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

Botstrap History

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter, and released as an open source product in August 2011 on GitHub.

**In June 2014 Bootstrap was the No.1 project on GitHub!**

Why Use Bootstrap?

Advantages of Bootstrap:

* **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)

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](http://getbootstrap.com/getting-started/), and follow the instructions there.

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/3.3.4/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/3.3.4/js/bootstrap.min.js"></script>

Create First Web Page With Bootstrap

**1. Add the HTML5 doctype**

Bootstrap uses HTML elements and CSS properties that require the HTML5 doctype.

Always include the HTML5 doctype at the beginning of the page, along with the lang attribute and the correct character set:

<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">   
  </head>  
</html>

**2. Bootstrap 3 is mobile-first**

 Bootstrap 3 is designed to be responsive to mobile devices. Mobile-first styles are part of the core framework.

To ensure proper rendering and touch zooming, add the following <meta> tag inside the<head> element:

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

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1 part sets the initial zoom level when the page is first loaded by the browser.

**3. 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).

Two Basic Bootstrap Pages

The following example shows the code for a basic Bootstrap page (with a responsive fixed width container):

<!DOCTYPE html>  
<html lang="en">  
<head>  
  <title>Bootstrap Example</title>  
  <meta charset="utf-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link rel="stylesheet"href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">  
  <scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>  
  <scriptsrc="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>  
</head>  
<body>  
  
<div class="container">  
  <h1>My First Bootstrap Page</h1>  
  <p>This is some text.</p>   
</div>  
  
</body>  
</html>

The following example shows the code for a basic Bootstrap page (with a full width container):

<!DOCTYPE html>  
<html lang="en">  
<head>  
  <title>Bootstrap Example</title>  
  <meta charset="utf-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link rel="stylesheet"href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">  
  <scriptsrc="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>  
  <scriptsrc="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>  
</head>  
<body>  
  
<div class="container-fluid">  
  <h1>My First Bootstrap Page</h1>  
  <p>This is some text.</p>   
</div>  
  
</body>  
</html>

# Bootstrap Grid

## Bootstrap Grid System

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:

Bootstrap's grid system is responsive, and the columns will re-arrange automatically depending on the screen size.

**Grid Classes**

The Bootstrap grid system has four classes:

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

The classes above can be combined to create more dynamic and flexible layouts.

## Basic Structure of a Bootstrap Grid

The following is a basic structure of a Bootstrap grid:

<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>

First; create a row (<div class="row">). Then, add the desired number of columns (tags with appropriate .col-\*-\* classes). Note that numbers in .col-\*-\* should always add up to 12 for each row.

Below we have collected some examples of basic Bootstrap grid layouts.

## Three Equal Columns

The following example shows how to get a three equal-width columns starting at tablets and scaling to large desktops. On mobile phones, the columns will automatically stack:

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

## Two Unequal Columns

The following example shows how to get two various-width columns starting at tablets and scaling to large desktops:

<div class="row">  
  <div class="col-sm-4">.col-sm-4</div>  
  <div class="col-sm-8">.col-sm-8</div>  
</div>

# Bootstrap Text/Typography

## 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

In this chapter, we will look at some HTML elements that will be styled a little bit different by Bootstrap than browser defaults.

## <h1> - <h6>

By default, Bootstrap will style the HTML headings (<h1> to <h6>) in the following way:

# h1 Bootstrap heading (36px)

## h2 Bootstrap heading (30px)

### h3 Bootstrap heading (24px)

#### h4 Bootstrap heading (18px)

##### **h5 Bootstrap heading (14px)**

###### **h6 Bootstrap heading (12px)**

## <small>

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

# h1 heading secondary text

## h2 heading secondary text

### h3 heading secondary text

#### h4 heading secondary text

##### **h5 heading secondary text**

###### **h6 heading secondary text**

## <blockquote>

<blockquote> element in the following way:

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

From WWF's website

To show the quote on the right, use the .blockquote-reverse class:

.

## <dl><code><kbd><pre>

## 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:

## More Typography Classes

The Bootstrap classes below can be added to style HTML elements further:

|  |  |  |
| --- | --- | --- |
| **Class** | **Description** |  |
| .lead | Makes a paragraph stand out |
| .small | Indicates smaller text (set to 85% of the size of the parent) |
| .text-left | Indicates left-aligned text |
| .text-center | Indicates center-aligned text |
| .text-right | Indicates right-aligned text |
| .text-justify | Indicates justified text |
| .text-nowrap | Indicates no wrap text |
| .text-lowercase | Indicates lowercased text |
| .text-uppercase | Indicates uppercased text |
| .text-capitalize | Indicates capitalized text |
| .initialism | Displays the text inside an <abbr> element in a slightly smaller font size |
| .list-unstyled | Removes the default list-style and left margin on list items (works on both <ul> and <ol>). This class only applies to immediate children list items (to remove the default list-style from any nested lists, apply this class to any nested lists as well) |
| .list-inline | Places all list items on a single line |
| .dl-horizontal | Lines up the terms (<dt>) and descriptions (<dd>) in <dl> elements side-by-side. Starts off like default <dl>s, but when the browser window expands, it will line up side-by-side |
| .pre-scrollable | Makes a <pre> element scrollable |
|  |  |  |

# Bootstrap Table

## Bootstrap Basic Table

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

The .table class adds basic styling to a table:

| **Firstname** | **Lastname** | **Email** |
| --- | --- | --- |
| John | Doe | [john@example.com](mailto:john@example.com) |
| Mary | Moe | [mary@example.com](mailto:mary@example.com) |
| July | Dooley | [july@example.com](mailto:july@example.com) |

## Striped Rows

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

| **Firstname** | **Lastname** | **Email** |
| --- | --- | --- |
| John | Doe | john@example.com |
| Mary | Moe | mary@example.com |
| July | Dooley | july@example.com |

## Bordered Table

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

| **Firstname** | **Lastname** | **Email** |
| --- | --- | --- |
| John | Doe | john@example.com |
| Mary | Moe | mary@example.com |
| July | Dooley | july@example.com |

## 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:

## Contextual Classes

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

| **Firstname** | **Lastname** | **Email** |
| --- | --- | --- |
| John | Doe | john@example.com |
| Mary | Moe | mary@example.com |
| July | Dooley | july@example.com |

The contextual classes that can be used are:

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

# Bootstrap Images

## Rounded Corners

The .img-rounded class adds rounded corners to an image (IE8 do not support rounded corners):

<img src="cinqueterre.jpg" class="img-rounded" alt="Cinque Terre"width="304" height="236">

## Circle

The .img-circle class shapes the image to a circle (IE8 do not support rounded corners):

<img src="cinqueterre.jpg" class="img-circle" alt="Cinque Terre"width="304" height="236">

## Thumbnail

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

<img src="cinqueterre.jpg>

## 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 <img> 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:

<img class="img-responsive" src="img\_chania.jpg" alt="Chania">

## Alerts

Bootstrap provides an easy way to create predefined alert messages:

Alerts are created with the .alert class, followed by one of the four contextual classes.alert-success, .alert-info, .alert-warning or .alert-danger:

<div class="alert alert-success">  
  <strong>Success!</strong> Indicates a successful or positive action.  
</div>  
  
<div class="alert alert-info">  
  <strong>Info!</strong> Indicates a neutral informative change or action.  
</div>  
  
<div class="alert alert-warning">  
  <strong>Warning!</strong> Indicates a warning that might need attention.  
</div>  
  
<div class="alert alert-danger">  
  <strong>Danger!</strong> Indicates a dangerous or potentially negative action.  
</div>

## Closing Alerts

To close the alert message, add class="close" and data-dismiss="alert" to a link or a button element:

<div class="alert alert-success">  
  <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>  
  <strong>Success!</strong> Indicates a successful or positive action.  
</div>

## Animated Alerts

The .fade and .in classes adds a fading effect when closing the alert message:

<div class="alert alert-success fade in" >

# Bootstrap Buttons

## Button Styles

Bootstrap provides seven styles of buttons:

Default Primary Success Info Warning Danger Link

To achieve the button styles above, Bootstrap has the following classes:

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

The following example shows the code for the different button styles:

<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:

Large Medium Small XSmall

The classes that define the different sizes are:

* .btn-lg
* .btn-md
* .btn-sm
* .btn-xs

The following example shows the code for different 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.

Button 1Button 2

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:

Active Primary Disabled Primary

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>

## 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:

To create a basic pagination, add the .pagination class to an <ul> element:

<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>

# Bootstrap Panel

## Panels

A panel in bootstrap is a bordered box with some padding around its content:

A Basic Panel

Panels are created with the .panel class, and content inside the panel has a .panel-body class:

<div class="panel panel-default">  
  <div class="panel-body">A Basic Panel</div>  
</div>

## Panel Heading

The .panel-heading class adds a heading to the panel:

<div class="panel panel-default">  
  <div class="panel-heading">Panel Heading</div>  
  <div class="panel-body">Panel Content</div>  
</div>

## Panel Footer

The .panel-footer class adds a footer to the panel:

<div class="panel panel-default">  
  <div class="panel-body">A Basic Panel</div>  
  <div class="panel-footer">Panel Footer</div>  
</div>

## Panel Group

To group many panels together, wrap a <div> with class .panel-group around them.

The .panel-group class clears the bottom-margin of each panel:

<div class="panel-group">  
  <div class="panel panel-default">  
    <div class="panel-body">Panel Content</div>  
  </div>  
  <div class="panel panel-default">  
    <div class="panel-body">Panel Content</div>  
  </div>  
</div>

## Panels with Contextual Classes

To color the panel, use contextual classes (.panel-default, .panel-primary,.panel-success, .panel-info, .panel-warning, or .panel-danger):

Panel with panel-default class

Panel Content

Panel with panel-primary class

Panel Content

Panel with panel-success class

Panel Content

Panel with panel-info class

Panel Content

Panel with panel-warning class

Panel Content

Panel with panel-danger class

Panel Content

# Bootstrap Dropdowns

## Basic Dropdown

A dropdown menu is a toggleable menu that allows the user to choose one value from a predefined list:

Dropdown Example

<div class="dropdown">  
  <button class="btn btn-primary dropdown-toggle" type="button" data-toggle="dropdown">Dropdown Example  
  <span class="caret"></span></button>  
  <ul class="dropdown-menu">  
    <li><a href="#">HTML</a></li>  
    <li><a href="#">CSS</a></li>  
    <li><a href="#">JavaScript</a></li>  
  </ul>  
</div>

### Example Explained

The .dropdown class indicates a dropdown menu.

To open the dropdown menu, use a button or a link with a class of .dropdown-toggleand the data-toggle="dropdown" attribute.

The .caret class creates a caret arrow icon (), which indicates that the button is a dropdown.

Add the .dropdown-menu class to a <ul> element to actually build the dropdown menu.

## Dropdown Divider

The .divider class is used to separate links inside the dropdown menu:

<li class="divider"></li>

## Dropdown Header

Dropdown Header Example

The .dropdown-header class is used to add headers inside the dropdown menu:

<li class="dropdown-header">Dropdown header 1</li>

## Disable an Item

Dropdown Disabled Example

To disable an item in the dropdown menu, use the .disabled class:

<li class="disabled"><a href="#">CSS</a></li>

## Dropdown Position

Dropdown Right Example

To right-align the dropdown, add the .dropdown-menu-right class to the element with .dropdown-menu:

<ul class="dropdown-menu dropdown-menu-right">

Dropup Example

If you want the dropdown menu to expand upwards instead of downwards, change the <div> element with class="dropdown" to "dropup":

<div class="dropup">

## Dropdown Accessibility

To help improve accessibility for people using screen readers, you should include the following role and aria-\* attributes, when creating a dropdown menu:

<div class="dropdown">  
  <button class="btn btn-default dropdown-toggle" type="button" id="menu1"data-toggle="dropdown">Tutorials  
  <span class="caret"></span></button>  
  <ul class="dropdown-menu" role="menu" aria-labelledby="menu1">  
    <li role="presentation"><a role="menuitem" href="#">HTML</a></li>  
    <li role="presentation"><a role="menuitem" href="#">CSS</a></li>  
    <li role="presentation"><a role="menuitem" href="#">JavaScript</a></li>  
    <li role="presentation" class="divider"></li>  
    <li role="presentation"><a role="menuitem" href="#">About Us</a></li>  
  </ul>  
</div>

# Bootstrap Tabs and Pill

## Menus

Most web pages have some kind of a menu.

In HTML, a menu is often defined in an unordered list <ul> (and styled afterwards), like this:

<ul>  
  <li><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>

If you want to create a horizontal menu of the list above, add the .list-inline class to <ul>:

<ul class="list-inline">

Or you can display the menu above with Bootstraps' Tabs and Pills (see below).

## Tabs

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

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

The following example creates navigation tabs:

<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.

The following example adds a dropdown menu to "Menu 1":

<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>

## Pills

Pills are created with <ul class="nav nav-pills">. Also mark the current page with<li class="active">:

<ul class="nav nav-pills">  
  <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>

### Vertical Pills

Pills can also be displayed vertically. Just add the .nav-stacked class.

The following example places the vertical pill menu inside the last column. So, on a large screen the menu will be displayed to the right. But on a small screen, the content will automatically adjust itself into a single-column layout:

<div class="col-md-3">  
  <ul class="nav nav-pills nav-stacked">  
    <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>  
</div>

### Pills With Dropdown Menu

Pills can also hold dropdown menus.

The following example adds a dropdown menu to "Menu 1":

<ul class="nav nav-pills nav-stacked">  
  <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>

## Centered Tabs and Pills

To center/justify the tabs and pills, use the .nav-justified class.

Note that on screens that are smaller than 768px, the list items are stacked (content will remain centered):

<!-- Centered Tabs -->  
<ul class="nav nav-tabs nav-justified">  
  <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>  
  
<!-- Centered Pills -->  
<ul class="nav nav-pills nav-justified">  
  <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>

## Dynamic Tabs

To make the tabs toggable, add the data-toggle="tab" attribute to each link. Then add a .tab-pane class with a unique ID for every tab and wrap them inside a <div>element with class .tab-content.

If you want the tabs to fade in and out when clicking on them, add the .fade class to.tab-panel:

<ul class="nav nav-tabs">  
  <li class="active"><a data-toggle="tab" href="#home">Home</a></li>  
  <li><a data-toggle="tab" href="#menu1">Menu 1</a></li>  
  <li><a data-toggle="tab" href="#menu2">Menu 2</a></li>  
</ul>  
  
<div class="tab-content">  
  <div id="home" class="tab-pane fade in active">  
    <h3>HOME</h3>  
    <p>Some content.</p>  
  </div>  
  <div id="menu1" class="tab-pane fade">  
    <h3>Menu 1</h3>  
    <p>Some content in menu 1.</p>  
  </div>  
  <div id="menu2" class="tab-pane fade">  
    <h3>Menu 2</h3>  
    <p>Some content in menu 2.</p>  
  </div>  
</div>

## Dynamic Pills

The same code applies to pills; only change the data-toggle attribute to data-toggle="pill":

<ul class="nav nav-pills">  
  <li class="active"><a data-toggle="pill" href="#home">Home</a></li>  
  <li><a data-toggle="pill" href="#menu1">Menu 1</a></li>  
  <li><a data-toggle="pill" href="#menu2">Menu 2</a></li>  
</ul>  
  
<div class="tab-content">  
  <div id="home" class="tab-pane fade in active">  
    <h3>HOME</h3>  
    <p>Some content.</p>  
  </div>  
  <div id="menu1" class="tab-pane fade">  
    <h3>Menu 1</h3>  
    <p>Some content in menu 1.</p>  
  </div>  
  <div id="menu2" class="tab-pane fade">  
    <h3>Menu 2</h3>  
    <p>Some content in menu 2.</p>  
  </div>  
</div>

# Bootstrap Navigation

## Navigation Bars

A 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">.

The following example shows how to add a navigation bar to the top of the page:

<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, black navbar:

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

<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>  
    </div>  
  </div>  
</nav>

## 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:

<nav class="navbar navbar-inverse navbar-fixed-top">  
  <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>

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

<nav class="navbar navbar-inverse navbar-fixed-bottom">  
  <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>

## Navigation Bar With Dropdown

Navigation bars can also hold dropdown menus.

The following example adds a dropdown menu for the "Page 1" button:

<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.

In the following example we insert a "Sign Up" button and a "Login" button to the right in the navigation bar. We also add a glyphicon on each of the two new 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.

In the following example the navigation bar is replaced by a button in the top right corner. Only when the button is clicked, the navigation bar will be displayed:

<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 Form

## Bootstrap's Default Settings

Form controls automatically receive some global styling with Bootstrap:

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

## Bootstrap Form Layouts

Bootstrap provides three types of form layouts:

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

Standard rules for all three form layouts:

* Always use <form role="form"> (helps improve accessibility for people using screen readers)
* Wrap labels and form controls in <div class="form-group"> (needed for optimum spacing)
* Add class .form-control to all textual <input>, <textarea>, and <select>elements

## 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

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

<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>

**Tip:** If you don't include a label for every input, screen readers will have trouble with your forms. You can hide the labels for all devices, except screen readers, by using the .sr-only class:

<form class="form-inline" role="form">  
  <div class="form-group">  
    <label class="sr-only" for="email">Email address:</label>  
    <input type="email" class="form-control" id="email">  
  </div>  
  <div class="form-group">  
    <label class="sr-only" 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

**Tip:** Use Bootstrap's predefined grid classes to align labels and groups of form controls in a horizontal layout.

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

<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>

# Bootstrap Carousel Plugin

## 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">  
      <img src="img\_chania.jpg" alt="Chania">  
    </div>  
  
    <div class="item">  
      <img src="img\_chania2.jpg" alt="Chania">  
    </div>  
  
    <div class="item">  
      <img src="img\_flower.jpg" alt="Flower">  
    </div>  
  
    <div class="item">  
      <img src="img\_flower2.jpg" alt="Flower">  
    </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>

### Example Explained

**The outermost <div>:**

Carousels require the use of an id (in this case id="myCarousel") for carousel controls to function properly.

The class="carousel" specifies that this <div> contains a carousel.

The .slide class adds a CSS transition and animation effect, which makes the items slide when showing a new item. Omit this class if you do not want this effect.

The data-ride="carousel" attribute tells Bootstrap to begin animating the carousel immediately when the page loads.

**The "Indicators" part:**

The indicators are the little dots at the bottom of each slide (which indicates how many slides there are in the carousel, and which slide the user are currently viewing).

The indicators are specified in an ordered list with class .carousel-indicators.

The .data-target class points to the id of the carousel.

The .data-slide-to class specifies which slide to go to, when clicking on the specific dot.

**The "Wrapper for slides" part:**

The slides are specified in a <div> with class .carousel-inner.

The content of each slide is defined in a <div> with class .item. This can be text or images.

The .active class needs to be added to one of the slides. Otherwise, the carousel will not be visible.

**The "Left and right controls" part:**

This code adds "left" and "right" buttons that allows the user to go back and forth between the slides manually.

The data-slide attribute accepts the keywords "prev" or "next", which alters the slide position relative to its current position.

## Add Captions to Slides

Add <div class="carousel-caption"> within each <div class="item"> to create a caption for each slide:

<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">  
      <img src="img\_chania.jpg" alt="Chania">  
      <div class="carousel-caption">  
        <h3>Chania</h3>  
        <p>The atmosphere in Chania has a touch of Florence and Venice.</p>  
      </div>  
    </div>  
  
    <div class="item">  
      <img src="img\_chania2.jpg" alt="Chania">  
      <div class="carousel-caption">  
        <h3>Chania</h3>  
        <p>The atmosphere in Chania has a touch of Florence and Venice.</p>  
      </div>  
    </div>  
  
    <div class="item">  
      <img src="img\_flower.jpg" alt="Flower">  
      <div class="carousel-caption">  
        <h3>Flowers</h3>  
        <p>Beatiful flowers in Kolymbari, Crete.</p>  
      </div>  
    </div>  
  
    <div class="item">  
      <img src="img\_flower2.jpg" alt="Flower">  
      <div class="carousel-caption">  
        <h3>Flowers</h3>  
        <p>Beatiful flowers in Kolymbari, Crete.</p>  
      </div>  
    </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>

# Bootstrap Modal Plugin

**How To Create a Modal**

The following example shows how to create a basic modal:

<!-- Trigger the modal with a button -->  
<button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-target="#myModal">Open Modal</button>  
  
<!-- Modal -->  
<div id="myModal" class="modal fade" role="dialog">  
  <div class="modal-dialog">  
  
    <!-- Modal content-->  
    <div class="modal-content">  
      <div class="modal-header">  
        <button type="button" class="close" data-dismiss="modal">&times;</button>  
        <h4 class="modal-title">Modal Header</h4>  
      </div>  
      <div class="modal-body">  
        <p>Some text in the modal.</p>  
      </div>  
      <div class="modal-footer">  
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>  
      </div>  
    </div>  
  
  </div>  
</div>

### Example Explained

**The "Trigger" part:**

To trigger the modal window, you need to use a button or a link.

Then include the two data-\* attributes:

* data-toggle="modal" opens the modal window
* data-target="#myModal" points to the id of the modal

**The "Modal" part:**

The parent <div> of the modal must have an ID that is the same as the value of the data-target attribute used to trigger the modal ("myModal").

The .modal class identifies the content of <div> as a modal and brings focus to it.

The .fade class adds a transition effect which fades the modal in and out. Remove this class if you do not want this effect.

The attribute role="dialog" improves accessibility for people using screen readers.

The .modal-dialog class sets the proper width and margin of the modal.

**The "Modal content" part:**

The <div> with class="modal-content" styles the modal (border, background-color, etc.). Inside this <div>, add the modal's header, body, and footer.

The .modal-header class is used to define the style for the header of the modal. The<button> inside the header has a data-dismiss="modal" attribute which closes the modal if you click on it. The .close class styles the close button, and the .modal-titleclass styles the header with a proper line-height.

The .modal-body class is used to define the style for the body of the modal. Add any HTML markup here; paragraphs, images, videos, etc.

The .modal-footer class is used to define the style for the footer of the modal. Note that this area is right aligned by default.

## Modal Size

Change the size of the modal by adding the .modal-sm class for small modals or  .modal-lg class for large modals.

Add the size class to the <div> element with class .modal-dialog:

## Small Modal

<div class="modal-dialog modal-sm">

## Large Modal

<div class="modal-dialog modal-lg">

# Bootstrap Tooltip & Pop Over Plugin

## The Tooltip Plugin

The Tooltip plugin is small pop-up box that appears when the user moves the mouse pointer over an element:

## How To Create a Tooltip

To create a tooltip, add the data-toggle="tooltip" attribute to an element.

Use the title attribute to specify the text that should be displayed inside the tooltip:

<a href="#" data-toggle="tooltip" title="Hooray!">Hover over me</a>

**Note:** Tooltips must be initialized with jQuery: select the specified element and call thetooltip() method.

The following code will enable all tooltips in the document:

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

## Positioning Tooltips

By default, the tooltip will appear on top of the element.

Use the data-placement attribute to set the position of the tooltip on top, bottom, left or the right side of the element:

<a href="#" data-toggle="tooltip" data-placement="top"title="Hooray!">Hover</a>  
<a href="#" data-toggle="tooltip" data-placement="bottom"title="Hooray!">Hover</a>  
<a href="#" data-toggle="tooltip" data-placement="left"title="Hooray!">Hover</a>  
<a href="#" data-toggle="tooltip" data-placement="right"title="Hooray!">Hover</a>

# Bootstrap Scrollspy Plugin

## The Scrollspy Plugin

The Scrollspy plugin is used to automatically update links in a navigation list based on scroll position.

## How To Create a Scrollspy

The following example shows how to create a scrollspy:

<!-- The navbar: used to jump to a section in the scrollable area -->  
<nav class="navbar navbar-inverse navbar-fixed-top">  
...  
  <ul class="nav navbar-nav">  
    <li><a href="#section1">Section 1</a></li>  
    ...  
</nav>  
  
<!-- The scrollable area -->  
<div data-spy="scroll" data-target=".navbar" data-offset="12">  
  <!-- Section 1 -->  
  <div id="section1">  
    <h1>Section 1</h1>  
    <p>Try to scroll this page and look at the navigation bar while scrolling!</p>  
  </div>  
  ...  
</div>

### Example Explained

Add data-spy="scroll" to the element that should be used as the scrollable area (often this is the <body> element).

Then add the data-target attribute with a value of the id or the class name of the navigation bar (<nav>). This is to make sure that the navbar is connected with the scrollable area.

Note that scrollable elements must match the ID of the links inside the navbar's list items (<div id="section1"> matches <a href="#section1">).

The optional data-offset attribute specifies the number of pixels to offset from top when calculating the position of scroll. This is useful when you feel that the links inside the navbar changes the active state too soon or too early when jumping to the scrollable elements. Default is 10 pixels.

## Scrollspy Vertical Menu

In this example, we use the <body> element as the scrollable area, and vertical navigation pills as menu:

<body data-spy="scroll" data-target="#myScrollspy" data-offset="20">  
  
  <div class="container">  
    <div class="row">  
      <nav class="col-sm-3" id="myScrollspy">  
        <ul class="nav nav-pills nav-stacked">  
          <li><a href="#section1">Section 1</a></li>  
          ...  
        </ul>  
      </nav>  
      <div class="col-sm-9">  
        <div id="section1">   
          <h1>Section 1</h1>  
          <p>Try to scroll this page and look at the navigation list while scrolling!</p>  
        </div>   
        ...  
      </div>  
    </div>  
  </div>  
  
</body>