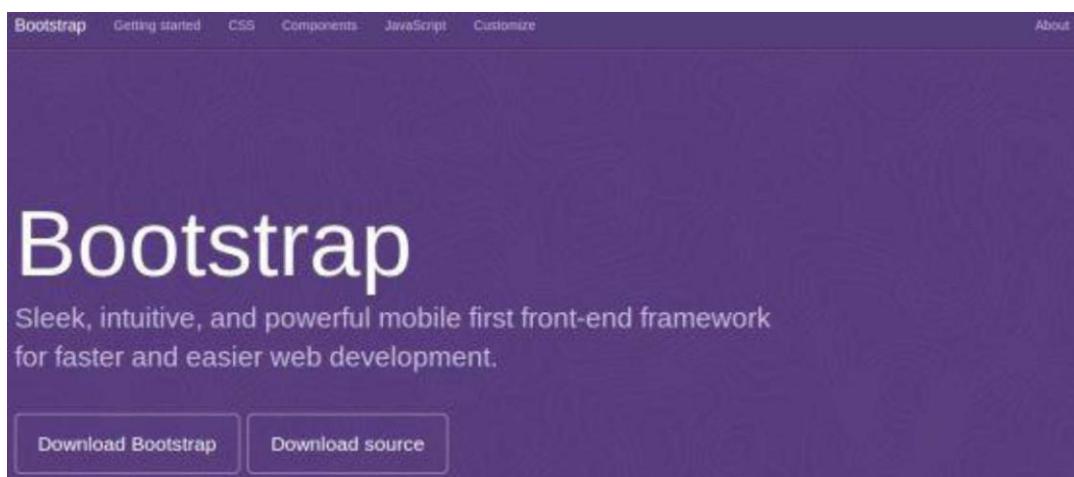


# Bootstrap Environment Setup

**I**t is very easy to setup and start using Bootstrap. This chapter will explain how to download and setup Bootstrap. It will also discuss the Bootstrap file structure, and demonstrate its usage with an example.

## Download Bootstrap

You can download the latest version of Bootstrap from <http://getbootstrap.com/>. When you click on this link, you will get to see a screen as below:



Here you can see two buttons:

- *Download Bootstrap*: Clicking this, you can download the precompiled and minified versions of Bootstrap CSS, JavaScript, and fonts. No documentation or original source code files are included.
- *Download Source*: Clicking this, you can get the latest Bootstrap LESS and JavaScript source code directly from GitHub.

If you work with Bootstrap's uncompiled source code, you need to compile the LESS files to produce usable CSS files. For compiling LESS files into CSS, Bootstrap officially supports only [Recess](#), which is Twitter's CSS hinter based on [less.js](#).

For better understanding and ease of use, we shall use precompiled version of Bootstrap throughout the tutorial . As the files are compiled and minified you don't have to bother every time including separate files for individual functionality. At the time of writing this tutorial the latest version (Bootstrap 3) was downloaded.

---

# File structure

## PRECOMPILED BOOTSTRAP

Once the compiled version Bootstrap is downloaded, extract the ZIP file, and you will see the following file/directory structure:

```
bootstrap/
  |-- css/
  |   |-- bootstrap.css
  |   |-- bootstrap.min.css
  |   |-- bootstrap-theme.css
  |   |-- bootstrap-theme.min.css
  |-- js/
  |   |-- bootstrap.js
  |   |-- bootstrap.min.js
  |-- fonts/
  |   |-- glyphicon-halflings-regular.eot
  |   |-- glyphicon-halflings-regular.svg
  |   |-- glyphicon-halflings-regular.ttf
  |   |-- glyphicon-halflings-regular.woff
```

As you can see there are compiled CSS and JS (bootstrap.\*), as well as compiled and minified CSS and JS (bootstrap.min.\*). Fonts from Glyphicons are included, as is the optional Bootstrap theme.

## BOOTSTRAP SOURCE CODE

If you downloaded the Bootstrap source code then the file structure would be as follows:

```
  |-- less/
  |-- js/
  |-- fonts/
  |-- dist/
  |   |-- css/
  |   |   |-- js/
  |   |   |   |-- fonts/
  |   |   |-- docs-assets/
  |   |   |-- examples/
  |   |   `-- *.html
```

- The files under *less/*, *js/*, and *fonts/* are the source code for Bootstrap CSS, JS, and icon fonts (respectively).
- The *dist/* folder includes everything listed in the precompiled download section above.
- *docs-assets/*, *examples/*, and all *\*.html* files are Bootstrap documentation.

## HTML Template

A basic HTML template using Bootstrap would look like as this:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap 101 Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media
        queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page
```

```

    via file:// -->
<!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/
    html5shiv.js"></script>
  <script src="https://oss.maxcdn.com/libs/respond.js/1.3.0/
    respond.min.js"></script>
<![endif]-->
</head>
<body>
  <h1>Hello, world!</h1>

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

```

Here you can see the **jquery.js** and **bootstrap.min.js** and **bootstrap.min.css** files are included to make a normal HTM file to Bootstrapped Template.

More details about each of the elements in this above piece of code will be discussed in the chapter [Bootstrap CSS Overview](#).

This template structure is already included as part of the **Try it** tool. Hence in all the examples (in the following chapters) of this tutorial you shall only see the contents of the **<body>** element. Once you click on the **Try it** option available at the top right corner of example, you will see the entire code.

## Example

Now let's try an example using the above template. Try following example using Try it option available at the top right corner of the below sample code box:

```
<h1>Hello, world!</h1>
```

In all the subsequent chapters we have used dummy text from the site <http://www.lipsum.com/>.

---

# Bootstrap Grid System

**I**n this chapter we shall discuss the Bootstrap Grid System.

## What is a Grid?

As put by wikipedia:

In graphic design, a grid is a structure (usually two-dimensional) made up of a series of intersecting straight (vertical, horizontal) lines used to structure content. It is widely used to design layout and content structure in print design. In web design, it is a very effective method to create a consistent layout rapidly and effectively using HTML and CSS.

To put it simple words grids in web design organise and structure content, makes websites easy to scan and reduces cognitive load on users.

## What is Bootstrap Grid System?

As put by the official documentation of Bootstrap for grid system:

Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases. It includes predefined classes for easy layout options, as well as powerful mixins for generating more semantic layouts.

Let us understand the above statement. Bootstrap 3 is mobile first in the sense that the code for Bootstrap now starts by targeting smaller screens like mobile devices, tablets, and then “expands” components and grids for larger screens such as laptops, desktops.

## MOBILE FIRST STRATEGY

- **Content**
    - Determine what is most important.
  - **Layout**
    - Design to smaller widths first.
-

- Base CSS address mobile device first; media queries address for tablet, desktops.
- **Progressive Enhancement**
  - Add elements as screen size increases.

## Working of Bootstrap Grid System

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the Bootstrap grid system works:

- Rows must be placed within a **.container** class for proper alignment and padding.
- Use rows to create horizontal groups of columns.
- Content should be placed within columns, and only columns may be immediate children of rows.
- Predefined grid classes like **.row** and **.col-xs-4** are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on **.rows**.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three **.col-xs-4**.

## Media Queries

Media query is a really fancy term for "conditional CSS rule". It simply applies some CSS based on certain conditions set forth. If those conditions are met, the style is applied.

Media Queries in Bootstrap allow you to move, show and hide content based on viewport size. Following media queries are used in LESS files to create the key breakpoints in the Bootstrap 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) { ... }
```

Occasionally these are expanded 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) { ... }
```

Media queries have two parts, a device specification and then a size rule. In the above case, the following rule is set:

Lets consider this line:

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

For all devices no matter what kind with *min-width*: @screen-sm-min if the width of the screen gets smaller than @screen-sm-max, *then do something*.

## Grid options

The following table summarizes aspects of how Bootstrap grid system works across multiple devices:

	<b>Extra small devices Phones (&lt;768px)</b>	<b>Small devices Tablets (≥768px)</b>	<b>Medium devices 2px</b>	<b>Large devices</b>
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
# of columns	12	12	12	12
Max column width	Auto	60px	78px	95px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

## BASIC GRID STRUCTURE

Following is basic structure of Bootstrap grid:

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

Let us see some simple grid examples:

- [Example: Stacked-to-horizontal](#)
- [Example: Medium and Large Device](#)
- [Example: Mobile, tablet, desktops](#)

### Bootstrap Grid System Example: Stacked-to-horizontal

Let us see a simple grid example with simple layout: two columns, two paragraphs per column. (Here styling for each column is used. You can avoid it.)

```

<div class="container">
  <h1>Hello, world!</h1>

  <div class="row">

    <div class="col-md-6" style="background-color: #dedef8; box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitatio ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

      <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.</p>
    </div>

    <div class="col-md-6" style="background-color: #dedef8; box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.</p>

      <p>Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnam aliquam quaerat voluptatem.</p>
    </div>
  </div>
</div>

```

## Hello, world!

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitatio ullamco laboris nisi ut aliquip ex ea commodo consequat.  
 Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.  
 Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnam aliquam quaerat voluptatem.

### Details

- `<div class="container">...</div>` element is added to ensure proper centering and maximum width for layout.
- Once container is added, next you need think in terms of rows. Add `<div class="row">...</div>` and columns `<div class="col-md-6"></div>` inside rows.
- Every row in the grid is made up of 12 units and you can define the desired size of your columns using those units. In our example we have two columns each made of 6 units wide i.e  $6+6=12$ .

You can try some more options like `<div class="col-md-3"></div>` and `<div class="col-md-9"></div>` or `<div class="col-md-7"></div>` and `<div class="col-md-5"></div>`

Experiment and make sure that the sum always needs to be 12.

## Bootstrap Grid System Example: Medium and Large Device

We have seen the basic grid system in [Example: Stacked-to-horizontal](#). Here we had used 2 divs and gave them the 50%/50% split at the medium viewport width:

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

But at large your design could really be better as a 33%/66%. So what we're going to do is set it up to change the column widths at the breakpoint:

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

Now Bootstrap is going to say "at the medium size, I look at classes with **md** in them and use those. At the large size, I look at classes with the word **lg** in them and use those. In this case, our 2 divs will go from a 50%/50% split and then up to a 33%/66%. Check it out in the following example. (Here styling for each column is used. You can avoid it.)

```
<div class="container">
  <h1>Hello, world!</h1>

  <div class="row">

    <div class="col-md-6 col-lg-4" style="background-color: #dedef8;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
      </p>

      <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
      </p>
    </div>

    <div class="col-md-6 col-lg-8" style="background-color: #dedef8;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.
      </p>

      <p>Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.
      </p>
    </div>
  </div>
</div>
```

Hello, world!

.....  
.....  
.....  
.....

.....  
.....  
.....  
.....

## Bootstrap Grid System Example: Mobile, Tablet, Desktops

We have seen an example for [Medium and Large Device](#). Now let us take it to another level, where we would want to change it for the extra small phone size as well. Say we want to add the option for the columns to be split 25%/75% for tablets, we go like this:

```
<div class="col-sm-3 col-md-6 col-lg-4">.....</div>
<div class="col-sm-9 col-md-6 col-lg-8">.....</div>
```

Now this gives us 3 different column layouts at each point. On a phone, it will be 25% on the left, and 75% on the right. On a tablet, it will be 50%/50% again, and on a large viewport, it will be 33%/66%. 3 different layouts for each of the 3 responsive sizes. Check it out in the following example. (Here styling for each column is used. You can avoid it.)

```
<div class="container">
  <h1>Hello, world!</h1>

  <div class="row">

    <div class="col-sm-3 col-md-6 col-lg-8"
        style="background-color: #dedef8;
        box-shadow: inset 1px -1px 1px #444,
                    inset -1px 1px 1px #444;">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
      </p>

      <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.
      </p>
    </div>

    <div class="col-sm-9 col-md-6 col-lg-4"
        style="background-color: #dedef8;
        box-shadow: inset 1px -1px 1px #444,
                    inset -1px 1px 1px #444;">
      <p>Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.
      </p>

      <p>Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnam aliquam quaerat voluptatem.
      </p>
    </div>
  </div>
</div>
```

Hello, world!

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium.  
Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incident ut labore et dolore magnam aliquam quaerat voluptatem.

## 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 class.**.clearfix** and the [responsive utility classes](#) as shown in the following example:

```
<div class="container">
  <div class="row" >
    <div class="col-xs-6 col-sm-3"
        style="background-color: #dedef8;
        box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    </div>
    <div class="col-xs-6 col-sm-3"
        style="background-color: #dedef8;box-shadow:
        inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
          eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
          enim ad minim veniam, quis nostrud exercitation ullamco laboris
          nisi ut aliquip ex ea commodo consequat.
      </p>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
          eiusmod tempor incididunt ut.
      </p>
    </div>

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

    <div class="col-xs-6 col-sm-3"
        style="background-color: #dedef8;
        box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco
          laboris nisi ut aliquip ex ea commodo consequat.
      </p>
    </div>
    <div class="col-xs-6 col-sm-3"
        style="background-color: #dedef8;box-shadow:
        inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
          eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
          enim ad minim
      </p>
    </div>
  </div>
</div>
```

Resize your viewport or check it out on your phone for a desired result of this example.

## Offset columns

Offsets are a useful feature for more specialized layouts. They can be used to push columns over for more spacing, for example. The **.col-xs-\*** classes don't support offsets, but they are easily replicated by using an empty cell.

To use offsets on large displays, use the **.col-md-offset-\*** classes. These classes increase the left margin of a column by \* columns where \* range from 1 to 11.

In the following example we have `<div class="col-md-6">..</div>`, we will center this using class **.col-md-offset-3**.

```
<div class="container">
  <h1>Hello, world!</h1>
```

```

<div class="row" >
  <div class="col-xs-6 col-md-offset-3"
    style="background-color: #dedef8;box-shadow:
    inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing
      elit.
    </p>
  </div>
</div>

```

Hello, world!

Lore ipsum dolor sit amet, consectetur adipisicing elit.

## Nesting columns

To nest your content with the default grid, add a new `.row` and set of `.col-md-*` columns within an existing `.col-md-*` column. Nested rows should include a set of columns that add up to 12.

In the following example, the layout has two columns, with the second one being split into four boxes over two rows.

```

<div class="container">

  <h1>Hello, world!</h1>

  <div class="row">

    <div class="col-md-3" style="background-color: #dedef8;box-shadow:
      inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <h4>First Column</h4>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    </div>

    <div class="col-md-9" style="background-color: #dedef8;box-shadow:
      inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <h4>Second Column- Split into 4 boxes</h4>
      <div class="row">
        <div class="col-md-6" style="background-color: #B18904;
          box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
          <p>Consectetur art party Tonx culpa semiotics. Pinterest
            assumenda minim organic quis.
        </p>
      </div>
      <div class="col-md-6" style="background-color: #B18904;
        box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
        <p> sed do eiusmod tempor incididunt ut labore et dolore magna
          aliqua. Ut enim ad minim veniam, quis nostrud exercitation
          ullamco laboris nisi ut aliquip ex ea commodo consequat.
      </p>
    </div>
  </div>

  <div class="row">
    <div class="col-md-6" style="background-color: #B18904;
      box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
      <p>quis nostrud exercitation ullamco laboris nisi ut
        aliquip ex ea commodo consequat.
    </p>
  </div>
</div>

```

```

        </p>
    </div>
    <div class="col-md-6" style="background-color: #B18904;
        box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit,
            sed do eiusmod tempor incididunt ut labore et dolore magna
            aliqua. Ut enim ad minim.</p>
    </div>
</div>

</div>

```

Hello, world!

First Column	Second Column- Split into 4 boxes
Consectetur adipisci velit, sed quia non numquam eiusmod tempora incidunt ut labore et dolore magnam aliquam ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	adipisicing elit, sed quia non numquam eiusmod tempora incidunt ut labore et dolore magnam aliquam ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
adipisicing elit, sed quia non numquam eiusmod tempora incidunt ut labore et dolore magnam aliquam ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	adipisicing elit, sed quia non numquam eiusmod tempora incidunt ut labore et dolore magnam aliquam ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
adipisicing elit, sed quia non numquam eiusmod tempora incidunt ut labore et dolore magnam aliquam ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	adipisicing elit, sed quia non numquam eiusmod tempora incidunt ut labore et dolore magnam aliquam ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

## Column ordering

Another nice feature of Bootstrap grid system is that you can easily write the columns in an order, and show them in another one. You can easily change the order of built-in grid columns with **.col-md-push-\*** and **.col-md-pull-\*** modifier classes where \* range from **1** to **11**.

In the following example we have two-columns layout with left column being the narrowest and acting as sidebar. We will swap the order of these columns using **.col-md-push-\*** and **.col-md-pull-\*** classes.

```

<div class="container">

    <h1>Hello, world!</h1>

    <div class="row">
        <p>Before Ordering</p>
        <div class="col-md-4" style="background-color: #dedef8;
            box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
            I am on left
        </div>
        <div class="col-md-8" style="background-color: #dedef8;
            box-shadow: inset 1px -1px 1px #444, inset -1px 1px 1px #444;">
            I am on right
        </div>
    </div><br>
    <div class="row">
        <p>After Ordering</p>
        <div class="col-md-4 col-md-push-8"
            style="background-color: #dedef8;
            box-shadow: inset 1px -1px 1px #444,
            inset -1px 1px 1px #444;">
            I was on left
        </div>
        <div class="col-md-8 col-md-pull-4"
            style="background-color: #dedef8;
            box-shadow: inset 1px -1px 1px #444,
            inset -1px 1px 1px #444;">
            I was on right
        </div>
    </div>

```

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

Hello, world!

Before Ordering

I am on left I am on right

After Ordering

I was on right I was on left

# Bootstrap Alerts

This chapter will discuss about alerts and the classes Bootstrap provides for alerts. Alerts provide a way to style messages to the user. They provide contextual feedback messages for typical user actions.

You can add an optional close icon to alert. For inline dismissal use the [Alerts jQuery plugin](#).

You can add an basic alert by creating a wrapper `<div>` and adding a class of `.alert` and one of the four contextual classes (e.g., `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`). The following example demonstrates this:

```
<div class="alert alert-success">Success! Well done its submitted.</div>
<div class="alert alert-info">Info! take this info.</div>
<div class="alert alert-warning">Warning ! Dont submit this.</div>
<div class="alert alert-danger">Error ! Change few things.</div>
```

Success! Well done its submitted.

Info! take this info.

Warning ! Dont submit this.

Error ! Change few things.

## Dismissal Alerts

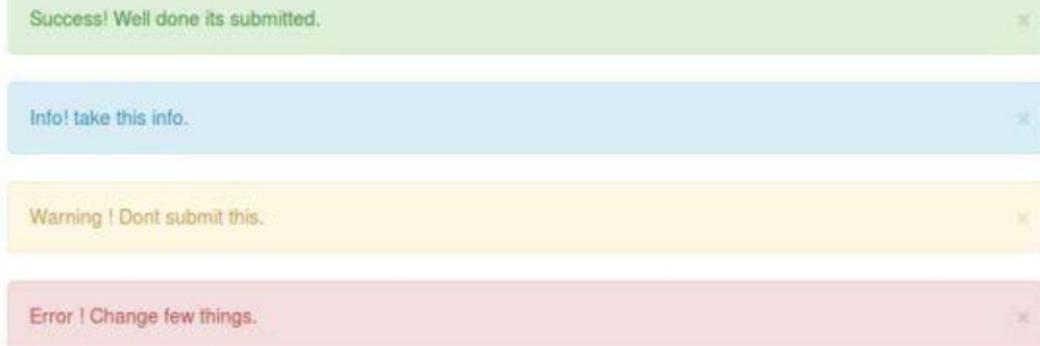
To build a dismissal alert:

- Add an basic alert by creating a wrapper `<div>` and adding a class of `.alert` and one of the four contextual classes (e.g., `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`)
- Also add optional `.alert-dismissible` to the above `<div>` class.
- Add a close button.

Following example demonstrates this:

```
<div class="alert alert-success alert-dismissible">
  <button type="button" class="close" data-dismiss="alert"
    aria-hidden="true">
    &times;
  </button>
  Success! Well done its submitted.
</div>
<div class="alert alert-info alert-dismissible">
  <button type="button" class="close" data-dismiss="alert"
    aria-hidden="true">
    &times;
  </button>
  Info! take this info.
</div>
<div class="alert alert-warning alert-dismissible">
  <button type="button" class="close" data-dismiss="alert"
    aria-hidden="true">
    &times;
  </button>
  Warning ! Dont submit this.
</div>
<div class="alert alert-danger alert-dismissible">
  <button type="button" class="close" data-dismiss="alert"
    aria-hidden="true">
    &times;
  </button>
  Error ! Change few things.
</div>
```

Be sure to use the `<button>` element with the `data-dismiss="alert"` data attribute.



## Links in Alerts

To get links in alerts:

- Add an basic alert by creating a wrapper `<div>` and adding a class of `.alert` and one of the four contextual classes (e.g., `.alert-success`, `.alert-info`, `.alert-warning`, `.alert-danger`)
- Use the `.alert-link` utility class to quickly provide matching colored links within any alert.

```
<div class="alert alert-success">
  <a href="#" class="alert-link">Success! Well done its submitted.</a>
</div>
```

# Bootstrap Progress Bars

**T**his chapter discusses about Bootstrap progress bars. The purpose of progress bars is to show that assets

are loading, in progress, or that there is action taking place regarding elements on the page.

Progress bars use CSS3 transitions and animations to achieve some of their effects. These features are not supported in Internet Explorer 9 and below or older versions of Firefox. Opera 12 does not support animations.

## Default Progress Bar

To create a basic progress bar:

- Add a `<div>` with a class of `.progress`.
- Next, inside the above `<div>`, add an empty `<div>` with a class of `.progress-bar`.
- Add a `style` attribute with the width expressed as a percentage. Say for example, `style="width: 60%;"`; indicates that the progress bar was at 60%.

Let us see an example below:

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="60"
      aria-valuemin="0" aria-valuemax="100" style="width: 40%;">
    <span class="sr-only">40% Complete</span>
  </div>
</div>
```



## Alternate Progress Bar

To create a progress bar with different styles:

- Add a `<div>` with a class of `.progress`.
- Next, inside the above `<div>`, add an empty `<div>` with a class of `.progress-bar` and class`progress-bar-`\* where \* could be `success`, `info`, `warning`, `danger`.

---

- Add a style attribute with the width expressed as a percentage. Say for example, style="60%"; indicates that the progress bar was at 60%.

Let us see an example below:

```
<div class="progress">
  <div class="progress-bar progress-bar-success" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 90%;">
    <span class="sr-only">90% Complete (Success)</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-info" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 30%;">
    <span class="sr-only">30% Complete (info)</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-warning" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 20%;">
    <span class="sr-only">20% Complete (warning)</span>
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-danger" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 10%;">
    <span class="sr-only">10% Complete (danger)</span>
  </div>
</div>
```



## Striped Progress Bar

To create a striped progress bar:

- Add a `<div>` with a class of **.progress** and **.progress-striped**.
- Next, inside the above `<div>`, add an empty `<div>` with a class of **.progress-bar** and class**progress-bar-\*** where \* could be **success**, **info**, **warning**, **danger**.
- Add a style attribute with the width expressed as a percentage. Say for example, `style="60%";` indicates that the progress bar was at 60%.

Let us see an example below:

```
<div class="progress progress-striped">
  <div class="progress-bar progress-bar-success" role="progressbar"
```

```

        aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
        style="width: 90%;">
            <span class="sr-only">90% Complete (Success)</span>
        </div>
    </div>
    <div class="progress progress-striped">
        <div class="progress-bar progress-bar-info" role="progressbar"
            aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
            style="width: 30%;">
            <span class="sr-only">30% Complete (info)</span>
        </div>
    </div>
    <div class="progress progress-striped">
        <div class="progress-bar progress-bar-warning" role="progressbar"
            aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
            style="width: 20%;">
            <span class="sr-only">20% Complete (warning)</span>
        </div>
    </div>
    <div class="progress progress-striped">
        <div class="progress-bar progress-bar-danger" role="progressbar"
            aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
            style="width: 10%;">
            <span class="sr-only">10% Complete (danger)</span>
        </div>
    </div>

```



## Animated Progress Bar

To create an animated progress bar:

- Add a `<div>` with a class of **.progress** and **.progress-striped**. Also add class **.active** to **.progress-striped**.
- Next, inside the above `<div>`, add an empty `<div>` with a class of **.progress-bar**.
- Add a style attribute with the width expressed as a percentage. Say for example, `style="60%"`; indicates that the progress bar was at 60%.

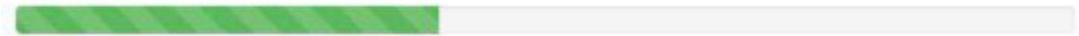
This will animate the stripes right to left.

Let us see an example below:

```

<div class="progress progress-striped active">
    <div class="progress-bar progress-bar-success" role="progressbar"
        aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
        style="width: 40%;">
        <span class="sr-only">40% Complete</span>
    </div>
</div>

```



## Stacked Progress Bar

You can even stack multiple progress bars. Place the multiple progress bars into the same `.progress` to stack them as seen in the following example:

```
<div class="progress">
  <div class="progress-bar progress-bar-success" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 40%;">
    <span class="sr-only">40% Complete</span>
  </div>
  <div class="progress-bar progress-bar-info" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 30%;">
    <span class="sr-only">30% Complete (info)</span>
  </div>
  <div class="progress-bar progress-bar-warning" role="progressbar"
    aria-valuenow="60" aria-valuemin="0" aria-valuemax="100"
    style="width: 20%;">
    <span class="sr-only">20% Complete (warning)</span>
  </div>
</div>
```



---

# Bootstrap Tables

**B**ootstrap provides a clean layout for building tables. Some of the table elements supported by Bootstrap are:

Tag	Description
<table>	Wrapping element for displaying data in a tabular format
<thead>	Container element for table header rows (<tr>) to label table columns
<tbody>	Container element for table rows (<tr>) in the body of the table
<tr>	Container element for a set of table cells (<td> or <th>) that appears on a single row
<td>	Default table cell
<th>	Special table cell for column (or row, depending on scope and placement) labels. Must be used within a <thead>
<caption>	Description or summary of what the table holds.

## Basic Table

If you want a nice, basic table style with just some light padding and horizontal dividers, add the base class of `.table` to any table as shown in the following example:

```
<table class="table">
  <caption>Basic Table Layout</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th>City</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Tanmay</td>
      <td>Bangalore</td>
    </tr>
    <tr>
      <td>Sachin</td>
      <td>Mumbai</td>
    </tr>
  </tbody>
</table>
```

---

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

Basic Table Layout

Name	City
Tanmay	Bangalore
Sachin	Mumbai

## Optional Table Classes

Along with the base table markup and the .table class, there are a few additional classes that you can use to style the markup. Following sections will give you a glimpse of all these classes.

### STRIPED TABLE

By adding the `.table-striped` class, you will get stripes on rows within the `<tbody>` as seen in the following example:

```
<table class="table table-striped">
  <caption>Striped Table Layout</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th>City</th>
      <th>Pincode</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Tanmay</td>
      <td>Bangalore</td>
      <td>560001</td>
    </tr>
    <tr>
      <td>Sachin</td>
      <td>Mumbai</td>
      <td>400003</td>
    </tr>
    <tr>
      <td>Uma</td>
      <td>Pune</td>
      <td>411027</td>
    </tr>
  </tbody>
</table>
```

Striped Table Layout

Name	City	Pincode
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

## BORDERED TABLE

By adding the `.table-bordered` class, you will get borders surrounding every element and rounded corners around the entire table as seen in the following example:

```
<table class="table table-bordered">
  <caption>Bordered Table Layout</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th>City</th>
      <th>Pincode</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Tanmay</td>
      <td>Bangalore</td>
      <td>560001</td>
    </tr>
    <tr>
      <td>Sachin</td>
      <td>Mumbai</td>
      <td>400003</td>
    </tr>
    <tr>
      <td>Uma</td>
      <td>Pune</td>
      <td>411027</td>
    </tr>
  </tbody>
</table>
```

Bordered Table Layout

Name	City	Pincode
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

## HOVER TABLE

By adding the `.table-hover` class, a light gray background will be added to rows while the cursor hovers over them, as seen in the following example:

```
<table class="table table-hover">
  <caption>Hover Table Layout</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th>City</th>
      <th>Pincode</th>
    </tr>
  </thead>
  <tbody>
    <tr>
```

```

        <td>Tanmay</td>
        <td>Bangalore</td>
        <td>560001</td>
    </tr>
    <tr>
        <td>Sachin</td>
        <td>Mumbai</td>
        <td>400003</td>
    </tr>
    <tr>
        <td>Uma</td>
        <td>Pune</td>
        <td>411027</td>
    </tr>
</tbody>
</table>

```

Hover Table Layout

Name	City	Pincode
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

## CONDENSED TABLE

By adding the `.table-condensed` class, row padding is cut in half to condense the table, as seen in the following example. This is useful if you want denser information.

```





```

Condensed Table Layout

Name	City	Pincode
Tanmay	Bangalore	560001
Sachin	Mumbai	400003
Uma	Pune	411027

## Contextual classes

The Contextual classes shown in following table will allow you to change the background color of your table rows or individual cells.

Class	Description
.active	Applies the hover color to a particular row or cell
.success	Indicates a successful or positive action
.warning	Indicates a warning that might need attention
.danger	Indicates a dangerous or potentially negative action

These classes can be applied to <tr>, <td> or <th>.

```
<table class="table">
  <caption>Contextual Table Layout</caption>
  <thead>
    <tr>
      <th>Product</th>
      <th>Payment Date</th>
      <th>Status</th>
    </tr>
  </thead>
  <tbody>
    <tr class="active">
      <td>Product1</td>
      <td>23/11/2013</td>
      <td>Pending</td>
    </tr>
    <tr class="success">
      <td>Product2</td>
      <td>10/11/2013</td>
      <td>Delivered</td>
    </tr>
    <tr class="warning">
      <td>Product3</td>
      <td>20/10/2013</td>
      <td>In Call to confirm</td>
    </tr>
    <tr class="danger">
      <td>Product4</td>
      <td>20/10/2013</td>
      <td>Declined</td>
    </tr>
  </tbody>
</table>
```

Contextual Table Layout		
Product	Payment Date	Status
Product1	23/11/2013	Pending
Product2	10/11/2013	Delivered
Product3	20/10/2013	In Call to confirm
Product4	20/10/2013	Declined

## Responsive tables

By wrapping any `.table` in `.table-responsive` class, you will make the table scroll horizontally up to small devices (under 768px). When viewing on anything larger than 768px wide, you will not see any difference in these tables.

```
<div class="table-responsive">
  <table class="table">
    <caption>Responsive Table Layout</caption>
    <thead>
      <tr>
        <th>Product</th>
        <th>Payment Date</th>
        <th>Status</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Product1</td>
        <td>23/11/2013</td>
        <td>Pending</td>
      </tr>
      <tr>
        <td>Product2</td>
        <td>10/11/2013</td>
        <td>Delivered</td>
      </tr>
      <tr>
        <td>Product3</td>
        <td>20/10/2013</td>
        <td>In Call to confirm</td>
      </tr>
      <tr>
        <td>Product4</td>
        <td>20/10/2013</td>
        <td>Declined</td>
      </tr>
    </tbody>
  </table>
</div>
```

### Responsive Table Layout

Product	Payment Date	Status
Product1	23/11/2013	Pending
Product2	10/11/2013	Delivered
Product3	20/10/2013	In Call to confirm
Product4	20/10/2013	Declined

1/4

2/4

# Bootstrap Forms

I

n this chapter we will study how to create forms with ease using Bootstrap. Bootstrap makes it easy with the simple HTML markup and extended classes for different styles of forms.

## Form Layout

Bootstrap provides you with following types of form layouts:

- Vertical (default) form
- Inline form
- Horizontal form

### VERTICAL OR BASIC FORM

The basic form structure comes with Bootstrap; individual form controls automatically receive some global styling. To create a basic form do the following:

- Add a role *form* to the parent `<form>` element.
- Wrap labels and controls in a `<div>` with class `.form-group`. This is needed for optimum spacing.
- Add a class of `.form-control` to all textual `<input>`, `<textarea>`, and `<select>` elements.

```
<form role="form">
  <div class="form-group">
    <label for="name">Name</label>
    <input type="text" class="form-control" id="name"
      placeholder="Enter Name">
  </div>
  <div class="form-group">
    <label for="inputfile">File input</label>
    <input type="file" id="inputfile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
</form>
```

---

```

        </label>
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
</form>

```

Name

Enter Name

File input

No file selected.

Example block-level help text here.

Check me out

## INLINE FORM

To create a form where all of the elements are inline, left aligned and labels are alongside, add the class `.form-inline` to the `<form>` tag.

```

<form class="form-inline" role="form">
    <div class="form-group">
        <label class="sr-only" for="name">Name</label>
        <input type="text" class="form-control" id="name"
               placeholder="Enter Name">
    </div>
    <div class="form-group">
        <label class="sr-only" for="inputfile">File input</label>
        <input type="file" id="inputfile">
    </div>
    <div class="checkbox">
        <label>
            <input type="checkbox"> Check me out
        </label>
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
</form>

```

Enter Name

No file selected.

Check me out

- By default Inputs, selects, and textareas have 100% width in Bootstrap. You need to set a width on the form controls when using inline form.
- Using the class `.sr-only` you can hide the labels of the inline forms.

## HORIZONTAL FORM

Horizontal forms stands apart from the others not only in the amount of markup, but also in the presentation of the form. To create a form that uses the horizontal layout, do the following:

- Add a class of `.form-horizontal` to the parent `<form>` element.
- Wrap labels and controls in a `<div>` with class `.form-group`.
- Add a class of `.control-label` to the labels.

```

<form class="form-horizontal" role="form">
  <div class="form-group">
    <label for="firstname" class="col-sm-2 control-label">First Name</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" id="firstname"
        placeholder="Enter First Name">
    </div>
  </div>
  <div class="form-group">
    <label for="lastname" class="col-sm-2 control-label">Last Name</label>
    <div class="col-sm-10">
      <input type="text" class="form-control" id="lastname"
        placeholder="Enter Last Name">
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <div class="checkbox">
        <label>
          <input type="checkbox" /> Remember me
        </label>
      </div>
    </div>
  </div>
  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <button type="submit" class="btn btn-default">Sign in</button>
    </div>
  </div>
</form>

```

The screenshot shows a sign-in form with the following structure:

- First Name:** Input field with placeholder "Enter First Name".
- Last Name:** Input field with placeholder "Enter Last Name".
- Remember me:** A checkbox labeled "Remember me".
- Sign in:** A large blue submit button.

## Supported Form Controls

Bootstrap natively supports the most common form controls mainly *input*, *textarea*, *checkbox*, *radio*, and *select*.

### INPUTS

The most common form text field is the *input*—this is where users will enter most of the essential form data. Bootstrap offers support for all native HTML5 input types: *text*, *password*, *datetime*, *datetime-local*, *date*, *month*, *time*, *week*, *number*, *email*, *url*, *search*, *tel*, and *color*. Proper *type* declaration is required to make *Inputs* fully styled.

```

<form role="form">
  <div class="form-group">
    <label for="name">Label</label>
    <input type="text" class="form-control" placeholder="Text input">
  </div>
</form>

```



## TEXTAREA

The textarea is used when you need multiple lines of input. Change `rows` attribute as necessary (fewer rows = smaller box, more rows = bigger box).

```
<form role="form">
  <div class="form-group">
    <label for="name">Text Area</label>
    <textarea class="form-control" rows="3"></textarea>
  </div>
</form>
```

## CHECKBOXES AND RADIOS

Checkboxes and radio buttons are great when you want users to choose from a list of preset options.

- When building a form, use `checkbox` if you want the user to select any number of options from a list. Use `radio` if you want to limit the user to just one selection.
- Use `.checkbox-inline` or `.radio-inline` class to a series of checkboxes or radios for controls appear on the same line.

The following example demonstrates both (default and inline) types:

```
<label for="name">Example of Default Checkbox and radio button </label>
<div class="checkbox">
  <label><input type="checkbox" value="">Option 1</label>
</div>
<div class="checkbox">
  <label><input type="checkbox" value="">Option 2</label>
</div>

<div class="radio">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios1"
           value="option1" checked> Option 1
  </label>
</div>
<div class="radio">
  <label>
    <input type="radio" name="optionsRadios" id="optionsRadios2"
           value="option2">
      Option 2 - selecting it will deselect option 1
  </label>
</div>
<label for="name">Example of Inline Checkbox and radio button </label>
<div>
  <label class="checkbox-inline">
    <input type="checkbox" id="inlineCheckbox1" value="option1"> Option 1
  </label>
  <label class="checkbox-inline">
```

```

<input type="checkbox" id="inlineCheckbox2" value="option2"> Option 2
</label>
<label class="checkbox-inline">
    <input type="checkbox" id="inlineCheckbox3" value="option3"> Option 3
</label>
<label class="checkbox-inline">
    <input type="radio" name="optionsRadiosinline" id="optionsRadios3"
        value="option1" checked> Option 1
</label>
<label class="checkbox-inline">
    <input type="radio" name="optionsRadiosinline" id="optionsRadios4"
        value="option2"> Option 2
</label>
</div>

```

#### Example of Default Checkbox and radio button

- Option 1
- Option 2
- Option 1
- Option 2 - selecting it will deselect option 1

#### Example of Inline Checkbox and radio button

- Option 1
- Option 2
- Option 3
- Option 1
- Option 2

## SELECTS

A select is used when you want to allow the user to pick from multiple options, but by default it only allows one.

- Use `<select>` for list options with which the user is familiar, such as states or numbers.
- Use `multiple="multiple"` to allow the user to select more than one option.

The following example demonstrates both (select and multiple) types:

```

<form role="form">
    <div class="form-group">
        <label for="name">Select list</label>
        <select class="form-control">
            <option>1</option>
            <option>2</option>
            <option>3</option>
            <option>4</option>
            <option>5</option>
        </select>

        <label for="name">Mutiple Select list</label>
        <select multiple class="form-control">
            <option>1</option>
            <option>2</option>
            <option>3</option>
            <option>4</option>
            <option>5</option>
        </select>
    </div>
</form>

```



## Static control

Use the class `.form-control-static` on a `<p>`, when you need to place plain text next to a form label within a horizontal form.

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label class="col-sm-2 control-label">Email</label>
    <div class="col-sm-10">
      <p class="form-control-static">email@example.com</p>
    </div>
  </div>
  <div class="form-group">
    <label for="inputPassword" class="col-sm-2 control-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword"
        placeholder="Password">
    </div>
  </div>
</form>
```

Email

Password

## Form Control States

In addition to the `:focus` (i.e., a user clicks into the input or tabs onto it) state, Bootstrap offers styling for disabled inputs and classes for form validation.

### INPUT FOCUS

When an input receives `:focus`, the outline of the input is removed and a `box-shadow` is applied.

### DISABLED INPUTS

If you need to disable an input, simply adding the `disabled` attribute will not only disable it; it will also change the styling and the mouse cursor when the cursor hovers over the element.

### DISABLED FIELDSETS

Add the `disabled` attribute to a `<fieldset>` to disable all the controls within the `<fieldset>` at once.

### VALIDATION STATES

Bootstrap includes validation styles for error, warning, and success messages. To use, simply add the appropriate class (`.has-warning`, `.has-error`, or `.has-success`) to the parent element.

Following example demonstrates all the form control states:

```
<form class="form-horizontal" role="form">
  <div class="form-group">
```

```
<label class="col-sm-2 control-label">Focused</label>
<div class="col-sm-10">
    <input class="form-control" id="focusedInput" type="text"
          value="This is focused...">
</div>
</div>
<div class="form-group">
    <label for="inputPassword" class="col-sm-2 control-label">
        Disabled
    </label>
    <div class="col-sm-10">
        <input class="form-control" id="disabledInput" type="text"
              placeholder="Disabled input here..." disabled>
    </div>
</div>
<fieldset disabled>
    <div class="form-group">
        <label for="disabledTextInput" class="col-sm-2 control-label">
            Disabled input (Fieldset disabled)
        </label>
        <div class="col-sm-10">
            <input type="text" id="disabledTextInput" class="form-control"
                  placeholder="Disabled input">
        </div>
    </div>
    <div class="form-group">
        <label for="disabledSelect" class="col-sm-2 control-label">
            Disabled select menu (Fieldset disabled)
        </label>
        <div class="col-sm-10">
            <select id="disabledSelect" class="form-control">
                <option>Disabled select</option>
            </select>
        </div>
    </div>
</fieldset>
<div class="form-group has-success">
    <label class="col-sm-2 control-label" for="inputSuccess">
        Input with success
    </label>
    <div class="col-sm-10">
        <input type="text" class="form-control" id="inputSuccess">
    </div>
</div>
<div class="form-group has-warning">
    <label class="col-sm-2 control-label" for="inputWarning">
        Input with warning
    </label>
    <div class="col-sm-10">
        <input type="text" class="form-control" id="inputWarning">
    </div>
</div>
<div class="form-group has-error">
    <label class="col-sm-2 control-label" for="inputError">
        Input with error
    </label>
    <div class="col-sm-10">
        <input type="text" class="form-control" id="inputError">
    </div>
</div>
```

```
</form>
```

<b>Focused</b>	This is focused...
<b>Disabled</b>	Disabled input here...
<b>Disabled input (Fieldset disabled)</b>	Disabled input
<b>Disabled select menu (Fieldset disabled)</b>	Disabled select
<b>Input with success</b>	
<b>Input with warning</b>	
<b>Input with error</b>	

## Form Control Sizing

You can set heights and widths of forms using classes like `.input-lg` and `.col-lg-*` respectively. Following example demonstrates this:

```
<form role="form">
  <div class="form-group">
    <input class="form-control input-lg" type="text"
          placeholder=".input-lg">
  </div>

  <div class="form-group">
    <input class="form-control" type="text" placeholder="Default input">
  </div>

  <div class="form-group">
    <input class="form-control input-sm" type="text"
          placeholder=".input-sm">
  </div>
  <div class="form-group">
    <select class="form-control input-lg">
      <option value="">.input-lg</option>
    </select>
  </div>
  <div class="form-group">
    <select class="form-control">
      <option value="">Default select</option>
    </select>
  </div>
  <div class="form-group">
    <select class="form-control input-sm">
      <option value="">.input-sm</option>
    </select>
  </div>

  <div class="row">
    <div class="col-lg-2">
      <input type="text" class="form-control" placeholder=".col-lg-2">
    </div>
    <div class="col-lg-3">
```

```
<input type="text" class="form-control" placeholder=".col-lg-3">
</div>
<div class="col-lg-4">
    <input type="text" class="form-control" placeholder=".col-lg-4">
</div>
</div>
</form>
```

.input-lg

Default input

input sm

.input-lg



Default select



.input-sm



.col-lg-2

.col-lg-3

.col-lg-4

## Help Text

Bootstrap form controls can have a block level help text that flows with the inputs. To add a full width block of content, use the `.help-block` after the `<input>`. Following example demonstrates this:

```
<form role="form">
    <span>Example of Help Text</span>
    <input class="form-control" type="text" placeholder="">
    <span class="help-block">A longer block of help text that
        breaks onto a new line and may extend beyond one line.</span>
</form>
```

Example of Help Text

A longer block of help text that breaks onto a new line and may extend beyond one line.

# Bootstrap Buttons

**T**

his chapter will discuss about how to use Bootstrap button with examples. Anything that is given a class

of **.btn** will inherit the default look of a gray button with rounded corners. However Bootstrap provides some options to style buttons, which are summarized in the following table:

Class	Description
Btn	Default/ Standard button.
btn-primary	Provides extra visual weight and identifies the primary action in a set of buttons.
btn-success	Indicates a successful or positive action.
btn-info	Contextual button for informational alert messages.
btn-warning	Indicates caution should be taken with this action.
btn-danger	Indicates a dangerous or potentially negative action.
btn-link	Deemphasize a button by making it look like a link while maintaining button behavior.

Following example demonstrates all the above button classes:

```
<!-- Standard button -->
<button type="button" class="btn btn-default">Default Button</button>

<!-- Provides extra visual weight and identifies the primary action in a set of buttons -->
<button type="button" class="btn btn-primary">Primary Button</button>

<!-- Indicates a successful or positive action -->
<button type="button" class="btn btn-success">Success Button</button>

<!-- Contextual button for informational alert messages -->
<button type="button" class="btn btn-info">Info Button</button>

<!-- Indicates caution should be taken with this action -->
<button type="button" class="btn btn-warning">Warning Button</button>

<!-- Indicates a dangerous or potentially negative action -->
<button type="button" class="btn btn-danger">Danger Button</button>
```

```
<!-- Deemphasize a button by making it look like a link while maintaining button behavior -->
<button type="button" class="btn btn-link">Link Button</button>
```



## Button Size

The following table summarizes classes used to get buttons of various sizes:

Class	Description
.btn-lg	This makes button size large.
.btn-sm	This makes button size small.
.btn-xs	This makes button size with extra small.
.btn-block	This creates block level buttons those that span the full width of a parent.

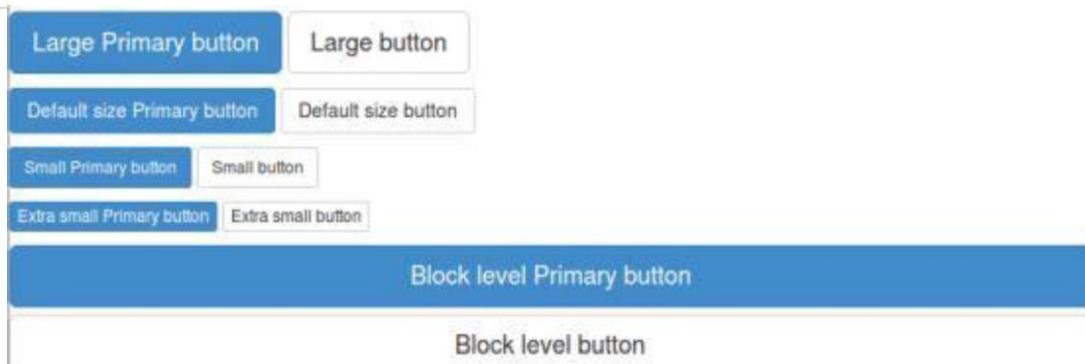
The following example demonstrates this:

```
<p>
  <button type="button" class="btn btn-primary btn-lg">
    Large Primary button
  </button>
  <button type="button" class="btn btn-default btn-lg">
    Large button
  </button>
</p>
<p>
  <button type="button" class="btn btn-primary">
    Default size Primary button
  </button>
  <button type="button" class="btn btn-default">
    Default size button
  </button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-sm">
    Small Primary button
  </button>
  <button type="button" class="btn btn-default btn-sm">
    Small button
  </button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-xs">
    Extra small Primary button
  </button>
  <button type="button" class="btn btn-default btn-xs">
    Extra small button
  </button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-lg btn-block">
    Block level Primary button
  </button>
</p>
```

```

<button type="button" class="btn btn-default btn-lg btn-block">
    Block level button
</button>
</p>

```



## Button State

Bootstrap provides classes which allow you change the state of buttons say active, disabled etc each of these are discussed in the following sections:

### ACTIVE STATE

Buttons will appear pressed (with a darker background, darker border, and inset shadow) when active. The following table summarizes classes used to make button element and anchor element active:

Element	Class
Button element	Use <b>.active</b> class to show that it is activated..
Anchor element	Use <b>.active</b> class to <a> buttons to show that it is activated.

The following example demonstrates this:

```

<p>
    <button type="button" class="btn btn-default btn-lg ">
        Default Button
    </button>
    <button type="button" class="btn btn-default btn-lg active">
        Active Button
    </button>
</p>
<p>
    <button type="button" class="btn btn-primary btn-lg ">
        Primary button
    </button>
    <button type="button" class="btn btn-primary btn-lg active">
        Active Primary button
    </button>
</p>

```



## DISABLED STATE

When you disable a button, it will fade in color by 50%, and lose the gradient.

The following table summarizes classes used to make button element and anchor element disabled:

Element	Class
Button element	Add the <b>disabled</b> attribute to <button> buttons.
Anchor element	Add the <b>disabled</b> class to <a> buttons. <i>Note: This class will only change the &lt;a&gt;'s appearance, not its functionality. You need to use custom JavaScript to disable links here.</i>

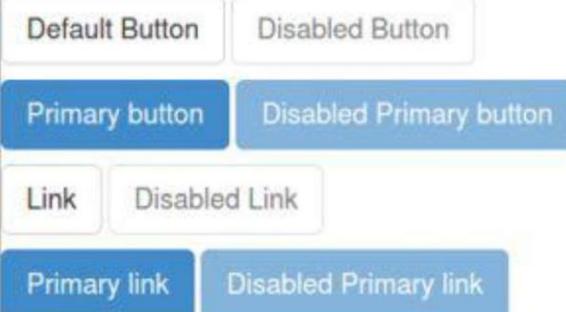
The following example demonstrates this:

```

<p>
  <button type="button" class="btn btn-default btn-lg">
    Default Button
  </button>
  <button type="button" class="btn btn-default btn-lg" disabled="disabled">
    Disabled Button
  </button>
</p>
<p>
  <button type="button" class="btn btn-primary btn-lg ">
    Primary button
  </button>
  <button type="button" class="btn btn-primary btn-lg" disabled="disabled">
    Disabled Primary button
  </button>
</p>
<p>
  <a href="#" class="btn btn-default btn-lg" role="button">
    Link
  </a>
  <a href="#" class="btn btn-default btn-lg disabled" role="button">
    Disabled Link
  </a>
</p>
<p>
  <a href="#" class="btn btn-primary btn-lg" role="button">
    Primary link
  </a>
  <a href="#" class="btn btn-primary btn-lg disabled" role="button">
    Disabled Primary link
  </a>
</p>

```

```
</a>  
</p>
```



## Button Tags

You may use button classes with `<a>`, `<button>`, or `<input>` element. But it is recommended that you use it with `<button>` elements mostly to avoid cross browser inconsistency issues.

The following example demonstrates this:

```
<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">
```



---

# Bootstrap Images

I

n this chapter we will learn Bootstrap support for images. Bootstrap provides three classes that can be used to apply some simple styles to images:

- `.img-rounded`: adds `border-radius:6px` to give the image rounded corners.
- `.img-circle`: makes the entire image round by adding `border-radius:50px`.
- `.img-thumbnail`: adds a bit of padding and a gray border:

The following example demonstrates this:

```



```



---



# Bootstrap Media Object

**T**his chapter discusses about Media object. These are abstract object styles for building various types of

components (like blog comments, Tweets, etc) that feature a left- or right-aligned image alongside textual content. The goal of the media object is to make the code for developing these blocks of information drastically shorter.

The goal of media objects (light markup, easy extendability) is achieved by applying classes to some simple markup. There are two forms to the media object:

- **.media:** This class allows to float a media object (images, video, audio) to the left or right of a content block.
- **.media-list:** If you are preparing a list where the items will be part of an unordered list, use class. Useful for comment threads or articles lists.

Let us see an example below of default media object:

```
<div class="media">
  <a class="pull-left" href="#">
    
  </a>
  <div class="media-body">
    <h4 class="media-heading">Media heading</h4>
    This is some sample text. This is some sample text.
    This is some sample text. This is some sample text.
    This is some sample text. This is some sample text.
    This is some sample text. This is some sample text.
  </div>
</div>
<div class="media">
  <a class="pull-left" href="#">
    
  </a>
  <div class="media-body">
    <h4 class="media-heading">Media heading</h4>
    This is some sample text. This is some sample text.
    This is some sample text. This is some sample text.
    This is some sample text. This is some sample text.
    This is some sample text. This is some sample text.
  </div>
</div>
```

```

        </a>
    <div class="media-body">
        <h4 class="media-heading">Media heading</h4>
        This is some sample text. This is some sample text.
        This is some sample text. This is some sample text.
        This is some sample text. This is some sample text.
        This is some sample text. This is some sample text.
    </div>
</div>
</div>
</div>

```



#### Media heading

This is some sample text. This is some sample text.



#### Media heading

This is some sample text. This is some sample text.



#### Media heading

This is some sample text. This is some sample text.

Let us see an example below of media list:

```

<ul class="media-list">
    <li class="media">
        <a class="pull-left" href="#">
            
        </a>
        <div class="media-body">
            <h4 class="media-heading">Media heading</h4>
            <p>This is some sample text. This is some sample text.
                This is some sample text. This is some sample text.
                This is some sample text. This is some sample text.
                This is some sample text. This is some sample text.</p>
            <!-- Nested media object -->
            <div class="media">
                <a class="pull-left" href="#">
                    
                </a>
                <div class="media-body">
                    <h4 class="media-heading">Nested media heading</h4>
                    This is some sample text. This is some sample text.
                    This is some sample text. This is some sample text.
                    This is some sample text. This is some sample text.
                    This is some sample text. This is some sample text.
                    <!-- Nested media object -->
                    <div class="media">
                        <a class="pull-left" href="#">
                            
                        </a>
                        <div class="media-body">
                            <h4 class="media-heading">Nested media heading</h4>
                            This is some sample text. This is some sample text.
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </li>

```

```

        This is some sample text. This is some sample text.
        This is some sample text. This is some sample text.
        This is some sample text. This is some sample text.
    
```

```

        </div>
    
```

```

        </div>
    
```

```

        </div>
    
```

```

        <!-- Nested media object -->
    
```

```

        <div class="media">
            <a class="pull-left" href="#">
                
            </a>
            <div class="media-body">
                <h4 class="media-heading">Nested media heading</h4>
                This is some sample text. This is some sample text.
                This is some sample text. This is some sample text.
                This is some sample text. This is some sample text.
                This is some sample text. This is some sample text.
            
```

```

                </div>
            
```

```

            </div>
        
```

```

        </div>
    
```

```

    </li>
    <li class="media">
        <a class="pull-right" href="#">
            
        </a>
        <div class="media-body">
            <h4 class="media-heading">Media heading</h4>
            This is some sample text. This is some sample text.
            This is some sample text. This is some sample text.
            This is some sample text. This is some sample text.
            This is some sample text. This is some sample text.
        
```

```

            </div>
        
```

```

    </li>
</ul>

```



### Media heading

This is some sample text. This is some sample text.



#### Nested media heading

This is some sample text. This is some sample text.



#### Nested media heading

This is some sample text. This is some sample text.



#### Nested media heading

This is some sample text. This is some sample text.

### Media heading

This is some sample text. This is some sample text.



# Bootstrap Glyphicons

## T

his chapter will discuss about Glyphicons, its use and some examples. Bootstrap bundles 200 glyphs in font format. First let us understand what are Glyphicons.

## What are Glyphicons?

Glyphicons are icon fonts which you can use in your web projects. [Glyphicons Halflings](#) are not free and require licensing, however their creator has made them available for Bootstrap projects free of cost.

It is recommended, as a thank you, we ask you to include an optional link back to GLYPHICONS whenever practical. — Bootstrap Documentation

## Where to find Glyphicons?

Now that we have downloaded Bootstrap 3.x version and understand its directory structure from chapter [Environment Setup](#), glyphicons can be found within the *fonts* folder. This contains following files:

- glyphicon-halflings-regular.eot
- glyphicon-halflings-regular.svg
- glyphicon-halflings-regular.ttf
- glyphicon-halflings-regular.woff

Associated CSS rules are present within *bootstrap.css* and *bootstrap-min.css* files within css folder of *dist* folder. You can see the available glyphicons at this link : [Glyphicons List](#)

## Usage

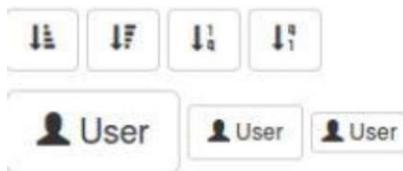
To use the icons, simply use the following code just about anywhere in your code. Leave a space between the icon and text for proper padding.

```
<span class="glyphicon glyphicon-search"></span>
```

Following example demonstrates this:

```
<p>
  <button type="button" class="btn btn-default">
```

```
<span class="glyphicon glyphicon-sort-by-attributes"></span>
</button>
<button type="button" class="btn btn-default">
  <span class="glyphicon glyphicon-sort-by-attributes-alt"></span>
</button>
<button type="button" class="btn btn-default">
  <span class="glyphicon glyphicon-sort-by-order"></span>
</button>
<button type="button" class="btn btn-default">
  <span class="glyphicon glyphicon-sort-by-order-alt"></span>
</button>
</p>
<button type="button" class="btn btn-default btn-lg">
  <span class="glyphicon glyphicon-user"></span> User
</button>
<button type="button" class="btn btn-default btn-sm">
  <span class="glyphicon glyphicon-user"></span> User
</button>
<button type="button" class="btn btn-default btn-xs">
  <span class="glyphicon glyphicon-user"></span> User
</button>
```



---

# Bootstrap Badges

**T**

his chapter will discuss about Bootstrap badges. Badges are similar to labels; the primary difference is that the corners are more rounded.

Badges are mainly used to highlight new or unread items. To use badges just add `<span class="badge">` to links, Bootstrap navs, and more.

The following example demonstrates this:

```
<a href="#">Mailbox <span class="badge">50</span></a>
```



When there are no new or unread items, badges will simply collapse via CSS's `:empty` selector provided no content exists within.

## Active nav states

You can place badges in active states of pill and list navigations. You can achieve this by placing `<span class="badge">` to active links, as demonstrated in the following example:

```
<h4>Example for Active State in Pill </h4>
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home <span class="badge">42</span></a></li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages <span class="badge">3</span></a></li>
</ul>
<br>
<h4>Example for Active State in navigations</h4>
<ul class="nav nav-pills nav-stacked" style="max-width: 260px;">
    <li class="active">
        <a href="#">
            <span class="badge pull-right">42</span>
            Home
        </a>
    </li>
    <li><a href="#">Profile</a></li>
    <li>
        <a href="#">
            <span class="badge pull-right">3</span>
            Messages
        </a>
    </li>
</ul>
```

```
</a>
</li>
</ul>
```

#### Example for Active State in Pill



#### Example for Active State in navigations



---

# Bootstrap Jumbotron

**T**

his chapter will discuss one more feature Bootstrap supports, the Jumbotron. As the name suggest this component can optionally increases the size of headings and add a lot of margin for landing page content. To use the Jumbotron:

- simply create a container `<div>` with the class of `.jumbotron`.
- In addition to a larger `<h1>`, the `font-weight` is reduced to 200px.

The following example demonstrates this:

```
<div class="container">
  <div class="jumbotron">
    <h1>Welcome to landing page!</h1>
    <p>This is an example for jumbotron.</p>
    <p><a class="btn btn-primary btn-lg" role="button">
      Learn more</a>
    </p>
  </div>
</div>
```



Welcome to landing  
page!

This is an example for jumbotron.

Learn more

To get a jumbotron full width, and without rounded corners use the `.jumbotron` class outside all `.container` classes and instead add a `.container` within as shown in the following example:

```
<div class="jumbotron">
  <div class="container">
    <h1>Welcome to landing page!</h1>
    <p>This is an example for jumbotron.</p>
    <p><a class="btn btn-primary btn-lg" role="button">
      Learn more</a>
    </p>
  </div>
</div>
```

# Welcome to landing page!

This is an example for jumbotron.

[Learn more](#)

# Bootstrap Affix Plugin

**T**

he affix plugin allows a `<div>` to become affixed to a location on the page. You can also toggle it's pinning on and off using this plugin. A common example of this is social icons. They will start in a location, but as the page hits a certain mark, the `<div>` will become locked in place and will stop scrolling with the rest of the page.

If you want to include this plugin functionality individually, then you will need [affix.js](#). Else, as mentioned in the chapter [Bootstrap Plugins Overview](#), you can include `bootstrap.js` or the minified `bootstrap.min.js`.

## USAGE

You can use the affix plugin via data attributes or manually with your own JavaScript as discussed below.

- **Via data attributes:** To easily add affix behavior to any element, just add `data-spy="affix"` to the element you want to spy on. Use offsets to define when to toggle the pinning of an element.

## EXAMPLE

The following example demonstrates the usage through data attributes:

```
<div class="container">
  <div class="jumbotron">
    <h1>Bootstrap Affix Plugin example</h1>
  </div>
  <div id="myNav" data-spy="affix" data-offset-top="60"
       data-offset-bottom="200">
    <div class="col-md-3">
      <ul class="nav nav-tabs nav-stacked affix" data-spy="affix"
          data-offset-top="190">
        <li class="active"><a href="#one">Tutorial One</a></li>
        <li><a href="#two">Tutorial Two</a></li>
        <li><a href="#three">Tutorial Three</a></li>
      </ul>
    </div>
    <div class="col-md-9">
      <h2 id="one">Tutorial One</h2>
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
      Nam eu sem tempor, varius quam at, luctus dui. Mauris magna
      metus, dapibus nec turpis vel, semper malesuada ante.
      Vestibulum id metus ac nisl bibendum scelerisque non non
      purus. Suspendisse varius nibh non aliquet sagittis. In
      tincidunt orci sit amet elementum vestibulum. Vivamus
      fermentum in arcu in aliquam. Quisque aliquam porta odio
```

in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

<hr>

<h2 id="two">Tutorial Two</h2>

<p> Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.</p>

<p>Vestibulum consectetur scelerisque lacus, ac fermentum lorem convallis sed. Nam odio tortor, dictum quis malesuada at, pellentesque vitae orci. Vivamus elementum, felis eu auctor lobortis, diam velit egestas lacus, quis fermentum metus ante quis urna. Sed at facilisis libero. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum bibendum blandit dolor. Nunc orci dolor, molestie nec nibh in, hendrerit tincidunt ante. Vivamus sem augue, hendrerit non sapien in, mollis ornare augue.</p>

<hr>

<h2 id="three">Tutorial Three</h2>

<p>Integer pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id, tincidunt sit amet mauris. </p>

<p>Phasellus vitae suscipit justo. Mauris pharetra feugiat ante id lacinia. Etiam faucibus mauris id tempor egestas. Duis luctus turpis at accumsan tincidunt. Phasellus risus risus, volutpat vel tellus ac, tincidunt fringilla massa. Etiam hendrerit dolor eget ante rutrum adipiscing. Cras interdum ipsum mattis, tempus mauris vel, semper ipsum. Duis sed dolor ut enim lobortis pellentesque ultricies ac ligula. Pellentesque convallis elit nisi, id vulputate ipsum ullamcorper ut. Cras ac pulvinar purus, ac viverra est. Suspendisse potenti. Integer pellentesque neque et elementum tempus. Curabitur bibendum in ligula ut rhoncus.</p>

<p>Quisque pharetra velit id velit iaculis pretium. Nullam a justo sed ligula porta semper eu quis enim. Pellentesque

```

        pellentesque, metus at facilisis hendrerit, lectus velit
        facilisis leo, quis volutpat turpis arcu quis enim. Nulla
        viverra lorem elementum interdum ultricies. Suspendisse
        accumsan quam nec ante mollis tempus. Morbi vel accumsan diam,
        eget convallis tellus. Suspendisse potenti.</p>
    </div>
</div>
</div>

```

## Bootstrap Affix Plugin example

Tutorial One

Tutorial Two

Tutorial Three

### Tutorial One

Nullam ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam ac, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In incidunt orci sit amet elementum vestibulum, Vivamus fermentum in aro in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisi leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faubibus ligula.

Vestibulum quis quam ut magna consequat faubibus. Pellentesque eget nisi a mi suscipit incidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit et laus blandit, commodo accuis justo viverra. Etiam vitae est arcu, Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac du quis, bibendum viverra erat. Maecenas mattis lectus enim, quis incidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.

### Tutorial Two

Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra incidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed

- **Via JavaScript:** You can affix an element manually with JavaScript as below:

```

$( '#myAffix' ).affix({
    offset: {
        top: 100, bottom: function () {
            return (this.bottom =
                $('.bs-footer').outerHeight(true))
        }
    }
})

```

## EXAMPLE

The following example demonstrates the usage through data attributes:

```

<div class="container">
    <div class="jumbotron">
        <h1>Bootstrap Affix Plugin example</h1>
    </div>
    <div>
        <div class="col-md-3">
            <ul class="nav nav-tabs nav-stacked affix" id="myNav">
                <li class="active"><a href="#one">Tutorial One</a></li>
                <li><a href="#two">Tutorial Two</a></li>
                <li><a href="#three">Tutorial Three</a></li>
            </ul>
        </div>
        <div class="col-md-9">
            <h2 id="one">Tutorial One</h2>

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

<hr>

<h2 id="two">Tutorial Two</h2>

<p> Nullam hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci. In dapibus velit blandit pharetra tincidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed consequat.</p>

<p>Vestibulum consectetur scelerisque lacus, ac fermentum lorem convallis sed. Nam odio tortor, dictum quis malesuada at, pellentesque vitae orci. Vivamus elementum, felis eu auctor lobortis, diam velit egestas lacus, quis fermentum metus ante quis urna. Sed at facilisis libero. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum bibendum blandit dolor. Nunc orci dolor, molestie nec nibh in, hendrerit tincidunt ante. Vivamus sem augue, hendrerit non sapien in, mollis ornare augue.</p>

<hr>

<h2 id="three">Tutorial Three</h2>

<p>Integer pulvinar leo id risus pellentesque vestibulum. Sed diam libero, sodales eget sapien vel, porttitor bibendum enim. Donec sed nibh vitae lorem porttitor blandit in nec ante. Pellentesque vitae metus ipsum. Phasellus sed nunc ac sem malesuada condimentum. Etiam in aliquam lectus. Nam vel sapien diam. Donec pharetra id arcu eget blandit. Proin imperdiet mattis augue in porttitor. Quisque tempus enim id lobortis feugiat. Suspendisse tincidunt risus quis dolor fringilla blandit. Ut sed sapien at purus lacinia porttitor. Nullam iaculis, felis a pretium ornare, dolor nisl semper tortor, vel sagittis lacus est consequat eros. Sed id pretium nisl. Curabitur dolor nisl, laoreet vitae aliquam id, tincidunt sit amet mauris. </p>

<p>Phasellus vitae suscipit justo. Mauris pharetra feugiat ante id lacinia. Etiam faucibus mauris id tempor egestas. Duis luctus turpis at accumsan tincidunt. Phasellus risus risus, volutpat vel tellus ac, tincidunt fringilla massa. Etiam hendrerit dolor eget ante rutrum adipiscing. Cras interdum ipsum mattis, tempus mauris vel, semper ipsum. Duis sed dolor

```

ut enim lobortis pellentesque ultricies ac ligula. Pellentesque
convallis elit nisi, id vulputate ipsum ullamcorper ut. Cras
ac pulvinar purus, ac viverra est. Suspendisse potenti. Integer
pellentesque neque et elementum tempus. Curabitur bibendum in
ligula ut rhoncus.</p>
<p>Quisque pharetra velit id velit iaculis pretium. Nullam a
justo sed ligula porta semper eu quis enim. Pellentesque
pellentesque, metus at facilisis hendrerit, lectus velit
facilisis leo, quis volutpat turpis arcu quis enim. Nulla
viverra lorem elementum interdum ultricies. Suspendisse
accumsan quam nec ante mollis tempus. Morbi vel accumsan diam,
egest convallis tellus. Suspendisse potenti.</p>
</div>
</div>
</div>
<script type="text/javascript">
$(function () {
  $('#myNav').affix({
    offset: {
      top: 60
    }
  });
});
</script>

```

## Bootstrap Affix Plugin example

[Tutorial One](#)

[Tutorial Two](#)

[Tutorial Three](#)

### Tutorial One

Loreum ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In incidunt orci sit amet elementum vestibulum. Vivamus fermentum in aro in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.

Vestibulum quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit incidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor vel et lacus blandit, commodo accumsan justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis incidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit felis.

### Tutorial Two

Nulum hendrerit justo non leo aliquet imperdiet. Etiam in sagittis lectus. Suspendisse ultrices placerat accumsan. Mauris quis dapibus orci, in dapibus velit blandit pharetra incidunt. Quisque non sapien nec lacus condimentum facilisis ut iaculis enim. Sed viverra interdum bibendum. Donec ac sollicitudin dolor. Sed fringilla vitae lacus at rutrum. Phasellus congue vestibulum ligula sed

## Positioning via CSS

In both the above situations, you must provide CSS for the positioning of your content. The affix plugin toggles between three classes, each representing a particular state: `.affix`, `.affix-top`, and `.affix-bottom`. Follow the below steps to set your CSS for either of the above usage options.

- To start, the plugin adds `.affix-top` to indicate the element is in its top-most position. At this point no CSS positioning is required.
- Scrolling past the element you want affixed should trigger the actual affixing. This is where `.affix` replaces `.affix-top` and sets `position: fixed;` (provided by Bootstrap's core CSS).
- If a bottom offset is defined, scrolling past that should replace `.affix` with `.affix-bottom`. Since offsets are optional, setting one requires you to set the appropriate CSS. In this case, add`position: absolute;` when necessary.

# Options

There are certain options which can be passed via data attributes or JavaScript are listed in the following table:

Option Name	Type/Default Value	Data attribute name	Description
offset	number   function   object <i>Default: 10</i>	data-offset	Pixels to offset from screen when calculating position of scroll. If a single number is provided, the offset will be applied in both top and bottom directions. To provide a unique, bottom and top offset just provide an object offset: { top: 10 } or offset: { top: 10, bottom: 5 }. Use a function when you need to dynamically calculate an offset.

---