
  </ul>
</div>
<a class="navbar-brand d-flex mx-auto" href="#">Navbar 2</a>
<div class="navbar-collapse collapse dual-collapse">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#">Link</a>
    </li>
  </ul>
</div>
</nav>

```

## Bootstrap 4 Navbar avec Brand Left, Links Center et Right

```

<nav class="navbar navbar-light navbar-togglerable-sm bg-faded justify-content-center">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
  data-target="#collapsingNavbar3">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a href="/" class="navbar-brand d-flex w-50 mr-auto">Brand</a>
  <div class="navbar-collapse collapse" id="collapsingNavbar3">
    <ul class="navbar-nav mx-auto w-100 justify-content-center">
      <li class="nav-item active">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
    </ul>
    <ul class="nav navbar-nav ml-auto w-100 justify-content-end">
      <li class="nav-item">
        <a class="nav-link" href="#">Right</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Right</a>
      </li>
    </ul>
  </div>
</nav>

```

## Plus sur la barre de navigation Bootstrap 4

[Personnalisation de la couleur, de l'alignement ou de la hauteur](#)

## Bootstrap 3 à Bootstrap 4 Changements CSS

Comme Bootstrap 4 est une réécriture **majeure**, de nombreux noms de classes Bootstrap 3.x ont été modifiés ou supprimés. La restructuration de composants tels que la Navbar et l'introduction de nouvelles classes CSS et la prise en charge de Flexbox signifient que la mise à niveau vers 4.x n'est *pas* un processus de conversion simple à partir de la version 3.x.

Cependant, il existe des classes CSS Bootstrap 3.x qui ont un remplacement Bootstrap 4 spécifique.

### Le nom / sélecteur de classe CSS passe de Bootstrap 3.3.7 à 4 (alpha 6)

{t} - représente un **niveau** ou un point d'arrêt (par exemple: sm, md, lg, etc.).  
xs tier est la valeur par défaut et ne doit pas être spécifiée: `col-3`, `col-6`, etc.

{u} - représente une taille d' **unité col** (ex: 1-12)

Bootstrap 3.x	Bootstrap 4
.col- {t} - {u}	.col- {t} - {u} ( leave {t} blank for xs )
.col- {t} -offset- {u}	.offset- {t} - {u} ( leave {t} blank for xs )
.col- {t} -push- {u}	.push- {t} - {u} ( leave {t} blank for xs )
.col- {t} -pull- {u}	.pull- {t} - {u} ( leave {t} blank for xs )
.panneau	.carte
.panel-rubrique	En-tête de carte
titre .panel	Titre de la carte
.panel-body	bloc-carte
.panel-footer	.card-footer
.panel-primaire	.card-primary.card-inverse
.panel-succès	.card-success.card-inverse
.panel-info	.card-info.card-inverse
.panel-warning	.card-warning.card-inverse

Bootstrap 3.x	Bootstrap 4
.panel-danger	.card-danger.card-inverse
.bien	.card.card-block
.la vignette	.card.card-block
.list-inline> li	.list-inline-item
.dropdown-menu> li	.dropdown-item
.nav navbar> li	.nav-item
.nav navbar> li> a	.nav-link
.navbar-right	.ml-auto
.navbar-btn	.nav-item
.navbar-fixe-top	.fixed-top
.nav empilé	.flex-colonne
.btn-default	.btn-secondaire
.img-responsive	.img-fluide
.img-circle	cercle circulaire
.img-arrondi	.rounded
.form-horizontal	(enlevé)
.radio	.form-check
.checkbox	.form-check
.input-lg	.form-control-lg
.input-sm	.form-control-sm
.control-label	Etiquette de contrôle .form
.table-condensé	.table-sm
.pagination> li	.page-item
.pagination> li> un	.page-link
.article	.carousel-item

Bootstrap 3.x	Bootstrap 4
.text-help	.form-control-feedback
.pull-droit	.Flotter à droite
.pull-left	.float-left
.center-bloc	.mx-auto
.collapse.in	.collapse.show
.hidden-sm	.hidden-md-down
.hidden-md	.hidden-lg-down
.hidden-xs	.hidden-xs-down
.visible-xs	.hidden-sm-up
.visible-sm	.hidden-xs-down.hidden-md-up
.visible-md	.hidden-sm-down.hidden-lg-up
.visible-lg	.hidden-md-down.hidden-xl-up
.étiquette	.badge
.badge	.badge.badge-pill

### Regarde aussi:

[Outil de migration Bootstrap 3.x vers 4](#)

[Quoi de neuf dans Bootstrap 4](#)

## Bootstrap 4 Alignement vertical

Éléments Obtenir **centre** ou Aligner en **bas** a verticalement toujours été un défi avec CSS et Bootstrap. L'alignement vertical souhaité peut se situer dans un conteneur parent ou par rapport à des éléments adjacents.

Maintenant que Bootstrap 4 est **flexbox par défaut**, il existe de nombreuses approches de l'alignement vertical utilisant les [marges automatiques](#) , les [utilitaires Flexbox](#) ou les [utilitaires d'affichage](#) ainsi que les [utilitaires d'alignement vertical](#) .

Au début, les [utilitaires d'alignement vertical](#) sembleraient un choix évident, mais ils *ne* fonctionnent qu'avec des éléments d'affichage en ligne et de tableau. Voici quelques options et scénarios d'alignement vertical Bootstrap 4 ...

### 1 - Centre vertical utilisant les marges automatiques:

L'un des moyens de centrer verticalement consiste à utiliser `my-auto`. Cela permettra de centrer l'élément dans son conteneur. Par exemple, `h-100` fait que la ligne est à sa hauteur et `my-auto` centre verticalement la colonne `col-sm-12`.

```
<div class="row h-100">
  <div class="col-sm-12 my-auto">
    <div class="card card-block w-25">Card</div>
  </div>
</div>
```

## Centre vertical utilisant la [démonstration des marges automatiques](#)

`my-auto` représente les marges sur l'axe des y vertical et est équivalent à :

```
margin-top: auto;
margin-bottom: auto;
```

## 2 - Centre vertical avec Flexbox:



Taller

With supporting text

Outline

Puisque Bootstrap 4 `.row` est maintenant `display:flex` vous pouvez simplement utiliser `align-self-center` sur n'importe quelle colonne pour le centrer verticalement ...

```
<div class="row">
  <div class="col-6 align-self-center">
    <div class="card card-block">
      Center
    </div>
  </div>
  <div class="col-6">
    <div class="card card-inverse card-danger">
      Taller
    </div>
  </div>
</div>
```

ou, utilisez `align-items-center` sur tout le `.row` pour `.row` verticalement tous les `col-*` de la ligne ...

```
<div class="row align-items-center">
  <div class="col-6">
```

```

        <div class="card card-block">
            Center
        </div>
    </div>
    <div class="col-6">
        <div class="card card-inverse card-danger">
            Taller
        </div>
    </div>
</div>

```

## Demo de colonnes verticales de hauteur différente

### 3 - Centre vertical utilisant des utilitaires d'affichage:

Bootstrap 4 a [utils d'affichage](#) qui peuvent être utilisés pour l' `display:table` , `display:table-cell` , `display:inline` , etc .. Ils peuvent être utilisés avec les [utils d'alignement vertical](#) pour aligner roues alignées, inline-block ou des éléments cellulaires de la table.

```

<div class="row h-50">
  <div class="col-sm-12 h-100 d-table">
    <div class="card card-block d-table-cell align-middle">
      I am centered vertically
    </div>
  </div>
</div>

```

## Centre vertical utilisant la [démonstration de l'affichage Utilis](#)

### Bootstrap 4 Centrage

Comment centrer un élément, une colonne ou un contenu dans une colonne fonctionne différemment dans Bootstrap 4.

## Centre horizontal

- `text-center` est toujours utilisé pour l' `display:inline` éléments en `display:inline`
- `mx-auto` remplace le `center-block` central par l' `display:block` central `display:block` éléments de `display:block`
- `offset-*` *OU* `mx-auto` peut être utilisé pour centrer les colonnes de la grille

`mx-auto` (marges auto-axe x) centrera l' `display:block` ou `display:flex` éléments `display:flex` d'une *largeur définie* ( `%` , `vw` , `px` , etc.). **Flexbox est utilisé par défaut** sur les colonnes de la grille, il existe donc également différentes méthodes de centrage flexbox.

**Centrer le texte ou les éléments en ligne:** `text-center`

```

<div class="container">
  <h1 class="text-center">i'm centered</h1>
</div>

```

```
        <div class="col text-center">i'm centered!</div>
    </div>
</div>
```

**display:block central display:block OU display:flex : mx-auto**

```
<div class="row">
  <div class="col-12">
    
  </div>
</div>
```

**Centrer les colonnes à l'aide de décalages:** `offset-*`

```
<div class="row">
  <div class="col-4 offset-4">
    <h6>I'm .col-4 centered (offset 4)
  </div>
</div>
```

Les colonnes peuvent *également* être centrées avec: `mx-auto`

```
<div class="row">
  <div class="col-4 mx-auto">
    <h6>I'm .col-4 centered</h6>
  </div>
</div>
```

## Démo Bootstrap 4 Centrage horizontal

---

# Centre vertical

Pour le centrage vertical dans Bootstrap 4 (axe des ordonnées), consultez la documentation sur: [Bootstrap 4 Vertical Align](#)

## Bootstrap 4 Column Order

Changer l'ordre (ou la position) était possible dans Bootstrap 3 en utilisant les classes `push` `pull`. Dans Bootstrap 4, les classes **push-pull** fonctionnent toujours et, en outre, l' **ordre des boîtes flexibles** peut être utilisé.

Dans Bootstrap 4, les classes **push-pull** sont maintenant `push-{viewport}-{units}` et `pull-{viewport}-{units}` et le `xs-` infix a été supprimé. Considérez cet exemple qui modifie l'ordre des colonnes à 1-3-2 disposition sur `xs` et `sm` :

```
<div class="row">
  <div class="col-3 col-md-6">
    1
  </div>
```

```
<div class="col-3 col-md-6 push-6 push-md-0">
  2
</div>
<div class="col-6 col-md-12 pull-3 pull-md-0">
  3
</div>
</div>
```

## Démo Bootstrap 4 Push Pull

---

Étant donné que la nouvelle version 4 est flexbox, **une autre option** consiste à utiliser les classes de l' **utilitaire Flexbox** pour modifier l'ordre des colonnes. Maintenant toute la largeur, 12 unité `col-*-12` colonnes peuvent être inversées en utilisant la commande **de FlexBox**.

```
<div class="row">
  <div class="col-md-12">
    Col 1
  </div>
  <div class="col-md-12 flex-first flex-md-unordered">
    Col 2
  </div>
</div>
```

## Démo de commande Flexbox

Lire Migration vers Bootstrap 4 en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/9090/migration-vers-bootstrap-4>



---

# Chapitre 26: Modals

## Remarques

Les modaux exigent que bootstrap.min.js fonctionne correctement.

Plus de détails peuvent être trouvés ici: <http://getbootstrap.com/javascript/#modals>

## Exemples

### HTML modal de base

Un modal est une fenêtre de dialogue qui peut être affichée sur la page en cours.

```
<!-- Clicking the button will open the modal window -->
<button type="button" class="btn btn-success btn-lg" data-toggle="modal" data-
target="#theModal">Open The Modal</button>

<!-- The Modal -->
<div id="theModal" class="modal fade" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal">&times;</button>
        <h4 class="modal-title">Text For The Modal Header</h4>
      </div>
      <div class="modal-body">
        <p>Text for The Modal Body.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
</div>
```

Lire Modals en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6320/modals>

# Chapitre 27: Navbar

## Exemples

### Base Navbar (fixée en haut de la page)

```
<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <!-- vvv Hamburger icon that gets shown when window reaches a certain scale vvv -->
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <!-- ^^^ Hamburger icon that gets shown when window reaches a certain scale ^^^ -->
      <a class="navbar-brand" href="#">WebSite Title</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a asp-controller="Home" asp-action="Contact">Contact</a></li>
      </ul>
    </div>
  </div>
</div>
```

### Sous-menu dans la barre de navigation

```
<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">WebSite Title</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="#">Home</a></li>
        <li><a href="#">About>About</a></li>
        <li><a asp-controller="Home" asp-action="Contact">Contact</a></li>
        <!-- vvv Create a submenu in the navbar vvv -->
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">Testing Stuff
          <b class="caret"></b></a>
```

```

        <ul class="dropdown-menu">
            <li><a href="#">SubItem</a></li>
            <li><a href="#">Something Sub-y</a></li>
        </ul>
    </li>
    <!-- ^^ Create a submenu in the navbar ^^ -->
</ul>
</div>
</div>
</div>

```

## Diviseur de barre de navigation

```

<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">WebSite Title</a>
    </div>
    <div class="navbar-collapse collapse">
      <ul class="nav navbar-nav">
        <li><a href="#">Home</a></li>
        <!-- vvv Create a divider in the nav vvv -->
        <li class="divider"></li>
        <!-- ^^ Create a divider in the nav ^^ -->
        <li><a href="#">About</a></li>
        <li><a asp-controller="Home" asp-action="Contact">Contact</a></li>
      </ul>
    </div>
  </div>
</div>

```

## Garder le lien de navigation actuel "actif"

```

// Add active class to active navigation link
$(document).ready(function () {
  $('ul.nav.navbar-nav').find('a[href="' + location.pathname + '"]')
    .closest('li').addClass('active');
});

```

## Changer le point d'arrêt Navbar (mobile vs normal)

max-width est le point d'arrêt

```

@media (max-width: 1200px) {
  .navbar-header {
    float: none;
  }
  .navbar-left, .navbar-right {

```

```

        float: none !important;
    }
    .navbar-toggle {
        display: block;
    }
    .navbar-collapse {
        border-top: 1px solid transparent;
        box-shadow: inset 0 1px 0 rgba(255,255,255,0.1);
    }
    .navbar-fixed-top {
        top: 0;
        border-width: 0 0 1px;
    }
    .navbar-collapse.collapse {
        display: none!important;
    }
    .navbar-nav {
        float: none!important;
        margin-top: 7.5px;
    }
    .navbar-nav>li {
        float: none;
    }
    .navbar-nav>li>a {
        padding-top: 10px;
        padding-bottom: 10px;
    }
    .collapse.in{
        display:block !important;
    }
    .navbar-nav .open .dropdown-menu {
        position: static;
        float: none;
        width: auto;
        margin-top: 0;
        background-color: transparent;
        border: 0;
        -webkit-box-shadow: none;
        box-shadow: none;
    }
}

```

## Fermez la barre de navigation réduite lorsque vous cliquez en dehors de la barre de navigation

```

jQuery('body').bind('click', function(e) {
    if(jQuery(e.target).closest('#navbar').length == 0) {
        // click happened outside of .navbar, so hide
        var opened = jQuery('.navbar-collapse').hasClass('collapse in');
        if ( opened === true ) {
            jQuery('#navbar2 .navbar-collapse').collapse('hide');
        }
    }
});

```

Lire Navbar en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/2267/navbar>

# Chapitre 28: Navs

## Exemples

### Navs Bootstrap

Les Navs disponibles dans Bootstrap ont un balisage partagé, en commençant par la classe de base `.nav`, ainsi que des états partagés. Permutuez les classes de modificateur pour basculer entre chaque style.

### Onglets

```
<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

### Pilules

```
<ul class="nav nav-pills">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

### Justifié

```
<ul class="nav nav-tabs nav-justified">
  ...
</ul>
<ul class="nav nav-pills nav-justified">
  ...
</ul>
```

### Avec les dropdowns

```
<ul class="nav nav-tabs">
  <li role="presentation" class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#" role="button" aria-
haspopup="true" aria-expanded="false">
      Dropdown <span class="caret"></span>
    </a>
    <ul class="dropdown-menu">
      ...
    </ul>
  </li>
</ul>
```

Lire Navs en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6505/navs>

# Chapitre 29: Onglets

## Exemples

### HTML de base

```
<ul class="nav nav-tabs" role="tablist">
  <li role="presentation">
    <a href="#id-of-content-1" role="tab" data-toggle="tab">Tab 1</a>
  </li>
  <li role="presentation">
    <a href="#id-of-content-2" role="tab" data-toggle="tab">Tab 2</a>
  </li>
  <li role="presentation">
    <a href="#id-of-content-3" role="tab" data-toggle="tab">Tab 3</a>
  </li>
</ul>

<div class="tab-content">
  <div role="tabpanel" id="id-of-content-1" class="tab-pane">Tab content 1</div>
  <div role="tabpanel" id="id-of-content-2" class="tab-pane">Tab content 2</div>
  <div role="tabpanel" id="id-of-content-3" class="tab-pane">Tab content 3</div>
</div>
```

Cela créera un jeu d'onglets avec 3 onglets et 3 divs de contenu associés.

### Onglets animés

Pour faire apparaître les onglets, ajoutez `.fade` à chaque `.tab-pane`. Le volet d'onglet actif doit également avoir la classe `.in` pour rendre le contenu initial visible.

```
<ul class="nav nav-tabs" role="tablist">
  <li role="presentation">
    <a href="#id-of-content-1" role="tab" data-toggle="tab">
      Tab 1
    </a>
  </li>
  <li role="presentation" class="active">
    <a href="#id-of-content-2" role="tab" data-toggle="tab">
      Tab 2
    </a>
  </li>
  <li role="presentation">
    <a href="#id-of-content-3" role="tab" data-toggle="tab">
      Tab 3
    </a>
  </li>
</ul>

<div class="tab-content">
  <div role="tabpanel" id="id-of-content-1" class="tab-pane fade">
    Tab content 1
  </div>
  <div role="tabpanel" id="id-of-content-2" class="tab-pane fade active in">
```

```
    Tab content 2
</div>
<div role="tabpanel" id="id-of-content-3" class="tab-pane fade">
    Tab content 3
</div>
</div>
```

Lire Onglets en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/5980/onglets>

---

# Chapitre 30: Pagination

## Introduction

Les liens de pagination indiquent qu'une série de contenus connexes existe sur plusieurs pages. Ils sont généralement utilisés lorsqu'une approche multi-pages de longues listes de contenu améliore les performances générales, telles que les résultats de recherche ou les boîtes de réception.

## Exemples

### Un exemple simple de pagination

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
</nav>
```

qui vous donne:



Lire **Pagination en ligne**: <https://riptutorial.com/fr/twitter-bootstrap/topic/10605/pagination>



---

# Chapitre 31: Panneaux

## Remarques

Le composant de panneau dans bootstrap est une zone (bordée) avec un remplissage autour de son contenu, et éventuellement des conteneurs de titre et de pied de page.

## Exemples

### Exemple de base

Par défaut, tout ce `.panel` fait `.panel` c'est appliquer des bordures et des remplissages de base pour contenir du contenu.

```
<div class="panel panel-default">
  <div class="panel-body">
    Basic panel example
  </div>
</div>
```

### Panneau avec cap

Ajoutez facilement un conteneur d'en-tête à votre panneau avec l'en `.panel-heading`. Vous pouvez également inclure n'importe quel `<h1>-<h6>` avec une classe `.panel-title` pour ajouter un titre pré-stylisé. Cependant, les tailles de police `<h1>-<h6>` sont remplacées par `.panel-heading`.

Pour une coloration correcte des liens, veillez à placer des liens dans les en-têtes au sein de `.panel-title`.

```
<div class="panel panel-default">
  <div class="panel-heading">Panel heading without title</div>
  <div class="panel-body">
    Panel content
  </div>
</div>

<div class="panel panel-default">
  <div class="panel-heading">
    <h3 class="panel-title">Panel title</h3>
  </div>
  <div class="panel-body">
    Panel content
  </div>
</div>
```

### Panneau avec pied de page

Enveloppez les boutons ou le texte secondaire dans `.panel-footer`. Notez que les pieds de

tableau de bord n'héritent **pas des** couleurs et des bordures lors de l'utilisation de variations contextuelles, car ils ne sont pas destinés à être au premier plan.

```
<div class="panel panel-default">
  <div class="panel-body">
    Panel content
  </div>
  <div class="panel-footer">Panel footer</div>
</div>
```

Lire Panneaux en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/2848/panneaux>

---

# Chapitre 32: Système de grille

## Introduction

Le système de grille de Bootstrap se compose de 12 unités appelées **colonnes** ( `.col-*` CSS) utilisées pour mettre en page le contenu de *gauche à droite* dans la fenêtre d'affichage. Les colonnes sont contenues dans les **lignes** (classe CSS `.row`) pour créer des groupes horizontaux de colonnes. Les lignes sont placées dans un **conteneur** fixe ou pleine largeur (respectivement `.container` ou `.container-fluid`) pour un alignement correct. Les colonnes ont un remplissage qui crée un espacement (appelé "gouttière") entre le contenu des colonnes.

## Remarques

Bootstrap comprend un premier système de grille fluide réactif et mobile qui permet de faire évoluer jusqu'à 12 colonnes en fonction de la taille du périphérique ou de la fenêtre d'affichage. Il comprend des classes prédéfinies pour créer rapidement des mises en page via une série de lignes et de colonnes contenant votre contenu.

## Exemples

### Requêtes médias

Les requêtes multimédias dans Bootstrap vous permettent de déplacer, d'afficher et de masquer le contenu en fonction de la taille de la fenêtre d'affichage. Les requêtes multimédia suivantes sont utilisées dans les fichiers LESS pour créer les points d'arrêt clés dans le système de grille Bootstrap:

```
/* Small devices (tablets, 768px and up) */
@media (min-width: @screen-sm-min) { ... }

/* Medium devices (desktops, 992px and up) */
@media (min-width: @screen-md-min) { ... }

/* Large devices (large desktops, 1200px and up) */
@media (min-width: @screen-lg-min) { ... }
```

Parfois, ils sont étendus pour inclure une largeur maximale afin de limiter les CSS à un ensemble plus restreint de périphériques:

```
@media (max-width: @screen-xs-max) { ... }
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }
@media (min-width: @screen-lg-min) { ... }
```

### Niveaux de grille Bootstrap (points d'arrêt)

Outre le concept d' **unités** de **colonne** , Bootstrap a différents **points d'arrêt** ou tailles de grille appelés niveaux. La grille Bootstrap 3 comporte quatre (4) niveaux pour accueillir différentes largeurs d'écran (ou de fenêtre). Les trois niveaux de Bootstrap sont `xs` , `sm` , `md` et `lg` . Les colonnes de grille de Bootstrap sont identifiées par différentes classes CSS de `col-{breakpoint}-{units}` .

Chaque niveau de grille **englobe une gamme** conçue pour s'adapter au mieux aux largeurs d'écran de périphérique standard telles que les ordinateurs de bureau, les ordinateurs portables, les tablettes et les smartphones.

Bootstrap utilise des requêtes média CSS pour créer des points d'arrêt réactifs qui établissent une limite pour chaque taille de grille. Ces tailles de grille vous permettent de modifier la disposition des colonnes pour correspondre au mieux aux différentes largeurs d'écran et aux différents périphériques\_\_, l'essence même du design réactif.

- `col-xs-*` - pour les *plus petites* largeurs d'écran comme les smartphones <768 px
- `col-sm-*` - pour les *petits* écrans comme les smartphones et les tablettes > = 768 px
- `col-md-*` - pour les *écrans de taille moyenne* comme les tablettes et les ordinateurs portables > = 992 px
- `col-lg-*` - pour les *grandes* largeurs d'écran telles que les ordinateurs de bureau > = 1200 px

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices (≥1200px)
<b>Grid behavior</b>	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
<b>Container width</b>	None (auto)	750px	970px	1170px
<b>Class prefix</b>	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
<b># of columns</b>	12			
<b>Column width</b>	Auto	~62px	~81px	~97px
<b>Gutter width</b>	30px (15px on each side of a column)			
<b>Nestable</b>	Yes			
<b>Offsets</b>	Yes			
<b>Column ordering</b>	Yes			

Référence: [Système de grille](#)

### Même largeur de colonne pour chaque appareil

Pour créer une colonne qui représente toujours 50% de la largeur de la fenêtre d'affichage (sur tous les périphériques), vous pouvez définir `col-*-6` pour chaque niveau.

```
<div class="col-xs-6 col-sm-6 col-md-6 col-lg-6">..</div>
```

Cependant, ceci est un balisage supplémentaire inutile, car `col-xs-6` signifie 6 unités sur `xs` et plus. Le plus petit niveau que vous définissez (`xs`, `sm` ou `md`) définit également la taille des plus grandes largeurs d'écran. Pour la *même* colonne de taille sur tous les niveaux, définissez simplement la largeur de la plus petite fenêtre.

*Code plus court:*

```
<div class="col-xs-6">..</div>
```

## Largeur de colonne différente pour chaque périphérique (design réactif)

Les classes `col-*-*` peuvent être **combinées** pour contrôler la largeur des colonnes sur différentes tailles de grille.

Par exemple, créez une colonne de largeur 50% au niveau `sm` et une colonne de largeur 25% au niveau `md` ...

```
<div class="col-md-3 col-sm-6">..</div>
```

Les grilles `sm`, `md` et `lg` seront toutes "empilées" verticalement à des largeurs de fenêtre inférieures à 768 pixels. C'est là que la grille `xs` s'insère. Les colonnes qui utilisent les classes `col-xs-*` ne s'empileront pas verticalement et continueront à réduire les plus petits écrans.

## Lignes et colonnes bootstrap

Le système de grille de Bootstrap comporte **12 unités** appelées **colonnes** pouvant être utilisées pour mettre en forme le contenu horizontalement dans la fenêtre d'affichage.

La raison d'une grille de 12 unités (au lieu de 10, 16, etc.) est que 12 se divise également en 6 (moitiés), 4 (quarts) et 3 (tiers). Cela facilite grandement l'adaptation à diverses configurations. Les colonnes de grille de Bootstrap sont identifiées par différentes classes CSS de `col-{breakpoint}-{units}`. [En savoir plus sur la largeur de la fenêtre d'affichage et les points d'arrêt \(niveaux AKA\)](#)

Ainsi, par exemple, `col-md-3` représente une colonne qui occupe 3 des 12 unités (ou 25%) sur une fenêtre de largeur moyenne (`md`). Pour utiliser une largeur de colonne dans votre mise en page, utilisez simplement la classe `col-{breakpoint}-{units}` appropriée dans votre balisage HTML.

```
<div class="col-{breakpoint}-{units}">
```

La largeur de la colonne est fluide (pas de largeur fixe), de sorte que les colonnes consomment un *pourcentage* de leur conteneur.

### Unités de colonne dans Bootstrap 3

- `col-*-1` : 1 sur 12 (largeur 8.33333333%)
- `col-*-2` : 2 sur 12 (largeur 16,66666667%)
- `col-*-3` : 3 sur 12 (largeur 25%)

- `col-*-4` : 4 of 12 (largeur 33.3333333%)
- `col-*-5` : 5 of 12 (largeur 41.66666667%)
- `col-*-6` : 6 of 12 (50% largeur)
- `col-*-7` : 7 of 12 (largeur 58.33333333%)
- `col-*-8` : 8 of 12 (66.66666667% largeur)
- `col-*-9` : 9 of 12 (largeur de 75%)
- `col-*-10` : 10 of 12 (largeur 83.33333333%)
- `col-*-11` : 11 de 12 (largeur 91.66666667%)
- `col-*-12` : 12 of 12 (100% largeur)

## Demo - Les 12 unités de colonne de Bootstrap

### La ligne bootstrap

La classe Bootstrap `.row` est utilisée pour contenir les colonnes. Les colonnes doivent *toujours* être placées dans des lignes et les lignes doivent toujours être placées à l'intérieur d'un conteneur (`container` ou `container-fluid`). La ligne utilise des marges négatives (-15px) pour garantir un espacement correct entre le contenu de la colonne et le bord du navigateur. Les lignes sont utilisées pour regrouper les colonnes horizontalement.

```
<div class="container">
  <div class="row">
    <!-- one more columns -->
    <div class="col-{breakpoint}-{units}">..</div>
  </div>
</div>
```

Les colonnes rempliront le `.row` horizontalement de gauche à droite et **envelopperont** une nouvelle ligne toutes les 12 unités de colonne. Par conséquent, vous pouvez utiliser `.row` s pour créer des **sauts horizontaux**, ou vous pouvez ajouter **plus de 12** unités de colonne dans un seul élément `.row` pour que les **colonnes** soient **enveloppées** (ou empilées) verticalement dans la fenêtre.

Lorsque vous utilisez le `.row` à la colonne (plus de 12 unités dans un `.row`), vous devrez **utiliser des réinitialisations réactives (ou des correctifs)** pour garantir un emballage uniforme du contenu irrégulier des colonnes. Ceci est essentiel lorsque le contenu des colonnes varie en hauteur.

### En savoir plus sur les colonnes et les lignes de la grille Bootstrap

[Bootstrap 3 problèmes de disposition de la grille fluide?](#)

[Bootstrap 3 - ligne imbriquée puis-je ajouter des colonnes à plus de 12?](#)

[Explication de la ligne et de la colonne Bootstrap](#)

[Comment fonctionne la grille Bootstrap \(moyenne\)](#)

### Conteneurs

Bootstrap nécessite un élément contenant pour envelopper le contenu du site et héberger notre système de grille. Vous pouvez choisir l'un des deux conteneurs à utiliser dans vos projets.

Utilisez la classe `.container` pour un conteneur à largeur fixe sensible.

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

Utilisez `.container-fluid` classe `.container-fluid` pour un conteneur pleine largeur couvrant toute la largeur de votre fenêtre.

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

Remarque: Les conteneurs ne peuvent pas être imbriqués (vous ne pouvez pas placer un conteneur dans un autre conteneur), en raison du `padding` et d'autres éléments.

## Colonnes de compensation

Ces classes augmentent la marge de gauche d'une colonne par \* colonnes. Par exemple, `.col-md-offset-4` déplace `.col-md-4` sur quatre colonnes.

```
<div class="row">
  <div class="col-lg-4"></div>
  <div class="col-lg-4 col-lg-offset-4"></div>
</div>
<div class="row">
  <div class="col-lg-5 col-lg-offset-1"></div>
  <div class="col-lg-5 col-lg-offset-1"></div>
</div>
```

## Manipulation de l'ordre des colonnes à l'aide de push and pull

```
<div class="container content">
  <div class="row">
    <!--Main Content-->
    <div class="col-lg-9 col-lg-push-3">
      Main Content
    </div>

    <!--Sidebar-->
    <div class="col-lg-3 col-lg-pull-9">
      Sidebar
    </div>
  </div>
</div>
```

Cela modifie l'ordre des colonnes de la grille intégrée.

Syntaxe: `.col-md-push-*` et `.col-md-pull-*`.

Plus:

[Manipulation de l'ordre des colonnes en utilisant col-lg-push et col-lg-pull dans Twitter Bootstrap 3](#)

Bootstrap 3: Colonnes Push / Pull uniquement sur des tailles d'écran plus petites  
Commande de colonne et empilement dans Bootstrap 3

Lire **Système de grille en ligne**: <https://riptutorial.com/fr/twitter-bootstrap/topic/3330/systeme-de-grille>



---

# Chapitre 33: Thèmes Bootstrap

## Exemples

### Thèmes d'amorçage et remplacements de règles

#### Quels sont les thèmes?

Il existe plusieurs aspects visuels pour Bootstrap, qui peuvent être trouvés à partir de sources telles que [Bootswatch](#), qui modifient le fichier `bootstrap.min.css`. Vous pouvez également créer votre propre thème de cette façon.

#### Quand modifier des thèmes et quand ajouter de nouvelles règles à un fichier `site.css`?

Quand modifier le

`bootstrap.min.css`

fichier, et quand ajouter votre propre fichier `.css`, tel que

`site.css`

?

Parfois, il y a des exigences de style, ce qui doit être fait, quel que soit le thème que vous utilisez. Ces règles devraient aller dans votre propre fichier `.css`, tel que `site.css`, afin que le thème principal puisse être modifié, les règles de `site.css` s'appliqueront de toute façon. Pour ce faire, il vous suffit de lier le thème bootstrap et vos propres règles pour remplacer celles existantes:

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

De cette façon, les thèmes Bootstrap peuvent être modifiés à tout moment sans perdre les règles obligatoires, appliquées à partir de `site.css`.

Lire Thèmes Bootstrap en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6381/themes-bootstrap>

# Chapitre 34: Twitter Personnalisation du style Bootstrap

## Remarques

Une chose à noter est qu'il faut mentionner le nom `custom.css` après le `bootstrap.css` principal, sinon les valeurs de `custom.css` ne seront pas réellement implémentées.

## Exemples

### Remplacement du CSS par défaut

Tout le monde aime le [bootstrap sur Twitter](#) , mais certains d'entre nous n'aiment pas sa conception par défaut. Donc, voici un guide simple sur la façon de commencer à personnaliser la conception de bootstrap. Twitter bootstrap lorsque cloné fournit un ensemble de fichiers CSS par défaut que nous pouvons remplacer.

Le fichier css de courrier que nous devons remplacer est le `bootstrap.min.css` sous le `bootstrap/dist/css` .

Pour remplacer la conception par défaut de bootstrap, suivez ces deux étapes simples.

1. `custom.css` un `custom.css` (ou nommez-le comme vous voulez) et liez-le à votre `index.html`

```
<html>
<head>
  <title>Customize Bootstrap</title>

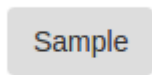
  <link rel="stylesheet" type="text/css" href="path/to/bootstrap.min.css">
  <!-- This mus be declared after the bootstrap.min.css -->
  <link rel="stylesheet" type="text/css" href="path/to/your/custom.css">
</head>
<body>
  <!-- Do something -->
</body>
</html>
```

2. Commencez à personnaliser. Par exemple, nous voulons changer la couleur du bouton par défaut. Si vous voulez utiliser le style de bouton par défaut de bootstrap, vous devez ajouter la classe `btn` sur la `<button class="btn">Sample</button>` . Il suffit d'écrire le code suivant sur votre `custom.css` .

```
.btn{
  background-color:red;
}
```

Le code ci-dessus produira quelque chose comme ça.

**Défaut :**



**Douane :**



Cette technique nous évitera de réécrire l'intégralité des styles de boutons déjà écrits par les contributeurs bootstrap. Cela nous a également évité d'écrire notre propre classe CSS qui est pour moi moins fastidieuse.

Lire Twitter Personnalisation du style Bootstrap en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6030/twitter-personnalisation-du-style-bootstrap>

---

# Chapitre 35: Utiliser Clearfix dans les lignes et les colonnes

## Introduction

Lors de la création de mises en page avancées, il se peut que vous deviez utiliser **plus de 12 unités de colonne** dans un seul élément `.row`. Le concept d' **enroulement de colonne** et de **réinitialisations réactives (AKA clearfixes)** est *essentiel* pour comprendre la conception réactive avec Bootstrap.

[Les bases de la grille bootstrap](#)

## Remarques

Les grilles Bootstraps sont remarquablement puissantes et élégantes. Cependant, vous devez vous rappeler que le nom du framework est "Bootstrap", pas "WeDidItForYou". Bootstrap **permet** une conception réactive, elle ne le **garantit** pas.

Il vous appartient encore de rendre votre conception réellement réactive et de donner à vos utilisateurs la meilleure expérience possible.

## Exemples

### La première tentative naïve

Avant de commencer, définissons quelques CSS pour les exemples. Ceci est la `head` section de notre échantillon. J'utilise toujours `border-radius` et `background-color` lorsque je teste, car cela simplifie la visualisation des divisions sans ajouter de taille de bordure susceptible d'affecter la taille des cellules.

```
<head>
  <title></title>
  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <style>
    /* colorize all col- */
    [class^="col-"] {
      min-height: 30px;
      border-radius: 10px;
      background-color: lightblue;
    }
    /* a tall cell */
    .cell-tall {
      height: 100px;
      background-color: orange;
    }
    /* a medium-height cell */
```

```

.cell-med {
  height: 50px;
  background-color: lightgreen;
}
/* padding top-bottom for some row examples */
.row.padded {
  padding: 1rem 0 1rem 0;
}
</style>
</head>

```

Avec cela, définissons une grille et regardons les résultats parfaits à toutes les tailles de fenêtre!

Utiliser `col-xs-6 col-md-3`

```

<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">1</div>
    <div class="col-xs-6 col-md-3">2</div>
    <div class="col-xs-6 col-md-3">3</div>
    <div class="col-xs-6 col-md-3">4</div>
    <div class="col-xs-6 col-md-3">5</div>
    <div class="col-xs-6 col-md-3">6</div>
    <div class="col-xs-6 col-md-3">7</div>
    <div class="col-xs-6 col-md-3">8</div>
    <div class="col-xs-6 col-md-3">9</div>
    <div class="col-xs-6 col-md-3">10</div>
    <div class="col-xs-6 col-md-3">11</div>
  </div>
</div>

```

`col-xs-6 col-md-3`



`col-xs-6 col-md-3`



Les deux images précédentes montrent le rendu à des tailles d'écran moyennes et petites.

Rappelez-vous que nous obtiendrons QUATRE colonnes sur medium + en raison de `col-md-3` et de TWO cellules en raison de `col-xs-6`.

Ça a l'air pas mal, non? Je pense que nous avons fini ici! Dit beaucoup de sites naïfs Bootstrap là-bas juste là pour attendre ...

## Le problème de la hauteur

Dans notre " [exemple naïf](#) ", toutes nos cellules avaient la même hauteur. Le navigateur a volontairement rompu les lignes exactement là où nous le voulions, et tout semblait convenir au monde entier. Jusqu'à ce que la hauteur entre en jeu.

Prenons l'exemple précédent et donnons une certaine hauteur à certaines cellules, peut-être comme vous le verrez sur une page de type tableau de bord.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">1</div>
    <div class="col-xs-6 col-md-3 cell-tall">2</div>
    <div class="col-xs-6 col-md-3 cell-tall">3</div>
    <div class="col-xs-6 col-md-3">4</div>
    <div class="col-xs-6 col-md-3">5</div>
    <div class="col-xs-6 col-md-3 cell-med">6</div>
    <div class="col-xs-6 col-md-3">7</div>
    <div class="col-xs-6 col-md-3">8</div>
    <div class="col-xs-6 col-md-3 cell-med">9</div>
    <div class="col-xs-6 col-md-3 cell-med">10</div>
    <div class="col-xs-6 col-md-3">11</div>
  </div>
</div>
```

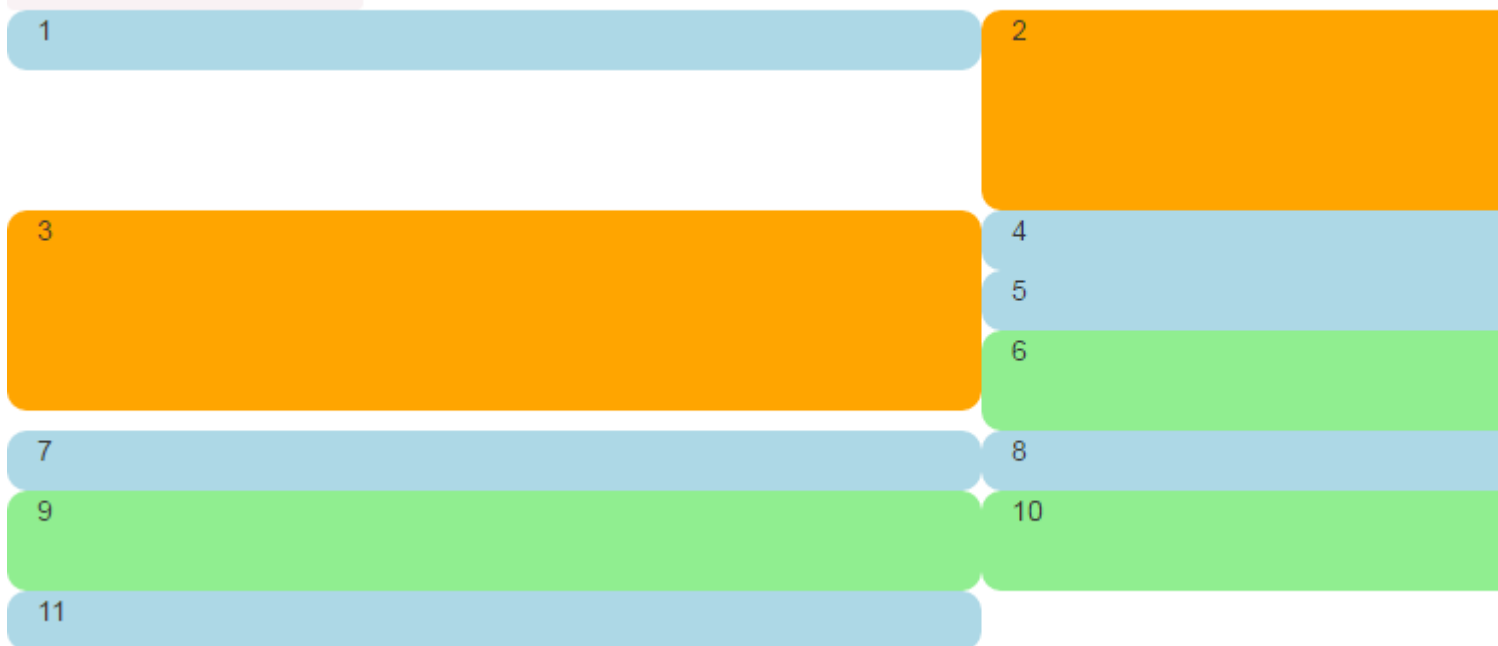
Ici, nous avons ajouté des CSS `cell-tall` et `cell-med` que nous avons définis ci-dessus. Cela aura pour effet de modifier la hauteur de certaines cellules. Je me demande comment ça va se passer ...

Les voici à nouveau à des tailles d'écran moyennes et petites:

`col-xs-6 col-md-3`



col-xs-6 col-md-3



Oh mon dieu, quel bordel. Je ne pense pas que ce soit ce que nous voulions. À taille moyenne-grande, 5 et 6 sont hors de propos et, en quelque sorte, ont fini par commencer une nouvelle ligne. En petite taille, nous avons deux cellules dans la première ligne et **quatre** dans la deuxième ligne, avec 4, 5 et 6 sur la droite pour les deux tailles d'écran!

Alors, comment pouvons-nous résoudre ce problème?

## Clearfix à la rescousse

Une façon d'aider la situation serait certainement d'utiliser plusieurs `row` :

```
<div class="container-fluid">
  <div class="row">
    <!-- cols -->
  </div>
  <div class="row">
    <!-- cols -->
  </div>
</div>
```

C'est généralement la première chose que les nouveaux Bootstrappers essaient. Il semble logique: « Je veux que quatre cellules dans chaque ligne, donc je vais juste créer une nouvelle `row` pour chaque 4 `col` divs ».

Mais ce raisonnement pose problème: Le but de Bootstrap 3 et de la prochaine version 4 est d'être **réactif** . En plaçant "quatre `col` d' `row` ", vous ne réfléchissez pas vraiment.

Une bonne compréhension de la classe CSS `clearfix` vous aidera à commencer à voir que les div de plusieurs `row` ont réellement entaché votre compréhension de la façon dont la conception **adaptative** était **censée** fonctionner. En bref, vous **ne pouvez** tout simplement **pas** savoir

combien de `col` à mettre en `row` toute façon - le navigateur n'a pas encore rendu votre travail!

Rappelez-vous dans *First Things First*, nous avons dit que vous devez penser en "inverse of 12"? Sans plus tarder, corrigeons notre problème ici, en utilisant les commentaires directement dans le code pour, espérons-le, éliminer toute confusion. Oui, cela ressemble beaucoup plus au code, mais la **plupart** des commentaires sont des commentaires.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">1</div>
    <div class="col-xs-6 col-md-3 cell-tall">2</div>
    <!--
      We have rendered TWO cells.
      On small and extra small devices, the viewport will render TWO cells
      (12 / 6 = 2), so we need a clearfix every TWO cells. We also need to
      say "don't show this clearfix when the viewport will render FOUR cells",
      which it will do at medium size and up (12 / 3 = 4). We do that by adding
      hidden-md and hidden-lg to our clearfix div, in effect instructing the
      browser to not show it at all on a wider screen.
    -->
    <div class="clearfix hidden-md hidden-lg"></div>
    <!-->
    <div class="col-xs-6 col-md-3 cell-tall">3</div>
    <div class="col-xs-6 col-md-3">4</div>
    <!--
      We have now rendered FOUR cells.
      We are never going to have more than FOUR cells side by side. So every
      FOURTH cell, we place a clearfix that will ALWAYS show. We do this by
      just leaving off any of the hidden-* classes.
    -->
    <div class="clearfix"></div>
    <!-->
    <div class="col-xs-6 col-md-3">5</div>
    <div class="col-xs-6 col-md-3 cell-med">6</div>
    <!--
      We have now rendered SIX cells.
      After the sixth cell, we are at a multiple of TWO, but not FOUR so we
      repeat the clearfix that we used after cell TWO.
    -->
    <div class="clearfix hidden-md hidden-lg"></div>
    <!-->
    <div class="col-xs-6 col-md-3">7</div>
    <div class="col-xs-6 col-md-3">8</div>
    <!--
      Now we have rendered EIGHT cells, which is a multiple of TWO AND FOUR,
      so we put in a clearfix that's always visible.
    -->
    <div class="clearfix"></div>
    <!-->
    <div class="col-xs-6 col-md-3 cell-med">9</div>
    <div class="col-xs-6 col-md-3 cell-med">10</div>
    <!--
      After the 10th cell, once again a multiple of TWO but not FOUR...
    -->
    <div class="clearfix hidden-md hidden-lg"></div>
    <!-->
    <div class="col-xs-6 col-md-3">11</div>
  </div>
</div>
```



Le `clearfix` est une classe CSS qui rend un minuscule (presque invisible) div, et son but est de « clairs » les `left` flotteurs qui ont été utilisés par les `col` divs.

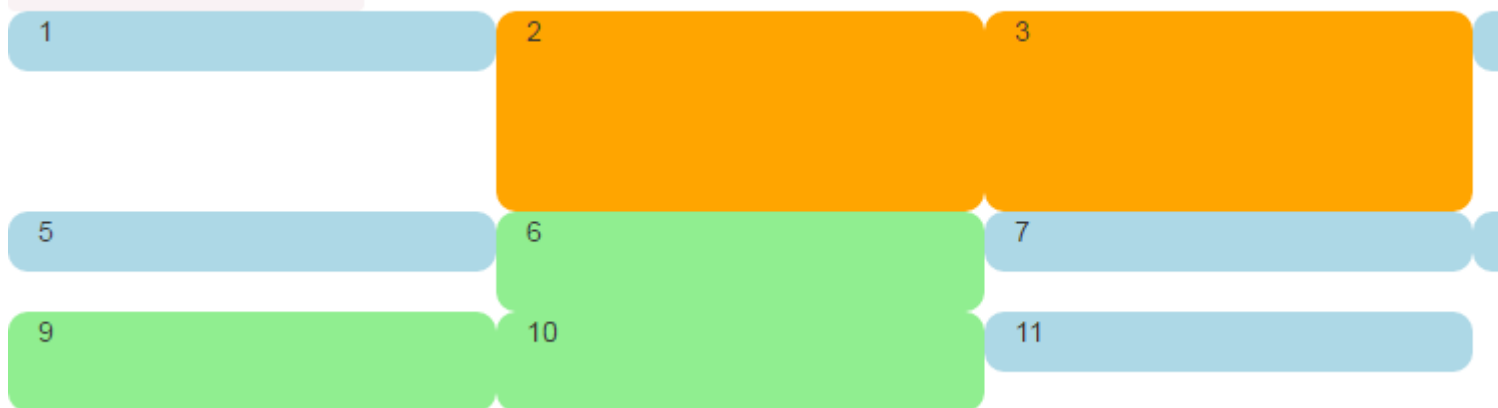
Le génie est vraiment dans les classes `hidden-sm`, `hidden-md`, etc. Ces classes sont placées **sur le `clearfix` div**, pas sur le `col` divs! Cela fait apparaître ou disparaître comme par magie le `clearfix` du flux de rendu à certaines largeurs de fenêtre d'affichage! Génie!

Bootstrap a un tableau déconcertant de `hidden-*` et `visible-*` dans la version 3 et, malheureusement, elles ne sont pas vraiment "l'inverse" les unes des autres. Je trouve donc le plus clair et le plus sûr d'utiliser juste toujours les `hidden-*` cours sur les clearfixes.

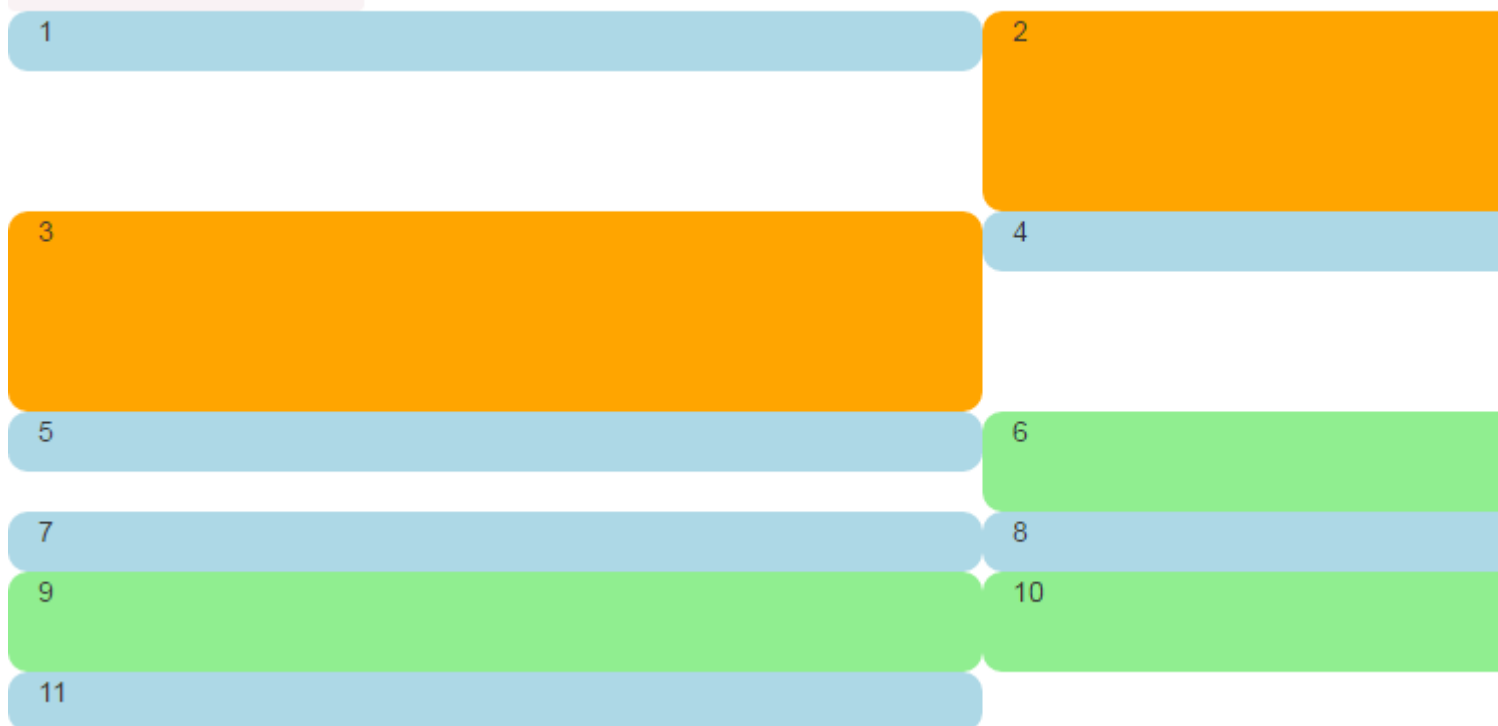
Cela semble changer pour le mieux dans Bootstrap 4, avec des classes telles que `hidden-*-up` et `hidden-*-down` (elles se débarrassent `visible-*` classes `visible-*` ).

Assez de verbiage, à quoi ça ressemble maintenant?

`col-xs-6 col-md-3`



`col-xs-6 col-md-3`



C'est ce que nous voulons! Dans le grand écran, nous avons toujours QUATRE de travers, dans le petit écran, toujours DEUX à travers. Pas plus d'empiler dans des endroits étranges, et les lacunes sont là où nous nous attendions à ce qu'ils soient.

## Un tableau de bord

Assez de ces choses arrondies colorées, mettons quelque chose de plus intéressant que les nombres dans ces div. Prenons le même ensemble de colonnes et créons un tableau de bord réel. Utilisez le CSS suivant:

```
<head>
  <title></title>
  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <style>
    body {
      padding-top: 15px;
    }
    .panel-tall .panel-body {
      height: 175px;
    }
    .panel-med .panel-body {
      height: 100px;
    }
    .panel-short .panel-body {
      height: 70px;
    }
  </style>
</head>
```

Et voici le code "tableau de bord":

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3">
      <div class="panel panel-default panel-med">
        <div class="panel-heading">
          Heading 1
        </div>
        <div class="panel-body">
          Body 1
        </div>
        <div class="panel-footer">
          Footer 1
        </div>
      </div>
    </div>
    <div class="col-xs-6 col-md-3 cell-tall">
      <div class="panel panel-danger panel-tall">
        <div class="panel-heading">
          Heading 2
        </div>
        <div class="panel-body">
          Body 2. Look out, this needs some attention!
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <div class="panel-footer">
            Footer 2
        </div>
    </div>
</div>
<!--
    On small and extra small devices, the viewport will render TWO cells
    (12 / 6 = 2), so we need a clearfix every TWO cells. We also need to
    say "don't show this clearfix when the viewport will render FOUR cells",
    which it will do at medium size and up (12 / 3 = 4). We do that by adding
    hidden-md and hidden-lg to our clearfix div, in effect instructing the
    browser to not show it at all on a wider screen.
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!-->
<div class="col-xs-6 col-md-3 cell-tall">
    <div class="panel panel-success panel-short">
        <div class="panel-heading">
            Heading 3
        </div>
        <div class="panel-body">
            Body 3. The file has successfully uploaded.
        </div>
        <div class="panel-footer">
            Footer 3
        </div>
    </div>
</div>
<div class="col-xs-6 col-md-3">
    <div class="panel panel-default panel-tall">
        <div class="panel-heading">
            Heading 4 Chart
        </div>
        <div class="panel-body">
            Body 4. Is this a cool graph or what?
        </div>
        <div class="panel-footer">
            Footer 4
        </div>
    </div>
</div>
</div>
<!--
    We are never going to have more than FOUR cells. So every FOURTH cell,
    we place a clearfix that will ALWAYS show. We do this by just leaving off
    any of the hidden-* classes.
-->
<div class="clearfix"></div>
<!-->
<div class="col-xs-6 col-md-3">
    <div class="panel panel-warning panel-short">
        <div class="panel-heading">
            Heading 5
        </div>
        <div class="panel-body">
            Body 5.
        </div>
        <div class="panel-footer">
            Footer 5
        </div>
    </div>
</div>
</div>

```

```

<div class="col-xs-6 col-md-3 cell-med">
  <div class="panel panel-warning panel-tall">
    <div class="panel-heading">
      Heading 6
    </div>
    <div class="panel-body">
      Body 6.
    </div>
  </div>
</div>
<!--
  After the sixth cell, we are at a multiple of TWO, but not FOUR so we
  repeat the clearfix that we used after cell TWO.
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!------>
<div class="col-xs-6 col-md-3">
  <div class="panel panel-info panel-tall">
    <div class="panel-heading">
      Heading 7
    </div>
    <div class="panel-body">
      Body 7.
    </div>
    <div class="panel-footer">
      Footer 7
    </div>
  </div>
</div>
<div class="col-xs-6 col-md-3">
  <div class="panel panel-info panel-med">
    <div class="panel-heading">
      Heading 8
    </div>
    <div class="panel-body">
      Body 8.
    </div>
    <div class="panel-footer">
      Footer 8
    </div>
  </div>
</div>
<!--
  Now we have rendered EIGHT cells, which is a multiple of TWO AND FOUR,
  so we put in a clearfix that's always visible.
-->
<div class="clearfix"></div>
<!------>
<div class="col-xs-6 col-md-3 cell-med">
  <div class="panel panel-info panel-short">
    <div class="panel-heading">
      Heading 9
    </div>
    <div class="panel-body">
      Body 9.
    </div>
    <div class="panel-footer">
      Footer 9
    </div>
  </div>
</div>

```

```

<div class="col-xs-6 col-md-3 cell-med">
  <div class="panel panel-info panel-tall">
    <div class="panel-heading">
      Heading 10
    </div>
    <div class="panel-body">
      Body 10.
    </div>
    <div class="panel-footer">
      Footer 10
    </div>
  </div>
</div>
<!--
  After the 10th cell, once again a multiple of TWO but not FOUR...
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!------>
<div class="col-xs-6 col-md-3">
  <div class="panel panel-info panel-tall">
    <div class="panel-heading">
      Heading 11
    </div>
    <div class="panel-body">
      Body 11.
    </div>
    <div class="panel-footer">
      Footer 11
    </div>
  </div>
</div>
</div>
</div>
</div>

```

Ce code ressemblera à ceci:

Heading 1
Body 1
Footer 1

Heading 2
Body 2. Look out, this needs some attention!
Footer 2

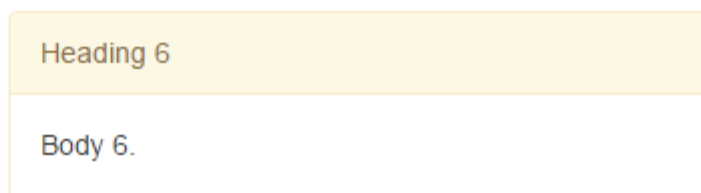
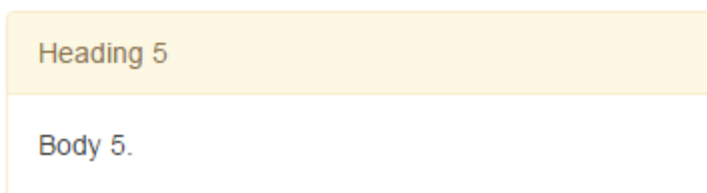
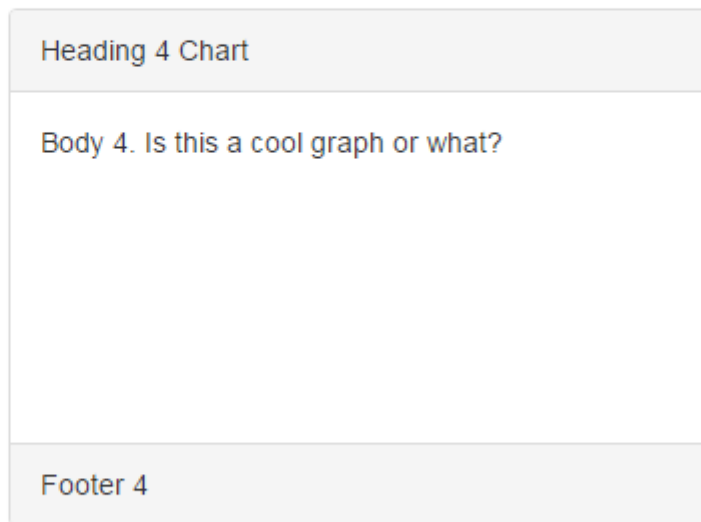
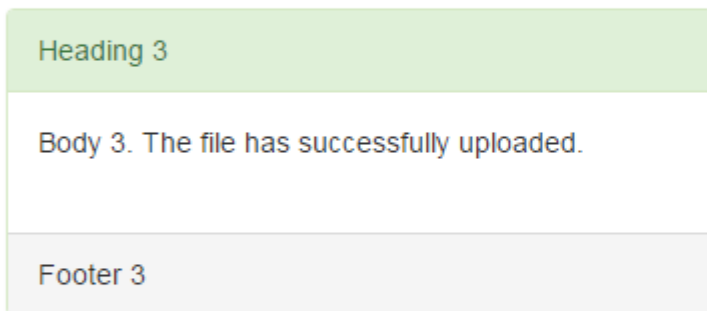
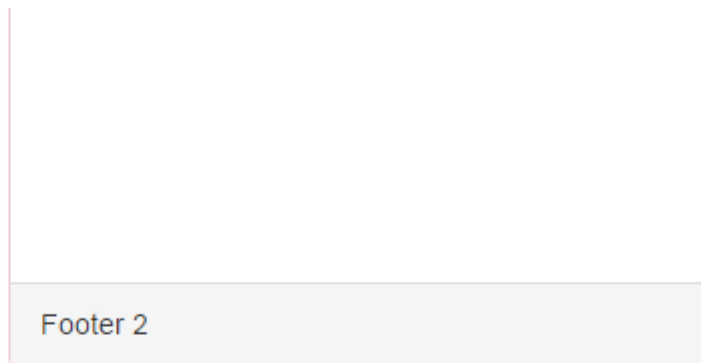
Heading 3
Body 3. The file has successfully uploaded.
Footer 3

Heading 5
Body 5.
Footer 5

Heading 6
Body 6.

Heading 7
Body 7.
Footer 7

Et comme ça dans les petites fenêtres:



Au fait, j'utilise la classe de `panel` Bootstrap 3, qui disparaîtra dans Bootstrap 4 et sera remplacée par la `card` beaucoup plus descriptive et spécifique. En regardant ces images, vous pouvez voir pourquoi la `card` sera un nom bien meilleur que le `panel` ambigu.

## 2,4,6 Disposition avec Clearfixes

Voici une mise en page qui rend deux, quatre ou six cellules en fonction de la taille de l'écran.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-6 col-md-3 col-lg-2">1</div>
    <div class="col-xs-6 col-md-3 col-lg-2 cell-tall">2</div>
  <!--
    On small and extra small devices, the viewport will render TWO cells
    (12 / 6 = 2), so we need a clearfix every TWO cells. We also need to
    say "don't show this clearfix when the viewport will render FOUR cells",
    which it will do at medium size (12 / 3 = 4). We do that by adding
    hidden-md and hidden-lg to our clearfix div, in effect instructing the
    browser to not show it at all on a wider screen.
  -->
  <div class="clearfix hidden-md hidden-lg"></div>
<!-->>
```

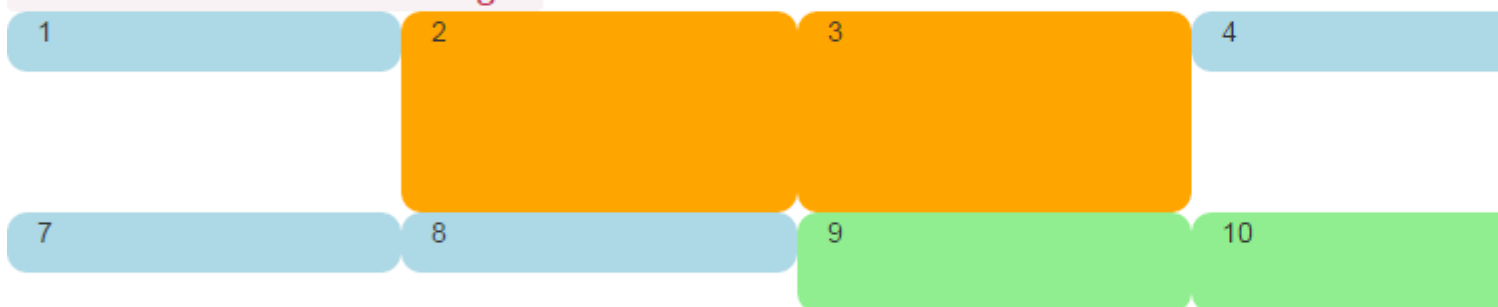
```

<div class="col-xs-6 col-md-3 col-lg-2 cell-tall">3</div>
<div class="col-xs-6 col-md-3 col-lg-2">4</div>
<!--
    After the FOURTH cell, we need a clearfix, but it still needs to be
    hidden on LARGE viewports, because remember we will have a maximum of
    SIX cells now.
-->
<div class="clearfix hidden-lg"></div>
<!------>
<div class="col-xs-6 col-md-3 col-lg-2">5</div>
<div class="col-xs-6 col-md-3 col-lg-2 cell-med">6</div>
<!--
    After the SIXTH cell, we need to show on SMALL and LARGE, but not on
    MEDIUM. Remember, our MEDIUM viewport only wants a clearfix when we
    are at a multiple of FOUR.
-->
<div class="clearfix hidden-md"></div>
<!------>
<div class="col-xs-6 col-md-3 col-lg-2">7</div>
<div class="col-xs-6 col-md-3 col-lg-2">8</div>
<!--
    Now we have rendered EIGHT cells, which is a multiple of TWO AND FOUR,
    so we put in a clearfix that's not visible on LARGE, because we are NOT
    at a multiple of SIX.
-->
<div class="clearfix hidden-lg"></div>
<!------>
<div class="col-xs-6 col-md-3 col-lg-2 cell-med">9</div>
<div class="col-xs-6 col-md-3 col-lg-2 cell-med">10</div>
<!--
    After the 10th cell, small only.
-->
<div class="clearfix hidden-md hidden-lg"></div>
<!------>
<div class="col-xs-6 col-md-3 col-lg-2">11</div>
</div>
</div>

```

Grand écran:

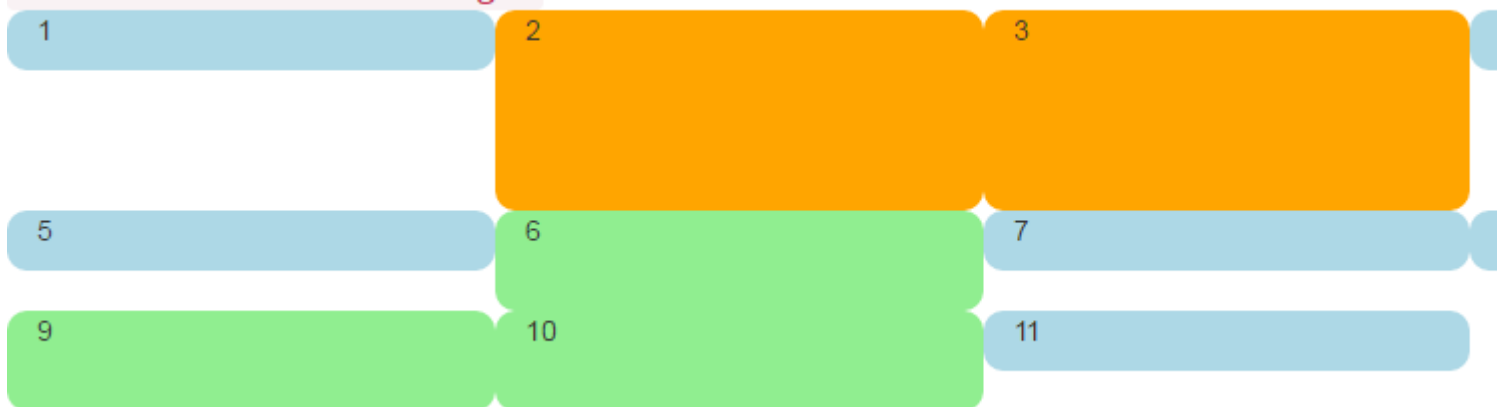
col-xs-6 col-md-3 col-lg-2



Écran moyen:

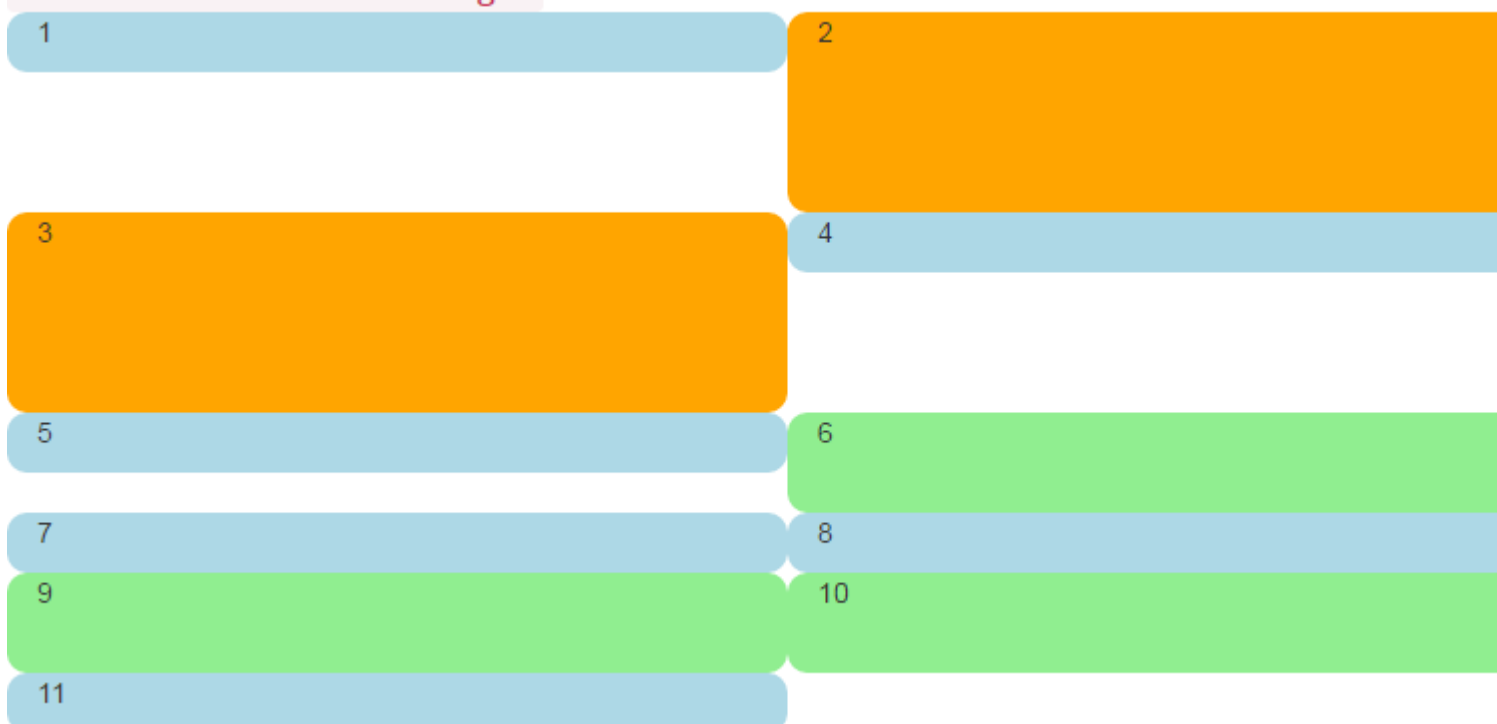


```
col-xs-6 col-md-3 col-lg-2
```



Petit écran:

```
col-xs-6 col-md-3 col-lg-2
```



## Pourquoi les colonnes bootstrap dépasseraient-elles 12 dans une rangée?

Il existe de nombreux scénarios `.row` lesquels il est *nécessaire* d'avoir des unités de colonne supérieures à 12 dans un seul élément `.row`. Ceci est connu sous le nom d' [enveloppe de colonne](#).

Si plus de 12 colonnes sont placées dans une seule ligne, chaque groupe de colonnes supplémentaires sera, comme une seule unité, placé sur une nouvelle ligne.

Par exemple, considérons une disposition où nous voulons ...

- 3 colonnes sur des périphériques grand et moyen, et
- 2 colonnes sur petits et petits appareils

## Large



## Small



Pour obtenir cette mise en page dans Bootstrap, nous utiliserions (correct) ..

```
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
```

### Démo méthode correcte

Comme vous le voyez dans l'exemple, le *total* des unités de colonne dans l'élément `.row` **dépasse 12** . Cette technique est connue sous le nom d' [encapsulation de colonnes](#) et constitue l'une des fonctions de conception réactives les plus puissantes de Bootstrap. La mise en page désirée *ne* serait *pas possible* (autre que la duplication du balisage) si nous tentions de nous en tenir à l' **idée reçue selon laquelle les unités de colonnes doivent ajouter jusqu'à 12 unités dans une**

## seule ligne .

La mise en page n'est *pas* possible lorsque nous ne dépassons pas 12 (faux) ..

```
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
<div class="row">
  <div class="col-xs-6 col-md-4"> x </div>
  <div class="col-xs-6 col-md-4"> x </div>
</div>
```

### Démo de méthode incorrecte (échec de 3 colonnes sur une grande)

N'oubliez pas qu'un `.row` n'est *pas* la même chose qu'une seule ligne dans la fenêtre d'affichage. Un `.row` est un regroupement de colonnes. Les colonnes dépassant 12 unités dans un seul élément `.row` seront `.row` [à une nouvelle ligne](#) (dans la fenêtre). C'est pourquoi il est essentiel de comprendre que les 12 colonnes représentent des **unités** horizontales dans la fenêtre d'affichage.

De plus, les [réinitialisations](#) réactives (clearfix) doivent être utilisées pour un emballage uniforme lorsque la [hauteur des colonnes varie](#) .

[Lire Utiliser Clearfix dans les lignes et les colonnes en ligne: https://riptutorial.com/fr/twitter-bootstrap/topic/6124/utiliser-clearfix-dans-les-lignes-et-les-colonnes](https://riptutorial.com/fr/twitter-bootstrap/topic/6124/utiliser-clearfix-dans-les-lignes-et-les-colonnes)

# Chapitre 36: Validation bootstrap

## Remarques

- Cette technique de validation ne peut être utilisée que sur des entrées figurant dans un formulaire.
- Les propriétés doivent avoir au moins une exigence de validation pour afficher la mise en évidence sur une validation `onSubmit()` échouée. Les types de données (autres que string) ont une exigence de type de données masquée, donc ne nécessitent pas d'annotation de données explicite. Les chaînes ne disposent pas de cette `[MinLengthAttribute(0)]`, donc pour forcer une vérification de validation avec les autres champs, ajoutez l'annotation de données `[MinLengthAttribute(0)]`.

## Exemples

### Utilisation d'ASP.NET MVC et d'annotations de données

Ajoutez ce qui suit à `Web.config` (dans le dossier `Views`), dans `<appSettings>` :

```
<add key="ClientValidationEnabled" value="true"/>
<add key="UnobtrusiveJavaScriptEnabled" value="true"/>
```

Ajoutez le bundle `jqueryval` à `BundleConfig.cs`:

```
bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(
    "~/Scripts/jqueryval/jquery.validate*"));
```

Ajoutez ce qui suit à toutes les pages nécessitant une validation (ou `_Layout.cshml`):

```
<!-- Reference to the jqueryval bundle -->
@Scripts.Render("~/bundles/jqueryval")

<!-- jQuery to apply bootstrap validation classes and glyphs to inputs -->
<script type="text/javascript">
    $.validator.defaults({
        highlight: function (element) {
            $(element).closest('.form-group').removeClass('has-success has-
feedback').addClass('has-error has-feedback'); // red highlighting
            $(element).closest('.form-group').find('.form-control-
feedback').removeClass('glyphicon-ok').addClass('glyphicon-remove'); // red cross glyphicon
        },
        unhighlight: function (element) {
            $(element).closest('.form-group').removeClass('has-error has-
feedback').addClass('has-success has-feedback'); // green highlighting
            $(element).closest('.form-group').find('.form-control-
feedback').removeClass('glyphicon-remove').addClass('glyphicon-ok'); // green tick glyphicon
        }
    });
```

```
</script>
```

Ajoutez des annotations de données aux champs pertinents du modèle:

```
using System.ComponentModel.DataAnnotations;  
  
[Required(ErrorMessage = "This field is required.")
```

Dans la vue, ajoutez ce qui suit à chaque entrée nécessitant une validation:

```
<!-- Validation messages -->  
<div class="text-danger">@Html.ValidationMessageFor(m => m.SomeField)</div>  
  
<!-- Bootstrap feedback span: -->  
<span class="glyphicon form-control-feedback"></span>
```

Ajoutez ce qui suit à l'action de contrôleur appropriée pour ajouter une validation côté serveur:

```
if (!ModelState.IsValid)  
{  
    return View(model);  
}  
else  
{  
    // continue with action  
}
```

---

## Exemple de saisie nécessitant une validation

Modèle:

```
[Required(ErrorMessage = "This field is required.")  
StringLength(maximumLength: 10, ErrorMessage = "This field must be 10 characters or less.")]  
public string SomeRequiredField { get; set; }
```

Vue:

```
<div class="form-group has-feedback">  
    <div class="col-md-4">  
        @Html.LabelFor(m => m.SomeRequiredField, new { @class = "control-label" })  
    </div>  
    <div class="col-md-8">  
        @Html.TextBoxFor(m => m.SomeRequiredField, new { @class = "form-control" })  
        <div class="text-danger">@Html.ValidationMessageFor(m => m.SomeRequiredField)</div>  
        <span class="glyphicon form-control-feedback"></span>  
    </div>  
</div>
```

---

## Optionnel

Ajoutez le jQuery suivant pour valider les entrées sur le flou, ainsi que sur submit:

```
$('#input').on('blur', function () {  
    $(this).valid();  
});
```

Lire Validation bootstrap en ligne: <https://riptutorial.com/fr/twitter-bootstrap/topic/6388/validation-bootstrap>

# Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec twitter-bootstrap	<a href="#">andreaem</a> , <a href="#">Atul Mishra</a> , <a href="#">bpoiss</a> , <a href="#">Community</a> , <a href="#">Evan</a> , <a href="#">Gleb Kemarsky</a> , <a href="#">H. Pauwelyn</a> , <a href="#">kernal lora</a> , <a href="#">Kritner</a> , <a href="#">MattD</a> , <a href="#">Mingle Li</a> , <a href="#">Nhan</a> , <a href="#">Prashanth Benny</a> , <a href="#">the12</a> , <a href="#">tmg</a> , <a href="#">Toby</a> , <a href="#">VincenzoC</a> , <a href="#">Vishnu Y S</a>
2	Alerte	<a href="#">JHS</a> , <a href="#">Madalina Taina</a> , <a href="#">tmg</a>
3	Badges et étiquettes Bootstrap	<a href="#">mmativ</a>
4	Bootstrap Affixe	<a href="#">Ilyas karim</a>
5	Bootstrap Dropdowns	<a href="#">Ismail Farooq</a> , <a href="#">MattD</a>
6	Bootstrap Navbar	<a href="#">Ilyas karim</a>
7	Boutons	<a href="#">Madalina Taina</a> , <a href="#">Muhammad Abdullah</a> , <a href="#">Richard Hamilton</a> , <a href="#">the12</a>
8	Carrousels	<a href="#">alex</a> , <a href="#">Boysenb3rry</a>
9	Classes d'utilité	<a href="#">ajju</a>
10	Colonnes	<a href="#">kybernaut.cz</a>
11	Composants Bootstrap	<a href="#">alex</a>
12	Conteneurs Bootstrap	<a href="#">Neha Chopra</a>
13	Dialogues modaux	<a href="#">alex</a> , <a href="#">mnoronha</a>
14	Formes	<a href="#">Community</a> , <a href="#">Jens</a> , <a href="#">Owen Pauling</a>
15	Glyphicons	<a href="#">Madalina Taina</a> , <a href="#">tmg</a> , <a href="#">Umer Farooq</a>
16	Groupe de liste	<a href="#">Ilyas karim</a>
17	Imbrication de la grille	<a href="#">neophyte</a> , <a href="#">ZimSystem</a>
18	Impression dans	<a href="#">MattD</a>

	Bootstrap.	
19	Info-bulle	<a href="#">Madalina Taina</a> , <a href="#">tmg</a>
20	Jumbotron	<a href="#">Gabriel Chi Hong Lee</a>
21	Les dropdowns	<a href="#">alex</a>
22	les tables	<a href="#">atjoedonahue</a>
23	Menus de navigation	<a href="#">Ignacio Correia</a>
24	Migration vers Bootstrap 4	<a href="#">Chris Farmer</a> , <a href="#">neophyte</a> , <a href="#">ZimSystem</a>
25	Modals	<a href="#">John Blanchard</a>
26	Navbar	<a href="#">Kritner</a> , <a href="#">Krunal Mevada</a> , <a href="#">kybernaut.cz</a>
27	Navs	<a href="#">leowebguy</a>
28	Onglets	<a href="#">DavidG</a> , <a href="#">tmg</a>
29	Pagination	<a href="#">TheDarkKnight</a>
30	Panneaux	<a href="#">JackPoint</a> , <a href="#">tmg</a>
31	Système de grille	<a href="#">Ani Menon</a> , <a href="#">Boysenb3rry</a> , <a href="#">bpoiss</a> , <a href="#">Harshal Patil</a> , <a href="#">leowebguy</a> , <a href="#">Madalina Taina</a> , <a href="#">Mingle Li</a> , <a href="#">mmativ</a> , <a href="#">Stephen Leppik</a> , <a href="#">the12</a> , <a href="#">tmg</a> , <a href="#">ZimSystem</a>
32	Thèmes Bootstrap	<a href="#">KAI</a>
33	Twitter Personnalisation du style Bootstrap	<a href="#">CENT1PEDE</a> , <a href="#">Vikas Yadav</a>
34	Utiliser Clearfix dans les lignes et les colonnes	<a href="#">Bruce Pierson</a> , <a href="#">ZimSystem</a>
35	Validation bootstrap	<a href="#">Amy Barrett</a> , <a href="#">mnoronha</a>