**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

Grid Classes

The Bootstrap grid system has four classes:

* xs (for phones - screens less than 768px wide)
* sm (for tablets - screens equal to or greater than 768px wide)
* md (for small laptops - screens equal to or greater than 992px wide)
* lg (for laptops and desktops - screens equal to or greater than 1200px wide)

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 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.

## Two Unequal Columns

<div class="row">  
  <div class="col-sm-4">.col-sm-4</div>  
  <div class="col-sm-8">.col-sm-8</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> element and all paragraphs (<p>).

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 differently by Bootstrap than browser defaults.

## <small>

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

## <blockquote>

Bootstrap will style the HTML <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:

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—

## <dl>

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

**Coffee**

- black hot drink

**Milk**

- white cold drink

## <code>

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

The following HTML elements: span, section, and div defines a section in a document.

## <kbd>

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

Use ctrl + p to open the Print dialog box.

## 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:

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

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

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

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 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 adds a hover effect (grey background color) 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>)

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 |

## Responsive Tables

The .table-responsive class creates a responsive table. The table will then scroll horizontally on small devices (under 768px). When viewing on anything larger than 768px wide, there is no difference

## Rounded Corners

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

<img src="cinqueterre.jpg" class="img-rounded" alt="Cinque Terre">

## Circle

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

## Thumbnail

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

## Responsive Images

Images come 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 display: block; and max-width: 100%; and height: auto; to the image

## Image Gallery

You can also use Bootstrap's grid system in conjunction with the .thumbnail class to create an image gallery.

## Responsive Embeds

Also let videos or slideshows scale properly on any device.

Classes can be applied directly to <iframe>, <embed>, <video>, and <object> elements.

The following example creates a responsive video by adding an .embed-responsive-item class to an <iframe> tag (the video will then scale nicely to the parent element). The containing <div> defines the aspect ratio of the video:

<div class="embed-responsive embed-responsive-16by9">  
  <iframe class="embed-responsive-item" src="..."></iframe>  
</div>

You can choose between two aspect ratio classes:

<!-- 16:9 aspect ratio -->  
<div class="embed-responsive embed-responsive-16by9">  
  <iframe class="embed-responsive-item" src="..."></iframe>  
</div>  
  
<!-- 4:3 aspect ratio -->  
<div class="embed-responsive embed-responsive-4by3">  
  <iframe class="embed-responsive-item" src="..."></iframe>  
</div>

## Creating a Jumbotron

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.

**Tip:** Inside a jumbotron you can put nearly any valid HTML, including other Bootstrap elements/classes.

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

## Jumbotron Inside Container

Place the jumbotron inside the <div class="container"> if you want the jumbotron to NOT extend to the edge of the screen:

<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>

## Jumbotron Outside Container

Place the jumbotron outside the <div class="container"> if you want the jumbotron to extend to the screen edges:

<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>  
<div class="container">  
  <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)

Use a <div> element with class .page-header to create a page header:

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

## Wells

The .well class adds a rounded border around an element with a gray background color and some padding

<div class="well">Basic Well</div>

## Well Size

Change the size of the well by adding the .well-sm class for small wells or  .well-lg class for large wells:

<div class="well well-sm">Small Well</div>  
<div class="well well-lg">Large Well</div>

By default, wells are medium in size.

## lerts

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

## Alert Links

Add the alert-link class to any links inside the alert box to create "matching colored links":

<div class="alert alert-success">  
  <strong>Success!</strong> You should <a href="#" class="alert-link">read this message</a>.  
</div>

## Closing Alerts

To close the alert message, add a .alert-dismissible class to the alert container. Then add class="close" and data-dismiss="alert" to a link or a button element (when you click on this the alert box will disappear).

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

**The aria-\* attribute and &times; explanation**  
  
To help improve accessibility for people using screen readers, you should include the aria-label="close" attribute, when creating a close button.  
  
&times; (×) is an HTML entity that is the preferred icon for close buttons, rather than the letter "x".  
For a list of all HTML Entities, [visit our HTML Entities Reference](https://www.w3schools.com/charsets/ref_html_entities_4.asp).

## Animated Alerts

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

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

## Button Styles

Bootstrap provides different styles of buttons:

* .btn
* .btn-default
* .btn-primary
* .btn-success
* .btn-info
* .btn-warning
* .btn-danger
* .btn-link
* <button type="button" class="btn">Basic</button>  
  <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

The classes that define the different sizes are:

* .btn-lg
* .btn-sm
* .btn-xs
* <button type="button" class="btn btn-primary btn-lg">Large</button>  
  <button type="button" class="btn btn-primary">Normal</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>

## Button Groups

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

Use a <div> element with class .btn-group to create 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>

**Tip:** Instead of applying button sizes to every button in a group, use class .btn-group-lg|sm|xs to size all buttons in the group:

<div class="btn-group btn-group-lg">  
  <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:

Use the class .btn-group-vertical to create a vertical button group:

<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>

## Justified Button Groups

To span the entire width of the screen, use the .btn-group-justified class:

<div class="btn-group btn-group-justified">  
  <a href="#" class="btn btn-primary">Apple</a>  
  <a href="#" class="btn btn-primary">Samsung</a>  
  <a href="#" class="btn btn-primary">Sony</a>  
</div>

**Note:** For <button> elements, you must wrap each button in a .btn-group class:

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

## Nesting Button Groups & Dropdown 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>

## Split Button Dropdowns

<div class="btn-group">  
  <button type="button" class="btn btn-primary">Sony</button>  
  <button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">  
    <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>

## Glyphicon Syntax

A glyphicon is inserted with the following syntax:

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

## Badges

Badges are numerical indicators of how many items are associated with a link:

[News **5**](javascript:void(0);)  
[Comments **10**](javascript:void(0);)  
[Updates **2**](javascript:void(0);)

The numbers (5, 10, and 2) are the badges.

Use the .badge class within <span> elements to create badges

<a href="#">News <span class="badge">5</span></a><br>  
<a href="#">Comments <span class="badge">10</span></a><br>  
<a href="#">Updates <span class="badge">2</span></a>

The following example shows how to add badges to buttons:

<button type="button" class="btn btn-primary">Primary <span class="badge">7</span></button>

## Labels

Labels are used to provide additional information about something:

Use the .label class,  followed by one of the six contextual classes .label-default, .label-primary, .label-success, .label-info, .label-warning or .label-danger, within a <span> element to create a label:

<h1>Example <span class="label label-default">New</span></h1>  
<h2>Example <span class="label label-default">New</span></h2>

<span class="label label-default">Default Label</span>  
<span class="label label-primary">Primary Label</span>  
<span class="label label-success">Success Label</span>  
<span class="label label-info">Info Label</span>  
<span class="label label-warning">Warning Label</span>  
<span class="label label-danger">Danger Label</span>

## Basic Progress Bar

A progress bar can be used to show a user how far along he/she is in a process.

To create a default progress bar, add a .progress class to a <div> element:

<div class="progress">  
  <div class="progress-bar" role="progressbar" aria-valuenow="70"  
  aria-valuemin="0" aria-valuemax="100" style="width:70%">  
    <span class="sr-only">70% Complete</span>  
  </div>  
</div>

**Note:** To help improve accessibility for people using screen readers, you should include the aria-\* attributes.

## Progress Bar With Label

Remove the .sr-only class from the progress bar to show a visible percentage:

<div class="progress">  
  <div class="progress-bar" role="progressbar" aria-valuenow="70"  
  aria-valuemin="0" aria-valuemax="100" style="width:70%">  
    70%  
  </div>  
</div>

Colored Progress Bars

Contextual classes are used to provide "meaning through colors".

The contextual classes that can be used with progress bars are:

* .progress-bar-success
* .progress-bar-info
* .progress-bar-warning
* .progress-bar-danger

<div class="progress">  
  <div class="progress-bar progress-bar-success" role="progressbar" aria-valuenow="40"  
  aria-valuemin="0" aria-valuemax="100" style="width:40%">  
    40% Complete (success)  
  </div>  
</div>  
  
<div class="progress">  
  <div class="progress-bar progress-bar-info" role="progressbar" aria-valuenow="50"  
  aria-valuemin="0" aria-valuemax="100" style="width:50%">  
    50% Complete (info)  
  </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:60%">  
    60% Complete (warning)  
  </div>  
</div>  
  
<div class="progress">  
  <div class="progress-bar progress-bar-danger" role="progressbar" aria-valuenow="70"  
  aria-valuemin="0" aria-valuemax="100" style="width:70%">  
    70% Complete (danger)  
  </div>  
</div>

## Striped Progress Bars

Progress bars can also be striped:

Add class .progress-bar-striped to add stripes to the progress bars:

<div class="progress">  
  <div class="progress-bar progress-bar-success progress-bar-striped" role="progressbar"  
  aria-valuenow="40" aria-valuemin="0" aria-valuemax="100" style="width:40%">  
    40% Complete (success)  
  </div>  
</div>

## Animated Progress Bar

Add class .active to animate the progress bar:

<div class="progress">  
  <div class="progress-bar progress-bar-striped active" role="progressbar"  
  aria-valuenow="40" aria-valuemin="0" aria-valuemax="100" style="width:40%">  
    40%  
  </div>  
</div>

## Stacked Progress Bars

Create a stacked progress bar by placing multiple bars into the same <div class="progress">:

<div class="progress">  
  <div class="progress-bar progress-bar-success" role="progressbar" style="width:40%">  
    Free Space  
  </div>  
  <div class="progress-bar progress-bar-warning" role="progressbar" style="width:10%">  
    Warning  
  </div>  
  <div class="progress-bar progress-bar-danger" role="progressbar" style="width:20%">  
    Danger  
  </div>  
</div>

## Basic Pagination

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

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>

## 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:

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

## Disabled State

A disabled link cannot be clicked:

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

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

## Pagination Sizing

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

<ul class="pagination pagination-lg">  
  <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>  
  
<ul class="pagination pagination-sm">  
  <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>

## Breadcrumbs

Another form for pagination, is breadcrumbs:

The .breadcrumb class indicates the current page's location within a navigational hierarchy:

<ul class="breadcrumb">  
  <li><a href="#">Home</a></li>  
  <li><a href="#">Private</a></li>  
  <li><a href="#">Pictures</a></li>  
  <li class="active">Vacation</li>  
</ul>

# **Bootstrap Pager**

Pager is also a form of pagination (as described in the previous chapter).

Pager provides previous and next buttons (links).

To create previous/next buttons, add the .pager class to an <ul> 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>

## Basic List Groups

The most basic list group is an unordered list with list items:

To create a basic list group, use an <ul> element with class .list-group, and <li> elements with class .list-group-item:

<ul class="list-group">  
  <li class="list-group-item">First item</li>  
  <li class="list-group-item">Second item</li>  
  <li class="list-group-item">Third item</li>  
</ul>

## List Group With Badges

To create a badge, create a <span> element with class .badge inside the list item:

<ul class="list-group">  
  <li class="list-group-item">New <span class="badge">12</span></li>  
  <li class="list-group-item">Deleted <span class="badge">5</span></li>  
  <li class="list-group-item">Warnings <span class="badge">3</span></li>  
</ul>

## List Group With Linked Items

To create a list group with linked items, use <div> instead of <ul> and <a> instead of <li>:

<div class="list-group">  
  <a href="#" class="list-group-item">First item</a>  
  <a href="#" class="list-group-item">Second item</a>  
  <a href="#" class="list-group-item">Third item</a>  
</div>

## Active State

Use the .active class to highlight the current item:

<div class="list-group">  
  <a href="#" class="list-group-item active">First item</a>  
  <a href="#" class="list-group-item">Second item</a>  
  <a href="#" class="list-group-item">Third item</a>  
</div>

## Disabled Item

To disable an item, add the .disabled class:

<div class="list-group">  
  <a href="#" class="list-group-item disabled">First item</a>  
  <a href="#" class="list-group-item">Second item</a>  
  <a href="#" class="list-group-item">Third item</a>  
</div>

## Contextual Classes

The classes for coloring list-items are: .list-group-item-success, list-group-item-info, list-group-item-warning, and .list-group-item-danger:

<ul class="list-group">  
  <li class="list-group-item list-group-item-success">First item</li>  
  <li class="list-group-item list-group-item-info">Second item</li>  
  <li class="list-group-item list-group-item-warning">Third item</li>  
  <li class="list-group-item list-group-item-danger">Fourth item</li>  
</ul>

## Custom Content

You can add nearly any HTML inside a list group item.

Bootstrap provides the classes .list-group-item-heading and .list-group-item-text which can be used as follows:

<div class="list-group">  
  <a href="#" class="list-group-item active">  
    <h4 class="list-group-item-heading">First List Group Item Heading</h4>  
    <p class="list-group-item-text">List Group Item Text</p>  
  </a>  
  <a href="#" class="list-group-item">  
    <h4 class="list-group-item-heading">Second List Group Item Heading</h4>  
    <p class="list-group-item-text">List Group Item Text</p>  
  </a>  
  <a href="#" class="list-group-item">  
    <h4 class="list-group-item-heading">Third List Group Item Heading</h4>  
    <p class="list-group-item-text">List Group Item Text</p>  
  </a>  
</div>

## Panels

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

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 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>

## Basic Dropdown

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

<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-toggle and 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 with a thin horizontal border:

<li class="divider"></li>

## Dropdown Header

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

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

## Disable and Active items

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

## Dropdown Position

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

## 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>

## Basic Collapsible

Collapsibles are useful when you want to hide and show large amount of content:

<button data-toggle="collapse" data-target="#demo">Collapsible</button>  
  
<div id="demo" class="collapse">  
Lorem ipsum dolor text....  
</div>

### **Example Explained**

The .collapse class indicates a collapsible element (a <div> in our example); this is the content that will be shown or hidden with a click of a button.

To control (show/hide) the collapsible content, add the data-toggle="collapse" attribute to an <a> or a <button> element. Then add the data-target="#id" attribute to connect the button with the collapsible content (<div id="demo">).

**Note:** For <a> elements, you can use the href attribute instead of the data-target attribute

<a href="#demo" data-toggle="collapse">Collapsible</a>  
  
<div id="demo" class="collapse">  
Lorem ipsum dolor text....  
</div>

By default, the collapsible content is hidden. However, you can add the .in class to show the content by default:

<div id="demo" class="collapse in">  
Lorem ipsum dolor text....  
</div>

## Collapsible Panel

<div class="panel-group">  
  <div class="panel panel-default">  
    <div class="panel-heading">  
      <h4 class="panel-title">  
        <a data-toggle="collapse" href="#collapse1">Collapsible panel</a>  
      </h4>  
    </div>  
    <div id="collapse1" class="panel-collapse collapse">  
      <div class="panel-body">Panel Body</div>  
      <div class="panel-footer">Panel Footer</div>  
    </div>  
  </div>  
</div>

## Collapsible List Group

<div class="panel-group">  
  <div class="panel panel-default">  
    <div class="panel-heading">  
      <h4 class="panel-title">  
        <a data-toggle="collapse" href="#collapse1">Collapsible list group</a>  
      </h4>  
    </div>  
    <div id="collapse1" class="panel-collapse collapse">  
      <ul class="list-group">  
        <li class="list-group-item">One</li>  
        <li class="list-group-item">Two</li>  
        <li class="list-group-item">Three</li>  
      </ul>  
      <div class="panel-footer">Footer</div>  
    </div>  
  </div>  
</div>

## Accordion

**Note:** Use the data-parent attribute to make sure that all collapsible elements under the specified parent will be closed when one of the collapsible item is shown.

<div class="panel-group" id="accordion">  
  <div class="panel panel-default">  
    <div class="panel-heading">  
      <h4 class="panel-title">  
        <a data-toggle="collapse" data-parent="#accordion" href="#collapse1">  
        Collapsible Group 1</a>  
      </h4>  
    </div>  
    <div id="collapse1" class="panel-collapse collapse in">  
      <div class="panel-body">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.</div>  
    </div>  
  </div>  
  <div class="panel panel-default">  
    <div class="panel-heading">  
      <h4 class="panel-title">  
        <a data-toggle="collapse" data-parent="#accordion" href="#collapse2">  
        Collapsible Group 2</a>  
      </h4>  
    </div>  
    <div id="collapse2" class="panel-collapse collapse">  
      <div class="panel-body">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.</div>  
    </div>  
  </div>  
  <div class="panel panel-default">  
    <div class="panel-heading">  
      <h4 class="panel-title">  
        <a data-toggle="collapse" data-parent="#accordion" href="#collapse3">  
        Collapsible Group 3</a>  
      </h4>  
    </div>  
    <div id="collapse3" class="panel-collapse collapse">  
      <div class="panel-body">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.</div>  
    </div>  
  </div>  
</div>

## 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">

## Tabs

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

<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:

<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>

### **Vertical Pills in a Row**

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**

<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>

## Toggleable / Dynamic Tabs

To make the tabs toggleable, 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-pane:

<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>

## Toggleable / 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>

## 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">.

<nav class="navbar navbar-default">  
  <div class="container-fluid">  
    <div class="navbar-header">  
      <a class="navbar-brand" href="#">WebSiteName</a>  
    </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>  
</nav>  
...

**Note:** All of the examples on this page will show a navigation bar that takes up too much space on small screens (however, the navigation bar will be on one single line on large screens - because Bootstrap is responsive). This problem (with the small screens) will be solved in the last example on this page.

## 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

Navigation Bar With Dropdown

<nav class="navbar navbar-inverse">  
  <div class="container-fluid">  
    <div class="navbar-header">  
      <a class="navbar-brand" href="#">WebSiteName</a>  
    </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>  
</nav>

## Navigation Bar With Dropdown

<nav class="navbar navbar-inverse">  
  <div class="container-fluid">  
    <div class="navbar-header">  
      <a class="navbar-brand" href="#">WebSiteName</a>  
    </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>  
</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>  
    <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>  
    </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>  
</nav>

## Navbar Buttons

To add buttons inside the navbar, add the .navbar-btn class on a Bootstrap button:

<nav class="navbar navbar-inverse">  
  <div class="container-fluid">  
    <div class="navbar-header">  
      <a class="navbar-brand" href="#">WebSiteName</a>  
    </div>  
    <ul class="nav navbar-nav">  
      <li class="active"><a href="#">Home</a></li>  
      <li><a href="#">Link</a></li>  
      <li><a href="#">Link</a></li>  
    </ul>  
    <button class="btn btn-danger navbar-btn">Button</button>  
  </div>  
</nav>

## Navbar Forms

To add form elements inside the navbar, add the .navbar-form class to a form element and add an input(s). Note that we have added a .form-group class to the div container holding the input. This adds proper padding if you have more than one inputs (you will learn more about this in the Forms chapter).

<nav class="navbar navbar-inverse">  
  <div class="container-fluid">  
    <div class="navbar-header">  
      <a class="navbar-brand" href="#">WebSiteName</a>  
    </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>  
    </ul>  
    <form class="navbar-form navbar-left" action="/action\_page.php">  
      <div class="form-group">  
        <input type="text" class="form-control" placeholder="Search">  
      </div>  
      <button type="submit" class="btn btn-default">Submit</button>  
    </form>  
  </div>  
</nav>

You can also use the .input-group and .input-group-addon classes to attach an icon or help text next to the input field. You will learn more about these classes in the Bootstrap Inputs chapter.

<form class="navbar-form navbar-left" action="/action\_page.php">  
  <div class="input-group">  
    <input type="text" class="form-control" placeholder="Search">  
    <div class="input-group-btn">  
      <button class="btn btn-default" type="submit">  
        <i class="glyphicon glyphicon-search"></i>  
      </button>  
    </div>  
  </div>  
</form>

## Navbar Text

Use the .navbar-text class to vertical align any elements inside the navbar that are not links (ensures proper padding and text color).

<nav class="navbar navbar-inverse">  
  <ul class="nav navbar-nav">  
    <li><a href="#">Link</a></li>  
    <li><a href="#">Link</a></li>  
  </ul>  
  <p class="navbar-text">Some text</p>  
</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>  
    <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>  
</nav>

## Collapsing The Navigation Bar

The navigation bar often 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'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

Standard rules for all three form layouts:

* 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 action="/action\_page.php">  
  <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" action="/action\_page.php">  
  <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" action="/action\_page.php">  
  <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 means that the labels are aligned next to the input field (horizontal) on large and medium screens. On small screens (767px and below), it will transform to a vertical form (labels are placed on top of each input).

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.

<form class="form-horizontal" action="/action\_page.php">  
  <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>

Supported Form Controls

Bootstrap supports the following form controls:

* input
* textarea
* checkbox
* radio
* select

## Bootstrap Input

<div class="form-group">  
  <label for="usr">Name:</label>  
  <input type="text" class="form-control" id="usr">  
</div>  
<div class="form-group">  
  <label for="pwd">Password:</label>  
  <input type="password" class="form-control" id="pwd">  
</div>

## Bootstrap Textarea

<div class="form-group">  
  <label for="comment">Comment:</label>  
  <textarea class="form-control" rows="5" id="comment"></textarea>  
</div>

## Bootstrap Checkboxes

<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="checkbox disabled">  
  <label><input type="checkbox" value="" disabled>Option 3</label>  
</div>

Use the .checkbox-inline class if you want the checkboxes to appear on the same line:

<label class="checkbox-inline"><input type="checkbox" value="">Option 1</label>  
<label class="checkbox-inline"><input type="checkbox" value="">Option 2</label>

## Bootstrap Radio Buttons

<div class="radio">  
  <label><input type="radio" name="optradio" checked>Option 1</label>  
</div>  
<div class="radio">  
  <label><input type="radio" name="optradio">Option 2</label>  
</div>  
<div class="radio disabled">  
  <label><input type="radio" name="optradio" disabled>Option 3</label>  
</div>

Use the .radio-inline class if you want the radio buttons to appear on the same line:

<label class="radio-inline"><input type="radio" name="optradio" checked>Option 1</label>

## Bootstrap Select List

<div class="container">

<h2>Form control: select</h2>

<p>The form below contains two dropdown menus (select lists):</p>

<form>

<div class="form-group">

<label for="sel1">Select list (select one):</label>

<select class="form-control" id="sel1">

<option>1</option>

<option>2</option>

<option>3</option>

<option>4</option>

</select>

<br>

<label for="sel2">Mutiple select list (hold shift to select more than one):</label>

<select multiple class="form-control" id="sel2">

<option>1</option>

<option>2</option>

<option>3</option>

<option>4</option>

<option>5</option>

</select>

</div>

</form>

</div>

## Static Control

If you need to insert **plain text** next to a form label within a horizontal form, use the .form-control-static class on a <p> element:

<form class="form-horizontal">  
  <div class="form-group">  
    <label class="control-label col-sm-2">Email:</label>  
    <div class="col-sm-10">  
      <p class="form-control-static">someone@example.com</p>  
    </div>  
  </div>  
</form>

## Bootstrap Input Groups

The .input-group class is a container to enhance an input by adding an icon, text or a button in front or behind it as a "help text".

The .input-group-addon class attaches an icon or help text next to the input field.

<form>  
  <div class="input-group">  
    <span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>  
    <input id="email" type="email" class="form-control" name="email" placeholder="Email">  
  </div>  
  <div class="input-group">  
    <span class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>  
    <input id="password" type="password" class="form-control" name="password" placeholder="Password">  
  </div>  
  <div class="input-group">  
    <span class="input-group-addon">Text</span>  
    <input id="msg" type="text" class="form-control" name="msg" placeholder="Additional Info">  
  </div>  
</form>

The .input-group-btn attaches a button next to an input. This is often used together with a search bar:

<form>  
  <div class="input-group">  
    <input type="text" class="form-control" placeholder="Search">  
    <div class="input-group-btn">  
      <button class="btn btn-default" type="submit">  
        <i class="glyphicon glyphicon-search"></i>  
      </button>  
    </div>  
  </div>  
</form>

## Bootstrap Form Control States

* **INPUT FOCUS** - The outline of the input is removed and a box-shadow is applied on focus
* **DISABLED INPUTS** - Add a disabled attribute to disable an input field
* **DISABLED FIELDSETS** - Add a disabled attribute to a fieldset to disable all controls within
* **READONLY INPUTS** - Add a readonly attribute to an input to prevent user input
* **VALIDATION STATES** - Bootstrap includes validation styles for error, warning, and success messages. To use, add .has-warning, .has-error, or .has-success to the parent element
* **ICONS** - You can add feedback icons with the .has-feedback class and an icon
* **HIDDEN LABELS** - Add a .sr-only class on non-visible labels

<form class="form-horizontal">  
  <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="Click to focus">  
    </div>  
  </div>  
  <div class="form-group">  
    <label for="disabledInput" class="col-sm-2 control-label">Disabled</label>  
    <div class="col-sm-10">  
      <input class="form-control" id="disabledInput" type="text" disabled>  
    </div>  
  </div>  
  <fieldset disabled>  
    <div class="form-group">  
      <label for="disabledTextInput" class="col-sm-2 control-label">Fieldset disabled</label>  
      <div class="col-sm-10">  
        <input type="text" id="disabledTextInput" class="form-control">  
      </div>  
    </div>  
    <div class="form-group">  
      <label for="disabledSelect" class="col-sm-2 control-label"></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 has-feedback">  
    <label class="col-sm-2 control-label" for="inputSuccess">  
    Input with success and icon</label>  
    <div class="col-sm-10">  
      <input type="text" class="form-control" id="inputSuccess">  
      <span class="glyphicon glyphicon-ok form-control-feedback"></span>  
    </div>  
  </div>  
  <div class="form-group has-warning has-feedback">  
    <label class="col-sm-2 control-label" for="inputWarning">  
    Input with warning and icon</label>  
    <div class="col-sm-10">  
      <input type="text" class="form-control" id="inputWarning">  
      <span class="glyphicon glyphicon-warning-sign form-control-feedback"></span>  
    </div>  
  </div>  
  <div class="form-group has-error has-feedback">  
    <label class="col-sm-2 control-label" for="inputError">  
    Input with error and icon</label>  
    <div class="col-sm-10">  
      <input type="text" class="form-control" id="inputError">  
      <span class="glyphicon glyphicon-remove form-control-feedback"></span>  
    </div>  
  </div>  
</form>

## Input Sizing in Forms

Set the heights of input elements using classes like .input-lg and .input-sm.

Set the widths of elements using grid column classes like .col-lg-\*and .col-sm-\*.

You can quickly size labels and form controls within a **Horizontal form** by adding .form-group-\* to the <div class="form-group"> element:

<div class="form-group form-group-lg">

## Column Sizing

The following examples shows input elements with different widths using different .col-xs-\* classes:

<div class="form-group row">  
  <div class="col-xs-2">  
    <label for="ex1">col-xs-2</label>  
    <input class="form-control" id="ex1" type="text">  
  </div>  
  <div class="col-xs-3">  
    <label for="ex2">col-xs-3</label>  
    <input class="form-control" id="ex2" type="text">  
  </div>  
  <div class="col-xs-4">  
    <label for="ex3">col-xs-4</label>  
    <input class="form-control" id="ex3" type="text">  
  </div>  
</div>

## Help Text

Use the .help-block class to add a block level help text in forms:

<div class="form-group">  
  <label for="pwd">Password:</label>  
  <input type="password" class="form-control" id="pwd" placeholder="Enter password">  
  <span class="help-block">This is some help text...</span>  
</div>

## Media Objects

Bootstrap provides an easy way to align media objects (like images or videos) to the left or to the right of some content. This can be used to display blog comments, tweets and so on:

## Basic Media Object

<!-- Left-aligned -->  
<div class="media">  
  <div class="media-left">  
    <img src="img\_avatar1.png" class="media-object" style="width:60px">  
  </div>  
  <div class="media-body">  
    <h4 class="media-heading">John Doe</h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>  
  
<!-- Right-aligned -->  
<div class="media">  
  <div class="media-body">  
    <h4 class="media-heading">John Doe</h4>  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>  
  </div>  
  <div class="media-right">  
    <img src="img\_avatar1.png" class="media-object" style="width:60px">  
  </div>  
</div>

Top, Middle or Bottom Alignment

The media object can also be top, middle or bottom aligned with the media-top, media-middle or media-bottom class

<!-- Media top -->  
<div class="media">  
  <div class="media-left media-top">  
    <img src="img\_avatar1.png" class="media-object" style="width:60px">  
  </div>  
  <div class="media-body">  
    <h4 class="media-heading">Media Top</h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>  
  
<!-- Media middle -->  
<div class="media">  
  <div class="media-left media-middle">  
    <img src="img\_avatar1.png" class="media-object" style="width:60px">  
  </div>  
  <div class="media-body">  
    <h4 class="media-heading">Media Middle</h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>  
  
<!-- Media bottom -->  
<div class="media">  
  <div class="media-left media-bottom">  
    <img src="img\_avatar1.png" class="media-object" style="width:60px">  
  </div>  
  <div class="media-body">  
    <h4 class="media-heading">Media Bottom</h4>  
    <p>Lorem ipsum...</p>  
  </div>  
</div>

## Nesting Media Objects

Media objects can also be nested (a media object inside a media object)…

## The Carousel Plugin

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

**Tip:**Plugins can be included individually (using Bootstrap's individual "carousel.js" file), or all at once (using "bootstrap.js" or "bootstrap.min.js").

## How To Create a 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>  
  </ol>  
  
  <!-- Wrapper for slides -->  
  <div class="carousel-inner">  
    <div class="item active">  
      <img src="la.jpg" alt="Los Angeles">  
    </div>  
  
    <div class="item">  
      <img src="chicago.jpg" alt="Chicago">  
    </div>  
  
    <div class="item">  
      <img src="ny.jpg" alt="New York">  
    </div>  
  </div>  
  
  <!-- Left and right controls -->  
  <a class="left carousel-control" href="#myCarousel" data-slide="prev">  
    <span class="glyphicon glyphicon-chevron-left"></span>  
    <span class="sr-only">Previous</span>  
  </a>  
  <a class="right carousel-control" href="#myCarousel" data-slide="next">  
    <span class="glyphicon glyphicon-chevron-right"></span>  
    <span class="sr-only">Next</span>  
  </a>  
</div>

**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 is currently viewing).

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

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

The data-slide-to attribute 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>  
  </ol>  
  
  <!-- Wrapper for slides -->  
  <div class="carousel-inner">  
    <div class="item active">  
      <img src="la.jpg" alt="Chania">  
      <div class="carousel-caption">  
        <h3>Los Angeles</h3>  
        <p>LA is always so much fun!</p>  
      </div>  
    </div>  
  
    <div class="item">  
      <img src="chicago.jpg" alt="Chicago">  
      <div class="carousel-caption">  
        <h3>Chicago</h3>  
        <p>Thank you, Chicago!</p>  
      </div>  
    </div>  
  
    <div class="item">  
      <img src="ny.jpg" alt="New York">  
      <div class="carousel-caption">  
        <h3>New York</h3>  
        <p>We love the Big Apple!</p>  
      </div>  
    </div>  
  </div>  
  
  <!-- Left and right controls -->  
  <a class="left carousel-control" href="#myCarousel" data-slide="prev">  
    <span class="glyphicon glyphicon-chevron-left"></span>  
    <span class="sr-only">Previous</span>  
  </a>  
  <a class="right carousel-control" href="#myCarousel" data-slide="next">  
    <span class="glyphicon glyphicon-chevron-right"></span>  
    <span class="sr-only">Next</span>  
  </a>  
</div>

## The Modal Plugin

The Modal plugin is a dialog box/popup window that is displayed on top of the current page

**Tip:**Plugins can be included individually (using Bootstrap's individual "modal.js" file), or all at once (using "bootstrap.js" or "bootstrap.min.js").

<!-- 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-title class 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:

<div class="modal-dialog modal-sm">

<div class="modal-dialog modal-lg">

## 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 the tooltip() 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>

**Tip:**You can also use the data-placement attribute with a value of "auto", which will let the browser decide the position of the tooltip. For example, if the value is "auto left", the tooltip will display on the left side when possible, otherwise on the right.

## The Popover Plugin

The Popover plugin is similar to tooltips; it is a pop-up box that appears when the user clicks on an element. The difference is that the popover can contain much more content.

**Tip:**Plugins can be included individually (using Bootstrap's individual "popover.js" file), or all at once (using "bootstrap.js" or "bootstrap.min.js").

<a href="#" data-toggle="popover" title="Popover Header" data-content="Some content inside the popover">Toggle popover</a>

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

The following code will enable all popovers in the document:

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

## Positioning Popovers

By default, the popover will appear on the right side of the element.

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

<a href="#" title="Header" data-toggle="popover" data-placement="top" data-content="Content">Click</a>  
<a href="#" title="Header" data-toggle="popover" data-placement="bottom" data-content="Content">Click</a>  
<a href="#" title="Header" data-toggle="popover" data-placement="left" data-content="Content">Click</a>  
<a href="#" title="Header" data-toggle="popover" data-placement="right" data-content="Content">Click</a>**Tip:**You can also use the data-placement attribute with a value of "auto", which will let the browser decide the position of the popover. For example, if the value is "auto left", the popover will display on the left side when possible, otherwise on the right.

## Closing Popovers

By default, the popover is closed when you click on the element again. However, you can use the data-trigger="focus" attribute which will close the popover when clicking outside the element:

<a href="#" title="Dismissible popover" data-toggle="popover" data-trigger="focus" data-content="Click anywhere in the document to close this popover">Click me</a>

**Tip:** If you want the popover to be displayed when you move the mouse pointer over the element, use the data-trigger attribute with a value of "hover":

<a href="#" title="Header" data-toggle="popover" data-trigger="hover" data-content="Some content">Hover over me</a>

## The Scrollspy Plugin

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

<!-- The scrollable area -->  
<body data-spy="scroll" data-target=".navbar" data-offset="50">  
  
<!-- The navbar - The <a> elements are 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>  
  
<!-- 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>  
...  
  
</body>

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 (.navbar). 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.

**Requires relative positioning:**The element with data-spy="scroll" requires the CSS **position** property, with a value of "relative" to work properly.

## Scrollspy Vertical 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>

## The Affix Plugin

The Affix plugin allows an element to become affixed (locked) to an area on the page. This is often used with navigation menus or social icon buttons, to make them "stick" at a specific area while scrolling up and down the page.

The plugin **toggles** this behavior on and off (changes the value of CSS position from static to fixed), depending on scroll position.

The following example shows how to create a horizontal affixed navigation menu:

<nav class="navbar navbar-inverse" data-spy="affix" data-offset-top="197">

The following example shows how to create a vertical affixed navigation menu:

<ul class="nav nav-pills nav-stacked" data-spy="affix" data-offset-top="205">

Example Explained

Add data-spy="affix" to the element you want affixed.

Optionally, add the data-offset-top|bottom attribute to calculate the position of the scroll.

How it works

The affix plugin toggles between three classes: .affix, .affix-top, and .affix-bottom. Each class represents a particular state. You must add CSS properties to handle the actual positions, with the exception of position:fixed on the .affix class.

* The plugin adds the .affix-top or .affix-bottom class to indicate the element is in its top-most or bottom-most position. Positioning with CSS is not required at this point.

* Scrolling past the affixed element triggers the actual affixing - This is where the plugin replaces the .affix-top or .affix-bottom class with the .affix class (sets position:fixed). At this point, you must add the CSS top or bottom property to position the affixed element in the page.

* If a bottom offset is defined, scrolling past it replaces the .affix class with .affix-bottom. Since offsets are optional, setting one requires you to set the appropriate CSS. In this case, add position:absolute when necessary.

In the first example above, the Affix plugin adds the .affix class (position:fixed) to the <nav> element when we have scrolled 197 pixels from the top. If you open the example, you will also see that we added the CSS top property with a value of 0 to the .affix class. This is to make sure that the navbar stays at the top of the page at all time, when we have scrolled 197 pixels from the top.

## Scrollspy & Affix

# Horizontal Menu (Navbar)

<body data-spy="scroll" data-target=".navbar" data-offset="50">  
  
<nav class="navbar navbar-inverse" data-spy="affix" data-offset-top="197">  
...  
</nav>  
  
</body>

# Vertical Menu (Sidenav)

<body data-spy="scroll" data-target="#myScrollspy" data-offset="15">  
  
<nav class="col-sm-3" id="myScrollspy">  
  <ul class="nav nav-pills nav-stacked" data-spy="affix" data-offset-top="205">  
  ...  
</nav>  
  
</body>

## Bootstrap Filters

Bootstrap does not have a component that allows filtering. However, we can use [jQuery](https://www.w3schools.com/jquery/default.asp) to filter / search for elements.

<script>  
$(document).ready(function(){  
  $("#myInput").on("keyup", function() {  
    var value = $(this).val().toLowerCase();  
    $("#myTable tr").filter(function() {  
      $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)  
    });  
  });  
});  
</script>

**Example explained:** We use jQuery to loop through each table rows to check if there are any text values that matches the value of the input field. The toggle() method hides the row (display:none) that does not match the search. We use the toLowerCase() method to convert the text to lower case, which makes the search case insensitive (allows "john", "John", and even "JOHN" on search).

## 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:

**Tip:** Remember that grid columns should add up to twelve for a row. More than that, columns will stack no matter the viewport.

## Grid Classes

The Bootstrap grid system has four classes:

* xs (for phones - screens less than 768px wide)
* sm (for tablets - screens equal to or greater than 768px wide)
* md (for small laptops - screens equal to or greater than 992px wide)
* lg (for laptops and desktops - screens equal to or greater than 1200px wide)

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

**Tip:** Each class scales up, so if you wish to set the same widths for xs and sm, you only need to specify xs.

Grid System Rules

Some Bootstrap grid system rules:

* Rows must be placed within a .container (fixed-width) or .container-fluid (full-width) 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 classes like .row and .col-sm-4 are available for quickly making grid 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 12 available columns you wish to span. For example, three equal columns would use three .col-sm-4
* Column widths are in percentage, so they are always fluid and sized relative to their parent element

## Basic Structure of a Bootstrap Grid

The following is a basic structure of a Bootstrap grid:

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

So, to create the layout you want, create a container (<div class="container">). Next, 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.

Grid Options

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Extra small <768px** | **Small >=768px** | **Medium >=992px** | **Large >=1200px** |
| **Class prefix** | .col-xs- | .col-sm- | .col-md- | .col-lg- |
| **Suitable for** | Phones | Tablets | Small Laptops | Laptops & Desktops |
| **Grid behaviour** | Horizontal at all times | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints | Collapsed to start, horizontal above breakpoints |
| **Container width** | None (auto) | 750px | 970px | 1170px |
| **# of columns** | 12 | 12 | 12 | 12 |
| **Column width** | Auto | ~62px | ~81px | ~97px |
| **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 |

## Bootstrap Grid Example: Stacked-to-horizontal

We will create a basic grid system that starts out stacked on extra small devices, before becoming horizontal on larger devices.

The following example shows a simple "stacked-to-horizontal" two-column layout, meaning it will result in a 50%/50% split on all screens, except for extra small screens, which it will automatically stack (100%):

<div class="container">  
  <h1>Hello World!</h1>  
  <div class="row">  
    <div class="col-sm-6" style="background-color:yellow;">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-sm-6" style="background-color:pink;">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>

**Tip:** The numbers in the .col-sm-\* classes indicates how many columns the div should span (out of 12). So, .col-sm-1 spans 1 column, .col-sm-4 spans 4 columns, .col-sm-6 spans 6 columns, etc.  
  
**Note:** Make sure that the sum always adds up to 12!

Bootstrap Grid Example: Small Devices

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Extra small** | **Small** | **Medium** | **Large** |
| **Class prefix** | .col-xs | .col-sm | .col-md | .col-lg |
| **Screen width** | <768px | >=768px | >=992px | >=1200px |

Assume we have a simple layout with two columns. We want the columns to be split 25%/75% for small devices.

**Tip:** Small devices are defined as having a screen width from **768 pixels to 991 pixels**.

For small devices we will use the .col-sm-\* classes.

We will add the following classes to our two columns:

<div class="col-sm-3">....</div>  
<div class="col-sm-9">....</div>

Now Bootstrap is going to say "at the small size, look for classes with -sm- in them and use those".

The following example will result in a 25%/75% split on small (and medium and large) devices. On extra small devices, it will automatically stack (100%)

<div class="container-fluid">  
  <h1>Hello World!</h1>  
  <div class="row">  
    <div class="col-sm-3" style="background-color:yellow;">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-sm-9" style="background-color:pink;">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>

For a 33.3%/66.6% split, you would use .col-sm-4 and .col-sm-8:

<div class="container-fluid">  
  <h1>Hello World!</h1>  
  <div class="row">  
    <div class="col-sm-4" style="background-color:yellow;">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-sm-8" style="background-color:pink;">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>

**Bootstrap Grid Example: Medium Devices**

In the previous chapter, we presented a grid example with classes for small devices. We used two divs (columns) and we gave them a 25%/75% split:

<div class="col-sm-3 **col-md-6**">....</div>  
<div class="col-sm-9 **col-md-6**">....</div>

The following example will result in a 25%/75% split on small devices and a 50%/50% split on medium (and large) devices. On extra small devices, it will automatically stack (100%):

<div class="container-fluid">  
  <h1>Hello World!</h1>  
  <div class="row">  
    <div class="col-sm-3 col-md-6" style="background-color:yellow;">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-sm-9 col-md-6" style="background-color:pink;">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>

**Bootstrap Grid Example: Large Devices**

So now we will add the column widths for large devices:

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

But on large devices the design may be better as a 33%/66% split.

**Tip:** Large devices are defined as having a screen width from **1200 pixels and above**.

For large devices we will use the .col-lg-\* classes.

Example

<div class="container-fluid">  
  <h1>Hello World!</h1>  
  <div class="row">  
    <div class="col-sm-3 col-md-6 col-lg-4" style="background-color:yellow;">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-sm-9 col-md-6 col-lg-8" style="background-color:pink;">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>

**Using Only Large**

In the example below, we only specify the .col-lg-6 class (without .col-md-\* and/or .col-sm-\*). This means that large devices will split 50%/50%. However, for medium AND small devices, it will stack vertically (100% width):

<div class="container-fluid">  
  <h1>Hello World!</h1>  
  <div class="row">  
    <div class="col-lg-6" style="background-color:yellow;">  
      <p>Lorem ipsum...</p>  
    </div>  
    <div class="col-lg-6" style="background-color:pink;">  
      <p>Sed ut perspiciatis...</p>  
    </div>  
  </div>  
</div>