



Bootstrap Tutorials

Contents

- Introduction to Bootstrap Framework
- What is Bootstrap?
- Why Bootstrap?
- How to get Bootstrap Framework and Setup Bootstrap Environment
- Bootstrap Containers
- Bootstrap Grids
- Default Settings using Bootstrap
- Tables using Bootstrap
- Image Shapes using Bootstrap
- Bootstrap Page Headers
- Buttons using Bootstrap
- Bootstrap Pagination
- Glyph icons using Bootstrap
- Bootstrap Forms
- Extending Form Controls
- Validation States on Form controls
- Bootstrap Carousel Plugin
- Tabs using Bootstrap
- Navigation Bars using Bootstrap
- Tooltips
- Popovers
- Alerts
- Modals

Introduction

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.

Bootstrap is a sleek, intuitive, and powerful front-end framework for faster and easier web development.

What is Bootstrap?

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs

Why Bootstrap?

Advantages of Bootstrap:

- **Save lots of time** — you can save lots of time and efforts using the Bootstrap predefined design templates and classes and concentrate on other development work.
- **Easy to use:** Anybody with just basic knowledge of HTML and CSS can start using Bootstrap
- **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops
- **Mobile-first approach:** In Bootstrap 3, mobile-first styles are part of the core framework
- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Safari, and Opera)
- **Open Source** — And the best part is, it is completely free to download and use

Where to Get Bootstrap?

There are two ways to start using Bootstrap on your own web site.

You can:

- Download Bootstrap from getbootstrap.com
- Include Bootstrap from a CDN

Downloading Bootstrap

If you want to download and host Bootstrap yourself, go to getbootstrap.com, and follow the instructions there.

Understanding the File Structure

Once downloaded the compiled Bootstrap, unzip the compressed folder to see the structure. You'll find the following file structure and contents.

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

Bootstrap CDN

If you don't want to download and host Bootstrap yourself, you can include it from a CDN (Content Delivery Network).

MaxCDN provide CDN support for Bootstrap's CSS and JavaScript. Also include jQuery:

```
!-- Latest compiled and minified CSS -->
<link rel="stylesheet"href="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0
/css/bootstrap.min.css">

<!-- jQuery library -->
<scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min
.js"></script>

<!-- Latest compiled JavaScript -->
<scriptsrc="http://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.mi
n.js"></script>
```

Setup Bootstrap page:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Basic HTML File</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
  <h1>Hello, world!</h1>
</body>
</html>
```

Making this HTML File a Bootstrapped Template

To make this HTML file a Bootstrapped template, just include the appropriate Bootstrap CSS and JS files

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Basic Bootstrap Template</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
  <script src="http://code.jquery.com/jquery.min.js"></script>
  <script src="js/bootstrap.min.js"></script>

</head>
<body>
  <h1>Hello, world!</h1>

</body>
</html>
```

Mobile first

To ensure proper rendering and touch zooming, **add the viewport meta tag** to your <head>.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

You can disable zooming capabilities on mobile devices by adding user-scalable=no to the viewport meta tag. This disables zooming, meaning users are only able to scroll, and results in your site feeling a bit more like a native application.

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

Bootstrap Containers

Bootstrap also requires a containing element to wrap site contents.

There are two container classes to choose from:

1. The `.container` class provides a responsive **fixed width container**
2. The `.container-fluid` class provides a **full width container**, spanning the entire width of the viewport

Note: Containers are not nestable (you cannot put a container inside another container).

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>CSS BootStrapDemo</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="css/bootstrap.min.css" />
  <script src="js/jquery-1.8.2.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
</head>
<body>
  <div class="container">
    <h1>Hello..Bootstrap</h1>
    <h2>Welcome Responsive Web pages</h2>
  </div>
</body>
</html>
```

Bootstrap Grids

Bootstrap's grid system allows up to 12 columns across the page.

If you do not want to use all 12 column individually, you can group the columns together to create wider columns:

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

The Bootstrap grid system has four classes:

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


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

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

```
    ...
```

```
  </div>
```

```
</div>
```

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

```
  <div class="col-*-*"></div>
```

```
</div>
```

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

```
  <div class="col-*-*"></div>
```

```
  <div class="col-*-*"></div>
```

```
  <div class="col-*-*"></div>
```

```
</div>
```

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

```
  ...
```

```
</div>
```

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

```
    <div class="col-sm-4">Col1</div>
```

```
    <div class="col-sm-4">Col2</div>
```

```
    <div class="col-sm-4">Col3</div>
```

```
  </div>
```

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

```
  <div class="col-sm-4">.col-sm-4</div>
```

```
  <div class="col-sm-8">.col-sm-8</div>
```

```
</div>
```

Example: Mobile and desktop

Don't want your columns to simply stack in smaller devices? Use the extra small and medium device grid classes by adding `.col-xs-*` `.col-md-*` to your columns.

```
<!-- Stack the columns on mobile by making one full-width and the other half-width -->
```

```
<div class="row">  
  <div class="col-xs-12 col-md-8">.col-xs-12 .col-md-8</div>  
  <div class="col-xs-6 col-md-4">.col-xs-6 .col-md-4</div>  
</div>
```

```
<!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
```

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

```
<!-- Columns are always 50% wide, on mobile and desktop -->
```

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

Example: Mobile, tablet, desktop

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

```
<div class="row">

<div class="col-xs-12 col-sm-6 col-md-8">.col-xs-12 .col-sm-6 .col-md-8</div>

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

<div class="row">

    <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>

    <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>

    <div class="col-xs-6 col-sm-4">.col-xs-6 .col-sm-4</div>
</div>
```

Bootstrap's Default Settings

Bootstrap's global default font-size is 14px, with a line-height of 1.428.

This is applied to the `<body>` and all paragraphs.

In addition, all `<p>` elements have a bottom margin that equals half their computed line-height (10px by default).

Bootstrap vs. Browser Defaults

`<small>`

In Bootstrap the HTML `<small>` element is used to create a lighter, secondary text in any heading as below:

h1 heading secondary text

`<abbr>`

Bootstrap will style the HTML `<abbr>` element in the following way:

WHO

<dl>

Bootstrap will style the HTML `<dl>` element in the following way:

Coffee

- black hot drink

Milk

- white cold drink

<mark>

Bootstrap will style the HTML `<mark>` element

Highlight

Contextual Colors and Backgrounds

Bootstrap also has some contextual classes that can be used to provide "meaning through colors".

The classes for text colors are: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, and `.text-danger`:

The classes for background colors are: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, and `.bg-danger`:

Bootstrap Basic Table

A basic Bootstrap table has a light padding and only horizontal dividers.

The `.table` class adds basic styling to a table

Striped Rows

The `.table-striped` class adds zebra-stripes to a table

Bordered Table

The `.table-bordered` class adds borders on all sides of the table and cells

Hover Rows

The `.table-hover` class enables a hover state on table rows

Condensed Table

The `.table-condensed` class makes a table more compact by cutting cell padding in half

You can also create Responsive tables:

```
<div class="table-responsive">  
  <table class="table">  
    </table>  
</div>
```

Contextual Classes

Contextual classes can be used to color table rows (`<tr>`) or table cells (`<td>`)

The contextual classes that can be used are:

Class	Description
<code>.active</code>	Applies the hover color to the table row or table cell
<code>.success</code>	Indicates a successful or positive action
<code>.info</code>	Indicates a neutral informative change or action
<code>.warning</code>	Indicates a warning that might need attention
<code>.danger</code>	Indicates a dangerous or potentially negative action

Bootstrap Image Shapes

Rounded Corners

The `.img-rounded` class adds rounded corners to an image

```

```

Circle

The `.img-circle` class shapes the image to a circle

```

```

Thumbnail

The `.img-thumbnail` class shapes the image to a thumbnail

```

```


Responsive Images

Images comes in all sizes. So do screens. Responsive images automatically adjust to fit the size of the screen.

Create responsive images by adding an `.img-responsive` class to the `` tag. The image will then scale nicely to the parent element.

The `.img-responsive` class applies `max-width: 100%;` and `height: auto;` to the image

```

```

Bootstrap Jumbotron and Page Header

A jumbotron indicates a big box for calling extra attention to some special content or information.

A jumbotron is displayed as a grey box with rounded corners. It also enlarges the font sizes of the text inside it.

Use a `<div>` element with class `.jumbotron` to create a jumbotron

```
<div class="container">
  <div class="jumbotron">
    <h1>Bootstrap Tutorial</h1>
    <p>Bootstrap is the most popular HTML, CSS, and JS framework for developing
      responsive, mobile-first projects on the web.</p>
  </div>
  <p>This is some text.</p>
  <p>This is another text.</p>
</div>
```

Creating a Page Header

A page header is like a section divider.

The `.page-header` class adds a horizontal line under the heading (+ adds some extra space around the element):

```
<div class="page-header">
  <h1>Example Page Header</h1>
</div>
```

Bootstrap Buttons

Button Styles

Bootstrap provides seven styles of buttons:

- `.btn-default`
- `.btn-primary`
- `.btn-success`
- `.btn-info`
- `.btn-warning`
- `.btn-danger`
- `.btn-link`

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

The button classes can be used on an `<a>`, `<button>`, or `<input>` element:

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

Button Sizes

Bootstrap provides four button sizes

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

Block Level Buttons

A block level button spans the entire width of the parent element.

Add class `.btn-block` to create a block level button:

```
<button type="button" class="btn btn-primary btn-block">Button 1</button>
```

Active/Disabled Buttons

A button can be set to an active (appear pressed) or a disabled (unclickable) state:

The class `.active` makes a button appear pressed, and the class `.disabled` makes a button unclickable:

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

Bootstrap Button Groups

Button Groups

Bootstrap allows you to group a series of buttons together (on a single line) in a button group:

```
<div class="btn-group">  
  <button type="button" class="btn btn-primary">Apple</button>  
  <button type="button" class="btn btn-primary">Samsung</button>  
  <button type="button" class="btn btn-primary">Sony</button>  
</div>
```

Vertical Button Groups

Bootstrap also supports vertical button groups:

```
<div class="btn-group-vertical">  
  <button type="button" class="btn btn-primary">Apple</button>  
  <button type="button" class="btn btn-primary">Samsung</button>  
  <button type="button" class="btn btn-primary">Sony</button>  
</div>
```

Nesting Button Groups & Drop Down Menus

Nest button groups to create drop down menus:

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Apple</button>
  <button type="button" class="btn btn-primary">Samsung</button>
  <div class="btn-group">
    <button type="button" class="btn btn-primary dropdown-toggle" data-
toggle="dropdown">
      Sony <span class="caret"></span></button>
    <ul class="dropdown-menu" role="menu">
      <li><a href="#">Tablet</a></li>
      <li><a href="#">Smartphone</a></li>
    </ul>
  </div>
</div>
```

Bootstrap Pagination

Basic Pagination

If you have a web site with lots of pages, you may wish to add some sort of pagination to each page.

A basic pagination in Bootstrap looks like this:

```
<ul class="pagination">
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
  <li><a href="#">4</a></li>
  <li><a href="#">5</a></li>
</ul>
```

Active State

The active state shows what is the current page

Add class `.active` to let the user know which page he/she is on:

Disabled State

A disabled link cannot be clicked

Add class `.disabled` if a link for some reason is disabled:

Add class `.pagination-lg` for larger blocks or `.pagination-sm` for smaller blocks:

Bootstrap Pager

What is Pager?

Pager is also a form of pagination.

Pager provides previous and next buttons (links).

To create previous/next buttons, add the `.pager` class to an `` element:

```
<ul class="pager">
  <li><a href="#">Previous</a></li>
  <li><a href="#">Next</a></li>
</ul>
```

Align Buttons

Use the `.previous` and `.next` classes to align each button to the sides of the page:

```
ul class="pager">
  <li class="previous"><a href="#">Previous</a></li>
  <li class="next"><a href="#">Next</a></li>
</ul>
```



Bootstrap Glyphicons

Bootstrap provides 200 glyphs from the [Glyphicons](#) Halflings set.


Glyphicons can be used in text, buttons, toolbars, navigation, forms, etc.

Here are some examples of glyphs:

Envelope glyph: 

Print glyph: 

Search glyph: 

Download glyph: 

Glyphicon Syntax

A glyph is inserted with the following syntax:

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

```
<p>Envelope icon: <span class="glyphicon glyphicon-envelope"></span></p>
```

```
<p>Envelope icon as a link:
```

```
  <a href="#"><span class="glyphicon glyphicon-envelope"></span></a>
</p>
```

```
<p>Search icon: <span class="glyphicon glyphicon-search"></span></p>
```

```
<p>Search icon on a button:
```

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

Bootstrap Forms

Bootstrap's Default Settings

Form controls automatically receive some global styling with Bootstrap:

All textual `<input>`, `<textarea>`, and `<select>` elements with class `.form-control` have a width of 100%.

Bootstrap Form Layouts

Bootstrap provides three types of form layouts:

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

Bootstrap Vertical Form (default)

The following example creates a vertical form with two input fields, one checkbox, and a submit button:

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

Bootstrap Inline Form

In an inline form, all of the elements are inline, left-aligned, and the labels are alongside.

Note: This only applies to forms within viewports that are at least 768px wide!

Additional rule for an inline form:

- Add class `.form-inline` to the `<form>` element

```
<form class="form-inline" role="form">
  <div class="form-group">
    <label for="email">Email address:</label>
    <input type="email" class="form-control" id="email">
  </div>
  <div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" id="pwd">
  </div>
  <div class="checkbox">
    <label><input type="checkbox"> Remember me</label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Bootstrap Horizontal Form

A horizontal form stands apart from the other forms both in the amount of markup, and in the presentation of the form.

Additional rules for a horizontal form:

- Add class `.form-horizontal` to the `<form>` element
- Add class `.control-label` to all `<label>` elements

```
<form class="form-horizontal" role="form">
  <div class="form-group">
    <label class="control-label col-sm-2" for="email">Email:</label>
    <div class="col-sm-10">
      <input type="email" class="form-
control" id="email" placeholder="Enter email">
    </div>
  </div>
  <div class="form-group">
    <label class="control-label col-sm-2" for="pwd">Password:</label>
    <div class="col-sm-10">
      <input type="password" class="form-
control" id="pwd" placeholder="Enter password">
    </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">Submit</button>
    </div>
  </div>
</form>
```

Search form

```
<form class="form-search">

  <input type="text" class="input-medium search-query">

  <button type="submit" class="btn">Search</button>

</form>
```

SignIn Form:

```
<div class="container">

  <form class="form-signin">
    <h2 class="form-signin-heading">Please sign in</h2>
    <label for="inputEmail" class="sr-only">Email address</label>
    <input type="email" id="inputEmail" class="form-control" placeholder="Email
address" required autofocus>
    <label for="inputPassword" class="sr-only">Password</label>
    <input type="password" id="inputPassword" class="form-control"
placeholder="Password" required>
    <div class="checkbox">
      <label>
        <input type="checkbox" value="remember-me"> Remember me
      </label>
    </div>
    <button class="btn btn-lg btn-primary btn-block" type="submit">Sign in</button>
  </form>

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

Extending form controls

Adding on top of existing browser controls, Bootstrap includes other useful form components.

Prepended and appended inputs

Add text or buttons before or after any text-based input. Do note that `select` elements are not supported here.

Default options

Wrap an `.add-on` and an `input` with one of two classes to prepend or append text to an input.

```
<div class="input-prepend">

<span class="add-on">@</span>

<input class="span2" id="prependedInput" type="text" placeholder="Username">

</div>

<div class="input-append">

<input class="span2" id="appendedInput" type="text">

<span class="add-on">.00</span>

</div>
```

Combined

Use both classes and two instances of `.add-on` to prepend and append an input

```
<div class="input-prepend input-append">

<span class="add-on">${</span>

<input class="span2" id="appendedPrependedInput" type="text">

<span class="add-on">.00</span>

</div>
```

Validation states

Bootstrap includes validation styles for error, warning, info, and success messages. To use, add the appropriate class to the surrounding `.control-group`.

```
<div class="control-group warning">

<label class="control-label" for="inputWarning">Input with warning</label>

<div class="controls">

<input type="text" id="inputWarning">

<span class="help-inline">Something may have gone wrong</span>

</div>

</div>
```

```
<div class="control-group error">

<label class="control-label" for="inputError">Input with error</label>

<div class="controls">

<input type="text" id="inputError">

<span class="help-inline">Please correct the error</span>

</div>

</div>
```

```
<div class="control-group info">

<label class="control-label" for="inputInfo">Input with info</label>

<div class="controls">
```



```
<input type="text" id="inputInfo">

<span class="help-inline">Username is already taken</span>

</div>

</div>
```

```
<div class="control-group success">

<label class="control-label" for="inputSuccess">Input with success</label>

<div class="controls">

<input type="text" id="inputSuccess">

<span class="help-inline">Woohoo!</span>

</div>

</div>
```

Bootstrap Carousel Plugin

The Carousel Plugin

The Carousel plugin is a component for cycling through elements, like a carousel (slideshow).

This plugin is included in "bootstrap.js" (or in the minified "bootstrap.min.js" file)

How to Create a Carousel

The following example shows how to create a basic carousel:

```
<div id="myCarousel" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
    <li data-target="#myCarousel" data-slide-to="3"></li>
  </ol>

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

    <div class="item">
      
    </div>

    <div class="item">
      
    </div>

    <div class="item">
```

```
        
    </div>
</div>

<!-- Left and right controls -->
<a class="left carousel-control" href="#myCarousel" role="button" data-
slide="prev">
    <span class="glyphicon glyphicon-chevron-left" aria-
hidden="true"></span>
    <span class="sr-only">Previous</span>
</a>
<a class="right carousel-control" href="#myCarousel" role="button" data-
slide="next">
    <span class="glyphicon glyphicon-chevron-right" aria-
hidden="true"></span>
    <span class="sr-only">Next</span>
</a>
</div>
```

Tabs

Tabs are created with `<ul class="nav nav-tabs">`.

Tip: Also mark the current page with `<li class="active">`.

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#">Home</a></li>
  <li><a href="#">Menu 1</a></li>
  <li><a href="#">Menu 2</a></li>
  <li><a href="#">Menu 3</a></li>
</ul>
```

Tabs With Dropdown Menu

Tabs can also hold dropdown menus.

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#">Home</a></li>
  <li class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#">Menu 1
    <span class="caret"></span></a>
    <ul class="dropdown-menu">
      <li><a href="#">Submenu 1-1</a></li>
      <li><a href="#">Submenu 1-2</a></li>
      <li><a href="#">Submenu 1-3</a></li>
    </ul>
  </li>
  <li><a href="#">Menu 2</a></li>
  <li><a href="#">Menu 3</a></li>
</ul>
```

Navigation Bars

Navigation bar is a navigation header that is placed at the top of the page.

With Bootstrap, a navigation bar can extend or collapse, depending on the screen size.

A standard navigation bar is created with `<nav class="navbar navbar-default">`.

```
<body>
<nav class="navbar navbar-default">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <div>
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Page 1</a></li>
        <li><a href="#">Page 2</a></li>
        <li><a href="#">Page 3</a></li>
      </ul>
    </div>
  </div>
</nav>
...
```

Inverted Navigation Bar

If you don't like the style of the default navigation bar, Bootstrap provides an alternative.

Just change the `.navbar-default` class into `.navbar-inverse`

Fixed Navigation Bar

The navigation bar can also be fixed at the top or at the bottom of the page.

A fixed navigation bar stays visible in a fixed position (top or bottom) independent of the page scroll.

The `.navbar-fixed-top` class makes the navigation bar fixed at the top:

The `.navbar-fixed-bottom` class makes the navigation bar fixed at the bottom:

Navigation Bar With Dropdown

Navigation bars can also hold dropdown menus.

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <div>
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li class="dropdown">
          <a class="dropdown-toggle" data-toggle="dropdown" href="#">Page 1
            <span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="#">Page 1-1</a></li>
            <li><a href="#">Page 1-2</a></li>
            <li><a href="#">Page 1-3</a></li>
          </ul>
        </li>
        <li><a href="#">Page 2</a></li>
        <li><a href="#">Page 3</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Right-Aligned Navigation Bar

The `.navbar-right` class is used to right-align navigation bar buttons.

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <div>
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Page 1</a></li>
        <li><a href="#">Page 2</a></li>
        <li><a href="#">Page 3</a></li>
      </ul>
      <ul class="nav navbar-nav navbar-right">
        <li><a href="#"><span class="glyphicon glyphicon-
user"></span> Sign Up</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-log-
in"></span> Login</a></li>
      </ul>
    </div>
  </div>
</nav>
```

Collapsing the Navigation Bar

The navigation bar takes up too much space on a small screen.

We should hide the navigation bar; and only show it when it is needed.

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#myNavbar">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">WebSiteName</a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Page 1</a></li>
        <li><a href="#">Page 2</a></li>
        <li><a href="#">Page 3</a></li>
      </ul>
      <ul class="nav navbar-nav navbar-right">
        <li><a href="#"><span class="glyphicon glyphicon-
user"></span> Sign Up</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-log-
in"></span> Login</a></li>
      </ul>
    </div>
  </div>
</nav>
```


Bootstrap Tooltips

A tooltip is a small pop up that appears when user places the mouse pointer over an element such as link or buttons to provide hint or information about the element being hovered.

Triggering the Tooltips

Tooltips can be triggered via JavaScript — just call the `tooltip()` Bootstrap method with the "id" or "class" [selector](#) of the target element in your JavaScript code

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

```
<a href="#" data-toggle="tooltip" data-placement="top" data-original-title="Default tooltip">Tooltip</a>
```

Positioning of Tooltips via JavaScript

```
$(document).ready(function(){
    $(".tip-top").tooltip({placement : 'top'});
    $(".tip-right").tooltip({placement : 'right'});
    $(".tip-bottom").tooltip({placement : 'bottom'});
    $(".tip-left").tooltip({ placement : 'left'});
});
```

Bootstrap Popovers

Popover is a small overlay of content that is used to display secondary information of any element when it is clicked by a user, like those on the iPad.

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

<button type="button" class="btn btn-primary" data-toggle="popover" data-
placement="top" title="Popover title" data-content="Default popover">Popover on
top</button>
```

Bootstrap Alerts

Warning Alerts

```
<div class="alert alert-warning">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Warning!</strong> There was a problem with your network connection.
</div>
```

Error or Danger Alerts

```
<div class="alert alert-error">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Error!</strong> A problem has been occurred while submitting your data.
</div>
```

Success or Confirmation Alerts

```
<div class="alert alert-success">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Success!</strong> Your message has been sent successfully.
</div>
```

Bootstrap Modals

Modals are basically a dialog box that is used to provide important information to the user or prompt user to take necessary actions before moving on.

Modal windows are widely used to warn users for situations like session time out or to receive their final confirmation before going to perform any critical actions such as saving or deleting important data.

```
<script type="text/javascript">
    $(document).ready(function () {
        $("#myModal").modal('show');
    });
</script>
</head>
<body>
<div id="myModal" class="modal fade">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">&times;</button>
                <h4 class="modal-title">Confirmation</h4>
            </div>
            <div class="modal-body">
                <p>Do you want to save changes you made to document before closing?</p>
                <p class="text-warning"><small>If you don't save, your changes will be
lost.</small></p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
                <button type="button" class="btn btn-primary">Save changes</button>
            </div>
        </div>
    </div>
</div>
</body>
</html>
```

Activate Modals via Data Attributes

You can activate a Bootstrap modal window by clicking on the button or link without writing any JavaScript code via data attributes.

<!-- Button HTML (to Trigger Modal) -->

```
<a href="#myModal" class="btn btn-lg btn-primary" data-toggle="modal">Launch Demo Modal</a>
```