



FREE eBook

LEARNING

twitter-bootstrap-3

Free unaffiliated eBook created from
Stack Overflow contributors.

#twitter-

bootstrap-3

Table of Contents

About.....	1
Chapter 1: Getting started with twitter-bootstrap-3.....	2
Remarks.....	2
Versions.....	2
Examples.....	2
Installation.....	2
Hello World.....	3
Bootstrap installation and getting started.....	4
Chapter 2: Buttons.....	9
Parameters.....	9
Examples.....	9
Button types.....	9
Button size.....	9
Chapter 3: Buttons in twitter-bootstrap-3.....	11
Examples.....	11
Using twitter-bootstrap `btn` class.....	11
different sizes of button in twitter-bootstrap-3.....	11
Add Glyphicon to Button.....	12
Chapter 4: Grid System.....	13
Examples.....	13
Media queries.....	13
Grid options.....	13
Stacked-to-horizontal.....	14
Fluid container.....	15
Mobile and desktop.....	15
Mobile, tablet, desktop.....	15
Column wrapping.....	16
Responsive column resets.....	16
Offsetting columns.....	17
Nesting columns.....	17

Column ordering	17
Less mixins and variables	18
Variables	18
Mixins	18
Chapter 5: Grid System	21
Examples	21
Bootstrap Grid System	21
title	21
Credits	23

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [twitter-bootstrap-3](#)

It is an unofficial and free twitter-bootstrap-3 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-3.

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

Chapter 1: Getting started with twitter-bootstrap-3

Remarks

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap is the second most-starred project on GitHub, with almost 100,000 stars and almost 45,000 forks.

Versions

Version	Release Date
3.3.7	2016-07-25

Examples

Installation

- **Direct Download** - [download link](#)
- **CDN** - [get here](#)
- **Bower** - `bower install bootstrap` [\[read\]](#)
- **NPM** - `npm install bootstrap` [\[read\]](#)
- **Composer** - `composer require twbs/bootstrap` [\[read\]](#)
- **Customize** - If you have your own config, you can customize [here](#).
- **Sass** - For Sass related projects you may get it [here](#).

Usage

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

Reference to the bootstrap js file is made using script tag just above body tag (see below). Also note bootstrap is using jQuery for most of its widgets - like accordion carousel etc.. so reference bootstrap js file below the jQuery js file.

****Sample****

```

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

<head>

  <title>Form Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>

<body>

  <div class="container">
    <h2>Form Email</h2>
    <form role="form">
      <div class="form-group">
        <label for="email">Email:</label>
        <input type="email" class="form-control" id="email" placeholder="Enter email">
      </div>
      <div class="form-group">
        <label for="pwd">Password:</label>
        <input type="password" class="form-control" id="pwd" placeholder="Enter password">
      </div>
      <div class="checkbox">
        <label><input type="checkbox"> Remember me</label>
      </div>
      <button type="submit" class="btn btn-default">Submit</button>
    </form>
  </div>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

</body>

</html>

```

Hello World

The following `HTML` page illustrates a simple Hello World page using `Bootstrap 3`.

The page contains a full-width navigation bar with example links and a drop-down control. The navigation bar takes advantage of Bootstrap's mobile first capabilities. It starts collapsed in mobile views and become horizontal as the available viewport width increases.

In addition, a `jumbotron` element has been used to display featured information.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Hello World</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
</head>

```

```

<body>

  <nav class="navbar navbar-default">
    <div class="container-fluid">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
          <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="#">Hello, World!</a>
      </div>

      <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

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

    <div class="container">
      <div class="jumbotron">
        <h1>Bootstrap</h1>
        <p>This is a simple hero unit, a simple jumbotron-style component for calling
extra attention to featured content or information.</p>
        <p><a class="btn btn-primary btn-lg">Learn more</a></p>
      </div>
    </div>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

</body>
</html>

```

Bootstrap installation and getting started

Introduction

So you want to start using bootstrap for your project? Great! then lets get started right now!.

What is bootstrap? Bootstrap is an open source library which you can use to make amazing **responsive** projects with using responsive design and simple code. Responsive Design is a design philosophy where in the design of the system (the representation and the layout) responds/adapts depending upon the layout of the device. The primary reason to keep your design responsive is to increase the reach of your application to a larger user base using an array of devices.

Installation

Bootstrap can be installed in many different ways and for many different kind of projects. In the list below i've placed some download and tutorial links on how to install bootstrap

Download links

- **Direct Download** - [download link](#)
- **CDN** - [get here](#)
- **Bower** - `bower install bootstrap` [\[read\]](#)
- **NPM** - `npm install bootstrap` [\[read\]](#)
- **Composer** - `composer require twbs/bootstrap` [\[read\]](#)
- **Customize** - If you have your own config, you can customize [here](#).
- **Sass** - For Sass related projects you may get it [\[here\]\[1\]](#).

Basic information

So you've now installed bootstrap in your project. And now it is time to start using the great advantages of bootstrap. First i'm going to tell you some basic usage of bootstrap, after that i'll show some small examples and at the end i'll give you a startup code example which you can use as a starting template!

The grid system

Bootstrap uses a grid system. This grid system normally consists of 12 columns. Each of these 12 columns has the same width but can have different heights.

So we have a grid system that consists of 12 columns. We can use these columns to build our basic website. Let's say that we want to achieve the following layout:

menu - full width

sidebar - 1/3 of the screen

Main content 2/3 of the screen

Footer - full width

The menu

First we are going to look at the menu. As we know the grid system works with 12 columns. Since we want the menu on full width we have to put the menu in all of the 12 columns. It will look like the example below

```
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
  Menu
</div>
```

By giving the menu a `col-lg-12` class we indicate the following:

col - The `col` in the classname stands for columns.

lg - The lg in the classname stands for the width of the screen, in this case large.

12 - the 12 in the classname stands for the number of columns we want our menu to possess. since in this case we want the menu on a full width we get all the 12 columns (12/12)

Since we've used 12 of the 12 columns for our menu everything after the menu will be set on a new row.

The sidebar

The second item that we want to add to our template is the sidebar. Now as i've told we want the sidebar to be 1/3 of the screen. So what we are going to do is divide the 12 columns in 3. which is 4. We now know how many columns we want to fill to reach 1/3 of the screen. Follow the code below.

```
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
  The menu
</div>
<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
  The sidebar
</div>
```

The same as with the menu only now our number of col's differ from the menu.

col - The col in the classname stands for columns.

lg - The lg in the classname stands for the width of the screen, in this case large.

4 - the 4 in the classname stands for the number of columns we want our sidebar to possess. since in this case we want the sidebar to fill 1/3 of the screen so we'll only grab 4 of the 12 columns (4/12)

The content

Now on this row we've still got 8 columns left next to our sidebar. So now we are going to fill those up with our content. See the example code below

```
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
  The menu
</div>
<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
  The sidebar
</div>
<div class="col-lg-8 col-md-8 col-sm-8 col-xs-8">
  The main content
</div>
```

Now since we've filled up the remaining 8 columns of our 12 columns on this row the next section will again start on a new row with 12 columns.

The footer

The footer is, again just like the menu going to be a full width block on the screen so we'll grab all 12 columns on this row for our footer, see the example code below.

```
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
  The menu
</div>
```

```

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
  The sidebar
</div>
<div class="col-lg-8 col-md-8 col-sm-8 col-xs-8">
  The main content
</div>
<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
  The footer
</div>

```

So now we've created, with just a very small html file our first bootstrap template. But this is the very basics. Normally we would form this code a bit more to give it the full bootstrap experience. Some of these experiences are described down below.

Using rows and containers As i've told in the simple example above bootstrap uses rows of 12 columns. when a row is filled with 12 columns bootstrap will start on a new row. the best way to show these rows is by using row classes. We'll also use a container class. this is like a body tag, in this container we'll put all of our code. You can either choose between a container or a container-fluid class. The only difference is that the container-fluid class uses the full width of a screen and the container class doesn't. An example of these basic features is down below.

```

<div class="container-fluid">
  <div class="row">
    <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
      The menu consisting of 12 columns
    </div>
  </div>
  <div class="row">
    <div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
      The sidebar consisting of 4 columns
    </div>
    <div class="col-lg-8 col-md-8 col-sm-8 col-xs-8">
      The main content consisting of 8 columns
    </div>
  </div>
  <div class="row">
    <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
      The footer consisting of 12 columns
    </div>
  </div>
</div>

```

So we've now made a full page template with bootstrap. It's a very simple one indeed but start at the basics and later on you'll be able to use all sorts of bootstrap classes and functionalities. Last thing. The lg - md - sm and xs column names in the classes correspond, as i've told to the width of the page. LarGe, MeDium, SMall and XSmall. Don't forget you can use that to even style to columns differently on different width's by changing the amount of columns. Do remember a full row consists of 12 columns!

For more information visit: <http://getbootstrap.com/> For great examples visit: <http://expo.getbootstrap.com> or <http://bootsnipp.com/>

If you want me to add more information or if you'd come across some problems please let me know! and happy coding to you all!

Read [Getting started with twitter-bootstrap-3](https://riptutorial.com/twitter-bootstrap-3/topic/3060/getting-started-with-twitter-bootstrap-3) online: <https://riptutorial.com/twitter-bootstrap-3/topic/3060/getting-started-with-twitter-bootstrap-3>

Chapter 2: Buttons

Parameters

Class	Description
btn-default	Standard button. @brand-default: #fff
btn-primary	Provides extra visual weight and identifies the primary action. @brand-primary: darken(#428bca, 6.5%);
btn-success	Used to indicate a successful action. @brand-success: #5cb85c;
btn-info	Contextual button for providing information. @brand-info: #5bc0de;
btn-warning	Indicates caution should be applied by the user. @brand-warning: #f0ad4e;
btn-danger	Indicates a dangerous or negative action. @brand-danger: #d9534f;
btn-link	Use for link.

Examples

Button types

```
<button class="btn btn-default" type="button">Default</button>  
<button class="btn btn-primary" type="button">Primary</button>  
<button class="btn btn-success" type="button">Success</button>  
<button class="btn btn-info" type="button">Info</button>  
<button class="btn btn-warning" type="button">Warning</button>  
<button class="btn btn-danger" type="button">Danger</button>
```



```
<a class="btn btn-default" href="#" role="button">Link</a>  
<button class="btn btn-default" type="submit">Button</button>  
<input class="btn btn-default" type="button" value="Input">  
<input class="btn btn-default" type="submit" value="Submit">
```

Button size

The class for size button bootstrap is :

```
.btn-lg  
.btn-md  
.btn-sm  
.btn-xs
```



For example :

```
<button type="button" class="btn btn-primary btn-lg">Large</button>  
<button type="button" class="btn btn-primary btn-md">Medium</button>  
<button type="button" class="btn btn-primary btn-sm">Small</button>  
<button type="button" class="btn btn-primary btn-xs">XSmall</button>
```

Read Buttons online: <https://riptutorial.com/twitter-bootstrap-3/topic/5517/buttons>

Chapter 3: Buttons in twitter-bootstrap-3

Examples

Using twitter-bootstrap `btn` class

`btn` class of Twitter-bootstrap can be used with any of the following html elements.

1. anchor
2. button
3. input with both `type="button"` and `type="submit"`

Below are examples of all possible use cases of `btn` class

```
<a class="btn" href="#" role="button">Link</a>
<button class="btn" type="submit">Button</button>
<input class="btn" type="button" value="Input">
<input class="btn" type="submit" value="Submit">
```

Although `btn` class can be used in any of the above four ways but it is highly recommended using the `<button>` element whenever possible

Image of the above code is attached below

Link

Button

Input

Submit

Source

different sizes of button in twitter-bootstrap-3

twitter-bootstrap-3 has provided four different sizes of buttons

1. Large button `btn-lg`
2. Default button does not require any `btn` size
3. Small button `btn-sm`
4. Extra small button `btn-xs`

```
<button type="button" class="btn btn-lg">Large button</button>
<button type="button" class="btn">Default button</button>
<button type="button" class="btn btn-sm">Small button</button>
<button type="button" class="btn btn-xs">Extra small button</button>
```

Image of the above code is attached below

Large button

Default button

Small button

[Source](#)

Add Glyphicon to Button

Glyphicons can be used in text, buttons, toolbars, navigation, forms, etc (Source: W3Schools). Glyphicons are basically icon forms that can be used to style any of the aforementioned. These examples outline the usage of glyphicons inside two types of buttons by simply using a span inside the buttons which have a class of type glyphicon:

HTML Button

```
<button type="button" class="btn btn-info">
  <span class="glyphicon glyphicon-search"></span> Search
</button>
```



ASP Button

```
<asp:LinkButton runat="server" CssClass="btn btn-info" >
  <span class="glyphicon glyphicon-envelope"></span> Email
</asp:LinkButton>
```



Read Buttons in twitter-bootstrap-3 online: <https://riptutorial.com/twitter-bootstrap-3/topic/6071/buttons-in-twitter-bootstrap-3>

Chapter 4: Grid System

Examples

Media queries

We use the following media queries in our Less files to create the key breakpoints in our grid system.

```
/* Extra small devices (phones, less than 768px) */
/* No media query since this is the default in 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) { ... }
```

We occasionally expand on these media queries to include a max-width to limit CSS to a narrower set of devices.

```
@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) { ... }
```

Grid options

What is Bootstrap Grid System

Bootstrap grid system provides the quick and easy way to create responsive website layouts.

Bootstrap 3 includes predefined grid classes for quickly making grid layouts for different types of devices like cell phones, tablets, laptops and desktops, etc.

For example, you can use the `.col-xs-*` class to create grid columns for extra small devices like cell phones, similarly the `.col-sm-*` class for small screen devices like tablets, the `.col-md-*` class for medium size devices like desktops and the `.col-lg-*` for large desktop screens.

The following table summarizes some of the key features of the new grid system

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

Stacked-to-horizontal

Using a single set of `.col-md-*` grid classes, you can create a basic grid system that starts out stacked on mobile devices and tablet devices (the extra small to small range) before becoming horizontal on desktop (medium) devices. Place grid columns in any `.row`.

```
<div class="row">
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
```

```

</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>

```

Fluid container

Turn any fixed-width grid layout into a full-width layout by changing your outermost `.container` to `.container-fluid`.

```

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

```

Mobile and desktop

Don't want your columns to simply stack in smaller devices? Use the extra small and medium device grid classes by adding `.col-xs-*` `.col-md-*` to your columns. See the example below for a better idea of how it all works.

```

<!-- Stack the columns on mobile by making one full-width and the other half-width -->
<div class="row">
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
<div class="row">
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>

<!-- Columns are always 50% wide, on mobile and desktop -->
<div class="row">
  <div class="col-xs-6">.col-xs-6</div>
  <div class="col-xs-6">.col-xs-6</div>
</div>

```

Mobile, tablet, desktop

Build on the previous example by creating even more dynamic and powerful layouts with tablet `.col-sm-*` classes.

```

<div class="row">

```

```

<div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-md-8</div>
<div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>
</div>
<div class="row">
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
  <!-- Optional: clear the XS cols if their content doesn't match in height -->
  <div class="clearfix visible-xs-block"></div>
  <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
</div>

```

Column wrapping

If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line.

```

<div class="row">
  <div class="col-xs-9">.col-xs-9</div>
  <div class="col-xs-4">.col-xs-4<br>Since 9 + 4 = 13 &gt; 12, this 4-column-wide div gets
  wrapped onto a new line as one contiguous unit.</div>
  <div class="col-xs-6">.col-xs-6<br>Subsequent columns continue along the new line.</div>
</div>

```

Responsive column resets

With the four tiers of grids available you're bound to run into issues where, at certain breakpoints, your columns don't clear quite right as one is taller than the other. To fix that, use a combination of a `.clearfix` and our responsive utility classes.

```

<div class="row">
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>
  <div class="col-xs-6 col-sm-3">.col-xs-6 .col-sm-3</div>

  <!-- Add the extra clearfix for only the required viewport -->
  <div class="clearfix visible-xs-block"></div>

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

```

In addition to column clearing at responsive breakpoints, you may need to reset offsets, pushes, or pulls. See this in action in the grid example.

```

<div class="row">
  <div class="col-sm-5 col-md-6">.col-sm-5 .col-md-6</div>
  <div class="col-sm-5 col-sm-offset-2 col-md-6 col-md-offset-0">.col-sm-5 .col-sm-offset-2
  .col-md-6 .col-md-offset-0</div>
</div>

<div class="row">
  <div class="col-sm-6 col-md-5 col-lg-6">.col-sm-6 .col-md-5 .col-lg-6</div>
  <div class="col-sm-6 col-md-5 col-md-offset-2 col-lg-6 col-lg-offset-0">.col-sm-6 .col-md-5
  .col-md-offset-2 .col-lg-6 .col-lg-offset-0</div>

```

```
</div>
```

Offsetting columns

Move columns to the right using `.col-md-offset-*` classes. These classes increase the left margin of a column by `*` columns. For example, `.col-md-offset-4` moves `.col-md-4` over four columns.

```
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div>
</div>
<div class="row">
  <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
  <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div>
</div>
<div class="row">
  <div class="col-md-6 col-md-offset-3">.col-md-6 .col-md-offset-3</div>
</div>
```

You can also override offsets from lower grid tiers with `.col-*-offset-0` classes.

```
<div class="row">
  <div class="col-xs-6 col-sm-4">
</div>
  <div class="col-xs-6 col-sm-4">
</div>
  <div class="col-xs-6 col-xs-offset-3 col-sm-4 col-sm-offset-0">
</div>
</div>
```

Nesting columns

To nest your content with the default grid, add a new `.row` and set of `.col-sm-*` columns within an existing `.col-sm-*` column. Nested rows should include a set of columns that add up to 12 or fewer (it is not required that you use all 12 available columns).

```
<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>
```

Column ordering

Easily change the order of our built-in grid columns with `.col-md-push-*` and `.col-md-pull-*` modifier classes.

```
<div class="row">
  <div class="col-md-9 col-md-push-3">.col-md-9 .col-md-push-3</div>
  <div class="col-md-3 col-md-pull-9">.col-md-3 .col-md-pull-9</div>
</div>
```

Less mixins and variables

In addition to prebuilt grid classes for fast layouts, Bootstrap includes Less variables and mixins for quickly generating your own simple, semantic layouts.

Variables

Variables determine the number of columns, the gutter width, and the media query point at which to begin floating columns. We use these to generate the predefined grid classes documented above, as well as for the custom mixins listed below.

```
@grid-columns:          12;
@grid-gutter-width:     30px;
@grid-float-breakpoint: 768px;
```

Mixins

Mixins are used in conjunction with the grid variables to generate semantic CSS for individual grid columns.

```
// Creates a wrapper for a series of columns
.make-row(@gutter: @grid-gutter-width) {
  // Then clear the floated columns
  .clearfix();

  @media (min-width: @screen-sm-min) {
    margin-left: (@gutter / -2);
    margin-right: (@gutter / -2);
  }

  // Negative margin nested rows out to align the content of columns
  .row {
    margin-left: (@gutter / -2);
    margin-right: (@gutter / -2);
  }
}

// Generate the extra small columns
.make-xs-column(@columns; @gutter: @grid-gutter-width) {
  position: relative;
  // Prevent columns from collapsing when empty
  min-height: 1px;
```

```

// Inner gutter via padding
padding-left: (@gutter / 2);
padding-right: (@gutter / 2);

// Calculate width based on number of columns available
@media (min-width: @grid-float-breakpoint) {
  float: left;
  width: percentage((@columns / @grid-columns));
}
}

// Generate the small columns
.make-sm-column(@columns; @gutter: @grid-gutter-width) {
  position: relative;
  // Prevent columns from collapsing when empty
  min-height: 1px;
  // Inner gutter via padding
  padding-left: (@gutter / 2);
  padding-right: (@gutter / 2);

  // Calculate width based on number of columns available
  @media (min-width: @screen-sm-min) {
    float: left;
    width: percentage((@columns / @grid-columns));
  }
}

// Generate the small column offsets
.make-sm-column-offset(@columns) {
  @media (min-width: @screen-sm-min) {
    margin-left: percentage((@columns / @grid-columns));
  }
}

.make-sm-column-push(@columns) {
  @media (min-width: @screen-sm-min) {
    left: percentage((@columns / @grid-columns));
  }
}

.make-sm-column-pull(@columns) {
  @media (min-width: @screen-sm-min) {
    right: percentage((@columns / @grid-columns));
  }
}

// Generate the medium columns
.make-md-column(@columns; @gutter: @grid-gutter-width) {
  position: relative;
  // Prevent columns from collapsing when empty
  min-height: 1px;
  // Inner gutter via padding
  padding-left: (@gutter / 2);
  padding-right: (@gutter / 2);

  // Calculate width based on number of columns available
  @media (min-width: @screen-md-min) {
    float: left;
    width: percentage((@columns / @grid-columns));
  }
}

// Generate the medium column offsets

```

```

.make-md-column-offset(@columns) {
  @media (min-width: @screen-md-min) {
    margin-left: percentage((@columns / @grid-columns));
  }
}
.make-md-column-push(@columns) {
  @media (min-width: @screen-md-min) {
    left: percentage((@columns / @grid-columns));
  }
}
.make-md-column-pull(@columns) {
  @media (min-width: @screen-md-min) {
    right: percentage((@columns / @grid-columns));
  }
}

// Generate the large columns
.make-lg-column(@columns; @gutter: @grid-gutter-width) {
  position: relative;
  // Prevent columns from collapsing when empty
  min-height: 1px;
  // Inner gutter via padding
  padding-left: (@gutter / 2);
  padding-right: (@gutter / 2);

  // Calculate width based on number of columns available
  @media (min-width: @screen-lg-min) {
    float: left;
    width: percentage((@columns / @grid-columns));
  }
}

// Generate the large column offsets
.make-lg-column-offset(@columns) {
  @media (min-width: @screen-lg-min) {
    margin-left: percentage((@columns / @grid-columns));
  }
}
.make-lg-column-push(@columns) {
  @media (min-width: @screen-lg-min) {
    left: percentage((@columns / @grid-columns));
  }
}
.make-lg-column-pull(@columns) {
  @media (min-width: @screen-lg-min) {
    right: percentage((@columns / @grid-columns));
  }
}

```

Read Grid System online: <https://riptutorial.com/twitter-bootstrap-3/topic/5914/grid-system>

Chapter 5: Grid System

Examples

Bootstrap Grid System

Bootstrap uses a Grid System having rows and columns

In the Grid system, you are using a *row* class to create a horizontal box with a total of 12 columns of size 1 unit each for different screen size vertically. If you do not want to use all 12 columns individually, you can group the columns together to create wider columns.

Example : if you want to make a row of 3 columns - you have a div with class="row" (i.e one horizontal box) and 3 columns(class = col.xs.xx) each of size 3, size 2, size 7 (3+2+7 =12) for xs = extra small screen size of size

title

lorem ipsum

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

Grid Classes

The Bootstrap grid system has four classes for responsive design like this:

```
xs (for phones)
sm (for tablets)
md (for desktops)
lg (for larger desktops)
```

How to use?

For basic example 4 columns

```
<div class="row">
  <div class="col-sm-4">Your Div Content</div>
  <div class="col-sm-4">Your Div Content</div>
  <div class="col-sm-4">Your Div Content</div>
</div>
```

Example 4 columns

.col-sm-4

.col-sm-4

Read Grid System online: <https://riptutorial.com/twitter-bootstrap-3/topic/6173/grid-system>

Credits

S. No	Chapters	Contributors
1	Getting started with twitter-bootstrap-3	Abed Putra , alsobubbly , atjoedonahue , CENT1PEDE , claudios , Community , Deathstorm , Hussain Patel , Krunal Mevada , Nhan , Umut Esen
2	Buttons	Abed Putra , andreaem , Saeed.Gh
3	Buttons in twitter-bootstrap-3	Carl Bartlett , imaadhrizni , Muhammad Abdullah
4	Grid System	Huy Nguyen , kernal lora