

# Introduction to Bootstrap

By: Grant Hutchinson

## Index

Bootstrap Overview.....	Page 3
Download/Install.....	Page 4
Basic Usage.....	Page 4

### CSS

Working with grids.....	Page 5-7
Typography.....	Page 8
Tables.....	Page 9
Forms.....	Page 9-11
Buttons.....	Page 11-12
Images.....	Page 12
Helpers.....	Page 13

Official Bootstrap CSS Documentation:

<http://getbootstrap.com/css/>

## Bootstrap

**Bootstrap** is a set of tools to make it easier for web designers to code styles. It does this by utilizing pre-made HTML+CSS templates for: forms, buttons, typography, and UI elements.

**Bootstrap** was developed by Twitter internally as a way of consolidating web design tools. Over the years, Bootstrap grew with the support of twitter developers, to gain more functionality including compatibility across browsers and devices(responsive design).

In effect, Bootstrap is a series of LESS style sheets compiled to form each component of Bootstrap.

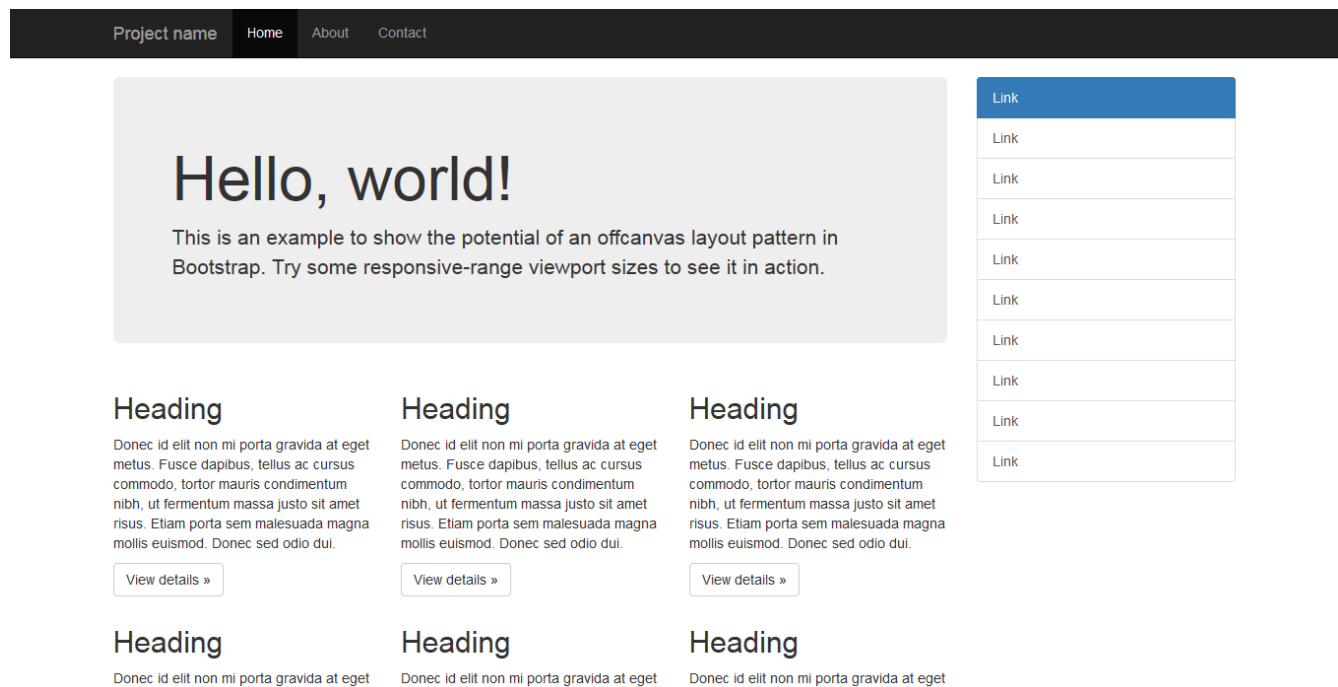


Image and page reference: [http://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](http://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

## Download:

<http://getbootstrap.com/getting-started/>

Bootstrap: - Minified CSS, already compiled, no docs or original source files  
<https://github.com/twbs/bootstrap/releases/download/v3.3.4/bootstrap-3.3.4-dist.zip>

Source code:  
<https://github.com/twbs/bootstrap/archive/v3.3.4.zip>

You can also link it in your page, without installing, using Bootstrap CDN:

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-
theme.min.css">
<!-- Latest compiled and minified JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>
```

## Basic Usage:

A typical bootstrapped page would include a link to bootstrap.min.css, as well as script elements for: jquery.min.js, bootstrap.min.js like so:

```
<!DOCTYPE html>
<html lang="en">
<head><title>Introduction to Bootstrap</title>
<link href="css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
<p>Hello, World!</p>
<!-- Place these elements at the bottom of the page, just before closing body, for faster loads -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

## **Try it!**

If your page displays “Hello, World!”, with styling, congratulations, you are ready to begin Bootstrap. If not, check that you've either:

- a) correctly replaced the src link with the bootstrap CDN link
- b) installed bootstrap in the correct location of your helloworld test
- c) check bootstrap src path is typed correctly.

## **Working with the grid**

Think of an entire piece of paper, now across the width, visually, divide it into 12, however you see fit! That's is the bottom line in bootstrap. Each column will be scaled based on its relative class number and column position, in comparison to the number 12!

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

12 divs, each individually spaced, padded, within a single row, all within a container.

.col-md-6	.col-md-6
-----------	-----------

2 divs, each stretched 6 columns. Each spaced, padded, within a single row, all within a container. Still equal to 12.

.col-md-8	.col-md-4
-----------	-----------

2 divs, the former is stretched 8 columns, while the latter is stretched 4. Each individually spaced, padded, within a single row, all within a container. Still equal to 12.

Depending on the device size you're working with, there are different ways of expressing

this grid.

Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-

### Try it!

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

## Explanation:

Bootstrap's grid system utilizes **containers** to hold the content, wrapping the site and grid. Each individual series of columns is wrapped in its **row**. If you add each column together, you'll notice the result is equal to 12. In instances where it is greater than 12, the column is moved to a new line and each proceeding column is lined up next to it.

These rows can then be **nested**, within each other, like this:

```
<div class="container">
  <div class="row">
    <div class="col-lg-9" style="background-color: red;">
      Level 1 col-lg-9
      <div class="row">
        <div class="col-md-6" style="background-color: green;">
          Level 2 col-md-6
        </div>
        <div class="col-md-6" style="background-color: blue;">
          Level 2 col-md-6
        </div>
      </div>
    </div>
  </div>
</div>
```

## Try that too!

Level 1: .col-sm-9

Level 2: .col-xs-8 .col-sm-6

Level2: .col-xs-4 .col-sm-6

You'll notice our default container is floating there in the middle of the page, to set our container from a fixed width, to the entire width of the page, use **container-fluid**

You can also offset your columns by adding **col-md-offset-N** (where N is the size of the offset) e.g.

```
<div class="row">
  <div class="col-md-4" style="background-color:red;">.col-md-4</div>
  <div class="col-md-4 col-md-offset-4" style="background-color:gray;">.col-md-4 .col-md-
offset-4</div>
</div>
<div class="row">
  <div class="col-md-3 col-md-offset-3" style="background-color:green;">.col-md-3 .col-md-
offset-3</div>
  <div class="col-md-3 col-md-offset-3" style="background-color:blue;">.col-md-3 .col-md-
offset-3</div>
</div>
<div class="row">
  <div class="col-md-6 col-md-offset-3" style="background-color:darkblue;">.col-md-6 .col-
md-offset-3</div></div>
```

## Typography

### Pre-formatted(styled) elements:

Headings <h1></h1><h2></h2><h3></h3> etc

Lists <ul><li> ... </li> </ul> <ol><li>.. </li></ol> class: list-style or list-unstyled, display:inline-block

Bold <strong> </strong>

Italics <em> </em>

Strikethrough <s></s>

Highlight <mark></mark>

Underline <u></u>

Small <small></small>

Commands(User Input) <kbd>cd </kbd>

Code <code> </code>

Block (multiple lines of code) <pre></pre>

Addresses <address></address>

Blockquote <blockquote>Some awesome quote – some guy</blockquote>

**Alignment:** e.g. <p class="text-center">center me!</p>

text-left

text-center

text-right

text-justify

text-nowrap

**Upper/lowercase/capitalize** <p class="text-capitalize">capitalize me!</p> = Capitalize Me!

text-lowercase

text-uppercase

text-capitalize



## **Tables**

There are at least 5 styled classes, for table they're

table - default bootstrapped table style

table-striped – every other row is striped a shade of a color

table-bordered - table is surrounded with a border

table-hover - rows highlight with a shade on hover

table-condensed - table columns padding is halved, unnoticable on desktop

table-responsive – makes table scroll horizontall on smaller devices

```
<table class="table table-striped">
  <tr>
    <td>Row</td>
    <td>Positive</td>
    <td>Negatives</td>
  </tr>
  <tr>
    <td>1</td>
    <td>Easier to style!</td>
    <td>Learning curve to grids</td>
  </tr>
  <tr>
    <td>2</td>
    <td>Consolidated styles</td>
    <td>Extra dependencies</td>
  </tr>
  <tr>
    <td>3</td>
    <td>Pre-made, templated, components</td>
    <td>Too awesome!</td>
  </tr>
</table>
```

## **Forms**

There are several types of forms:

form - no class needed, default form element is styled

form-horizontal – aligns everything(labels, inputs) in a horizontal layout

form-inline – adds left-alignment and inline-block controls (note may require width adjustments)

### Example Horizontal form Usage:

```
<form class="form-horizontal">
  <div class="form-group">
    <label for="inputEmail" class="col-md-2 control-label">Email</label>
    <div class="col-md-10">
      <input type="email" class="form-control" id="inputEmail"
placeholder="name@some.com" />
    </div>
  </div>
  <div class="form-group">
    <label for="inputPass" class="col-md-2 control-label">Pass</label>
    <div class="col-md-10">
      <input type="password" class="form-control" id="inputPass"
placeholder="Password" />
    </div>
  </div>
  <div class="form-group">
    <div class="col-md-10 col-md-offset-2">
      <input type="submit" class="btn btn-default" />
    </div>
  </div>
</form>
```

### Input types:

text,password, datetime, datetime-local, date, month, time, week, number ,email, url, search, tel, color

### Text Area

```
<textarea class="form-control" rows="4"></textarea>
```

**Checkbox** - you can also set disabled/checked as well as checkbox-inline

```
<div class="checkbox">
  <label>
    <input type="checkbox" value="" />
    Some Checkbox Option
  </label>
</div>
```

**Selects** – use `<select multiple class="form-control">` for multiple options at once

```
<select class="form-control">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
```

### Height sizing form inputs

`input-sm`

`input-lg`

both inputs and labels(a whole form group) at once with:

`form-group-sm`

`form-group-lg`

### Set widths

using `col-lg-N` (where n is the column grid size)

### Help text

use class `help-block` within a span, associated with the `aria-describedby="helpBlock"`

## Buttons

There are 3 different ways to represent a button `<a>` `<button>` `<input>` with `<button>` being the standard recommend method.

### Basic Usage:

```
<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">
```

## Options

preset styled buttons for different events

btn

btn-primary

btn-default

btn-success

btn-info

btn-warning

btn-danger

btn-link

## Sizes

btn-lg

btn-sm

btn-xs

## States

active/disabled

## Images

### Scale

Image elements are made responsive by the **img-responsive** class which scales and fits the image to its parents element.

### Center

You can center an image within its parent by using **center-block**

### Shapes

img-rounded

img-circle

img-thumbnail

## Try it!

```
  
    
  
```

## **Helpers**

### **Contextual colors**

You can display text in a different color based on an event you want to make evident to the user

classes:

text-muted  
text-primary  
text-success  
text-info  
text-warning  
text-danger

example usage:

```
<p class="text-warning"> An error occurred! Please re-enter the form</p>
```

### **Context backgrounds**

You can display background colors, in any color you like, based on any event you want to make evident to the user

classes:

bg-primary  
bg-success  
bg-info  
bg-warning  
bg-danger

example usage:

```
<p class="bg-warning"> An error occurred! Please re-enter the form</p>
```

### **Close icon**

```
<button type="button" class="close" aria-label="Close"><span aria-hidden="true">&times;</span></button>
```

### **Carets**

```
<span class="caret"></span>
```

### **Quick floats**

Quickly float an element left or right with **pull-left** or **pull-right**

### **Center blocks**

Quickly set an element to a display:block as well as centered utilizing the margin, using **center-block**