Bootstrap Tutorial

Bootstrap is the most popular and powerful front-end (HTML, CSS, and JS) framework for faster and easier responsive web development.

Bootstrap tutorials will help you to learn the fundamentals as well as the advanced features of the latest Bootstrap framework step-by-step, so that you can create responsive, interactive and feature rich websites with much less effort.

Tutorials are broken down into sections where each section containing a number of related topics that are packed with easy to understand explanations, real-life practice examples, smart workarounds as well as useful tips and important notes.

You can save a lot of time and efforts with Bootstrap — So bookmark this website and continue on.

If you're completely new to the Bootstrap we recommend you to start with the section that covers Bootstrap Basics and gradually move forward, by learning a little bit every day.

Tip: Since every chapter in this tutorial is somewhat related to each other. So it's a good idea to make sure that you have understood the each topic very clearly before moving on to the next chapter or section.

Bootstrap Examples

Every chapter in this tutorial contains practice examples that you can try and test yourself to extend your learning. The purpose of these examples is to provide you a better understanding of the usage of Bootstrap in your day to day implementation.

Check out the Bootstrap Examples »

Bootstrap Button Generator

An interactive online tool that will help you to quickly create your favorite Bootstrap buttons with various colors and icons without writing a single line of code.

Check out the Bootstrap Button Generator »

Previous Page Next Page

Bootstrap Introduction

Bootstrap comes equipped with HTML, CSS, and JavaScript for various web and user interface components.

What is Bootstrap

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for common user interface components like Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel and many other as well as optional JavaScript extensions.

Bootstrap also gives you ability to create responsive layout with much less efforts.

Advantages of Bootstrap

The biggest advantage of using Bootstrap is that it comes with free set of tools for creating flexible and responsive web layouts as well as common interface components.

Additionally, using the Bootstrap data APIs you can create advanced interface components like Scrollspy and Typeaheads without writing a single line of JavaScript.

Here are some more advantages, why one should opt for Bootstrap:

- **Save lots of time** — You can save lots of time and efforts using the Bootstrap predefined design templates and classes and concentrate on other development work.
- **Responsive features** — Using Bootstrap you can easily create responsive designs. Bootstrap responsive features make your web pages to appear more appropriately on different devices and screen resolutions without any change in markup.
- **Consistent design** — All Bootstrap components share the same design templates and styles through a central library, so that the designs and layouts of your web pages are consistent throughout your development.
- **Easy to use** — Bootstrap is very easy to use. Anybody with the basic working knowledge of HTML and CSS can start development with Bootstrap.
- **Compatible with browsers** — Bootstrap is created with modern browsers in mind and it is compatible with all modern browsers such as Mozilla Firefox, Google Chrome, Safari, Internet Explorer, and Opera.
- **Open Source** — And the best part is, it is completely free to download and use.

Note: Some CSS3 properties such as properties for rounded corners, gradients and shadows are used by the Bootstrap but not supported in older versions of the web browsers especially Internet Explorer 8 and earlier.

Previous Page Next Page

Bootstrap Getting Started

In this tutorial you will learn how to create a basic Bootstrap template using the Bootstrap 3 compiled version.

Getting Started with Bootstrap

Here, you will learn how easy it is to create a web page using Bootstrap. Before begin, be sure to have a code editor and some working knowledge of HTML and CSS.

If you're just starting out in web development, start learning from here »

OK, let's get straight into it.

Downloading the Bootstrap Files

There are two versions available for download, **compiled Bootstrap** and **Bootstrap source** files. You can download Bootstrap files from here.

Compiled download contain compiled and minified version of CSS and JavaScript files as well as icons in font format for faster and easier web development, while the source contain original source files for all CSS and JavaScript, along with a local copy of the docs.

For the purpose of better understanding we'll focus on the compiled Bootstrap files. It saves your time because you don't have to bother every time including separate files for individual functionality. It will also increase the performance of your website and saves the precious bandwidth when you decided to move your site on production because of lesser HTTP request and download size since files are compiled and minified.

Understanding the File Structure

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

```
bootstrap/
|── css/
|   |── bootstrap.css
|   |── bootstrap.min.css
```

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

As you can see compiled version of Bootstrap provides compiled CSS and JS files (`bootstrap.*`), as well as compiled and minified CSS and JS (`bootstrap.min.*`).

There are four font files (`glyphicons-halflings-regular.*`) inside the fonts folder. These fonts file includes more than 250 icons from the Glyphicon Halflings set.

Tip: This is the most basic form of Bootstrap for quick usage in any web project. Please note that all JavaScript plugins require jQuery to be included.

Creating Your First Web Page with Bootstrap

So far you have understood the structure and the purposes of Bootstrap files, now it's time to put Bootstrap into real use. In this section, we'll create a basic Bootstrap template that includes everything we mentioned in the file structure.

Let's walk through the following steps. At the end of the tutorial, you will have made an HTML file that displays "Hello world" message in your web browser.

Step 1: Creating a Basic HTML file

Open up your favorite code editor and create a new HTML file. Start with an empty window and type the following code and save it as "basic.html" on your desktop.

Example

Try this code »

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Basic HTML File</title>
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
    <h1>Hello, world!</h1>
</body>
</html>
```

Tip: Add the viewport `<meta>` tag inside the `<head>` section of your document to enable touch zooming and ensure proper rendering on mobile devices.

Step 2: Making this HTML File a Bootstrap Template

To make this HTML file a Bootstrap template, just include the appropriate Bootstrap CSS and JS files. You should include JavaScript files at the bottom of the page — before closing of the `<body>` tag (i.e. `</body>`) to improve the performance of your web pages.

Example

Try this code »

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Basic Bootstrap Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
    <!-- Optional Bootstrap theme -->
    <link rel="stylesheet" href="css/bootstrap-theme.min.css">
</head>
<body>
    <h1>Hello, world!</h1>
    <script src="js/jquery-1.11.3.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

And we're all set! after adding the Bootstrap's CSS and JS files and the required jQuery library, we can begin to develop any site or application with Bootstrap framework.

Step 3: Saving the file

Now save the file on your desktop as "bootstrap-template.html".

Note: It is important that the extension `".html"` is specified — some text editors, such as Notepad, will automatically save it as `".txt"` otherwise.

To open the file in a browser. Navigate to your file then double click on it. It will open in your default Web browser. (If it does not, open your browser and drag the file to it.)

Including Bootstrap's Files via CDN

You can also include the Bootstrap's CSS and JavaScript files as well as the jQuery library JavaScript file in your document using the freely available CDN (Content Delivery Network) links, if you don't want to download and host the Bootstrap or jQuery yourself.

CDNs can offer a performance benefit by reducing the loading time, because they are hosting the Bootstrap's files on multiple servers spread across the globe and when a user requests the file, it will be served from the server nearest to them.

Example

Try this code »

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Basic Bootstrap Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
    <!-- Optional Bootstrap theme -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap-theme.min.css">
</head>
<body>
    <h1>Hello, world!</h1>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</body>
</html>
```

In the above example, we've included the compiled and minified version of Bootstrap's CSS and JavaScript files as well as the necessary jQuery library using the CDN links. You'll also find these CDN links in most of the practice examples code throughout this site.

Tip: If the visitor to your site has already downloaded the Bootstrap's files from the same CDN while visiting the other sites, it will be loaded from the browser's cache instead of re-downloading, which leads to faster loading time.

Previous Page Next Page

Bootstrap Grid System

The Bootstrap grid system is the fastest and easy way to create a layout.

What is Bootstrap Grid System

Bootstrap grid system provides the quick and easy way to create responsive website layouts. As opposed to the previous Bootstrap 2.x grid system which is fixed by default, the new version, i.e. Bootstrap 3 introduces the responsive mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases.

Bootstrap 3 includes predefined grid classes for quickly making grid layouts for different types of devices like cell phones, tablets, laptops and desktops, etc. For example, you can use the `.col-xs-*` class to create grid columns for extra small devices like cell phones, similarly the `.col-sm-*` class for small screen devices like tablets, the `.col-md-*` class for medium size devices like desktops and the `.col-lg-*` for large desktop screens. The following table summarizes some of the key features of the new grid system

Features Bootstrap 3 Grid System	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
`Max container width`	None (auto)	`750px`	`970px`	`1170px`
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Class prefix	`.col-xs-`	`.col-sm-`	`.col-md-`	`.col-lg-`
Max column width	`Auto`	`~62px`	`~81px`	`~97px`
Gutter width	`15px` on each side of a column (i.e. `30px`)			

Above table demonstrates one important thing, applying any `.col-sm-*` class to an element will not only affect its styling on small devices, but also on medium and large devices having a screen size greater than or equal to `768px` (i.e. `≥768px`) if `.col-md-*` and `.col-lg-*` class is not present. Similarly the `.col-md-*` class will not only affect the styling of elements on medium devices, but also on large devices if a `.col-lg-*` class is not present.

Now the question arises how to create rows and columns using this 12 column responsive grid system. The answer is pretty simple, at first create a container that acts as a wrapper for your rows and columns using the `.container` class, after that create rows inside the container using the `.row` class, and to create columns inside any row you can use the class `.col-xs-*`, `.col-sm-*`, `.col-md-*` and `.col-lg-*`. The columns are actual content area where we will place our contents. Let's put all these things into real action.

Creating Two Column Layouts

The following example will show you how to create two column layouts for small, medium and large devices like tables, laptops and desktops etc. However, on mobile phones, the columns will automatically become horizontal as default.

Example

Try this code »

```
<div class="container">
    <!--Row with two equal columns-->
    <div class="row">
        <div class="col-sm-6"><!--Column left--></div>
        <div class="col-sm-6"><!--Column right--></div>
    </div>

    <!--Row with two columns divided in 1:2 ratio-->
    <div class="row">
        <div class="col-sm-4"><!--Column left--></div>
        <div class="col-sm-8"><!--Column right--></div>
    </div>

    <!--Row with two columns divided in 1:3 ratio-->
    <div class="row">
        <div class="col-sm-3"><!--Column left--></div>
        <div class="col-sm-9"><!--Column right--></div>
    </div>
</div>
```

Tip: Open the output of all grid examples inside the CodeLab editor in a new blank tab (click the  link next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.

Since Bootstrap grid system is based on 12 columns, so to keep the columns in a one line (i.e. side by side), the sum of the grid column numbers in each row should be equal to 12. If you see the above example carefully you will find the numbers of grid columns (i.e. `col-sm-*`) add up to twelve (`6+6`, `4+8` and `3+9`) for every row.

Creating Three Column Layouts

Similarly, you can create other layouts based on the above principle. The following example will typically create three column layouts for laptops and desktops screens. It also works in tablets in landscape mode if screen resolution is more than or equal to 992 pixels (e.g. Apple iPad). However, in portrait mode it will be horizontal as usual.

Example

Try this code »

```
<div class="container">
    <!--Row with three equal columns-->
    <div class="row">
        <div class="col-md-4"><!--Column left--></div>
        <div class="col-md-4"><!--Column middle--></div>
        <div class="col-md-4"><!--Column right--></div>
    </div>

    <!--Row with three columns divided in 1:4:1 ratio-->
    <div class="row">
        <div class="col-md-2"><!--Column left--></div>
        <div class="col-md-8"><!--Column middle--></div>
        <div class="col-md-2"><!--Column right--></div>
    </div>

    <!--Row with three columns divided unevenly-->
    <div class="row">
        <div class="col-md-3"><!--Column left--></div>
        <div class="col-md-7"><!--Column middle--></div>
        <div class="col-md-2"><!--Column right--></div>
    </div>
</div>
```

Note: If more than 12 grid columns are placed within a single row, then each group of extra columns, as a whole, will wrap onto a new line.

Bootstrap Layouts with Column Wrapping Feature

Now we are going to create more flexible layouts that changes the column orientation based on the viewport size. The following example will create a three column layout on medium devices like laptops and desktops, as well as on tablets (e.g. Apple iPad) in landscape mode, but on small devices like tablets in portrait mode, it will change into a two column layout where the third column moves at the bottom of the first two columns.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-sm-3 col-md-2"><!--Column one--></div>
        <div class="col-sm-9 col-md-8"><!--Column two--></div>
        <div class="col-sm-12 col-md-2"><!--Column three--></div>
    </div>
</div>
```

As you can see in the example above the sum of small grid column numbers (i.e. `col-sm-*`) is `3 + 9 + 12 = 24 > 12`, so the third `<div>` element with the class `.col-sm-12` that is adding the extra columns beyond the maximum 12 columns in a `.row`, gets wrapped onto a new line as one contiguous unit on small devices having the viewport width less than the 992 pixels.

Similarly, you can create even more adaptable layouts for your websites and applications using the Bootstrap's grid column wrapping feature. In the next section, we'll discuss the other aspect of this feature. Here're some ready to use Bootstrap grid examples.

Creating Multi-Column Layouts with Bootstrap 3 Grid System

With the new Bootstrap 3 mobile first grid system you can easily control how your website layout will render on different types of devices that have different screen sizes like cell phones, tablets, desktops, etc. Let's consider the following illustration.

In the above illustration there are total 12 content boxes in all devices, but its placement varies according to the device screen size, like in mobile device the layout is rendered as one column grid layout which has 1 column and 12 rows placed above one another, whereas in tablet it is rendered as two column grid layout which has 2 columns and 6 rows. Further, in medium screen size devices like laptops and desktops it is rendered as three column grid layout which as 3 columns and 4 rows and finally in large screen devices like large desktops it is rendered as four column grid layout which has 4 columns and 3 rows.

Now the question is how we can create such responsive layouts using this Bootstrap new mobile first grid system. Let's start with the medium device that can be a laptop or normal desktop.

Since our medium device layout has 3 columns and 4 rows i.e. 3x4 grid layout, so the HTML code for making such grid structure would be something like this.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-md-4"><p>Box 1</p></div>
        <div class="col-md-4"><p>Box 2</p></div>
        <div class="col-md-4"><p>Box 3</p></div>
        <div class="col-md-4"><p>Box 4</p></div>
        <div class="col-md-4"><p>Box 5</p></div>
        <div class="col-md-4"><p>Box 6</p></div>
        <div class="col-md-4"><p>Box 7</p></div>
        <div class="col-md-4"><p>Box 8</p></div>
        <div class="col-md-4"><p>Box 9</p></div>
        <div class="col-md-4"><p>Box 10</p></div>
        <div class="col-md-4"><p>Box 11</p></div>
        <div class="col-md-4"><p>Box 12</p></div>
    </div>
</div>
```

If you see the output of the above example in a laptop or desktop having screen or viewport width greater than or equal to `992px` and less than `1200px` you will find it has 4 rows where each row has 3 equal columns resulting in 3x4 grid layout.

But just wait, the above example has a major alignment issue. If height of any column is taller than the other it doesn't clear properly and break the layout. To fix this, use the combination of a `.clearfix` class and the responsive utility classes.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-md-4"><p>Box 1</p></div>
        <div class="col-md-4"><p>Box 2</p></div>
        <div class="col-md-4"><p>Box 3</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-md-4"><p>Box 4</p></div>
        <div class="col-md-4"><p>Box 5</p></div>
        <div class="col-md-4"><p>Box 6</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-md-4"><p>Box 7</p></div>
        <div class="col-md-4"><p>Box 8</p></div>
        <div class="col-md-4"><p>Box 9</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-md-4"><p>Box 10</p></div>
```

```
            <div class="col-md-4"><p>Box 11</p></div>
            <div class="col-md-4"><p>Box 12</p></div>
        </div>
    </div>
```

Since the default grid system has 12 columns and in our layout sum of the every three column number, i.e. `col-md-*` is equal to 12 that's why we cleared columns after every third occurrence. In any other scenario where columns numbers are different for every column you should use `.clearfix` after the column that makes the complete 12 column grid.

Note: The responsive utility class `.visible-md-block` makes the `.clearfix` class effective only on medium size devices and it is hidden on other devices.

Now it's time to customize our layout for other devices. First customize it for tablet. Since inside the tablet our layout rendered as 2x6 grids (i.e. 2 columns and 6 rows). So, go ahead and add the class `.col-sm-6` on every column.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-sm-6 col-md-4"><p>Box 1</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 2</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 3</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 4</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 5</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 6</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 7</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 8</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 9</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 10</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 11</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 12</p></div>
    </div>
</div>
```

Now, since the sum of every two column number, i.e. `col-sm-*` is equal to 12, so clear floats after every second occurrence of columns.

After clearing floats for small devices our final code would be:

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-sm-6 col-md-4"><p>Box 1</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 2</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 3</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 4</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 5</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 6</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 7</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 8</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 9</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 10</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="col-sm-6 col-md-4"><p>Box 11</p></div>
        <div class="col-sm-6 col-md-4"><p>Box 12</p></div>
    </div>
</div>
```

Tip: For convenience choose your primary target device and create layout for that device first after that add classes to make it responsive for other devices.

Similarly, you can customize the layout for larger devices like a large desktop screen. Here's the final code after combining the whole process.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 1</p></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 2</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 3</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 4</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="clearfix visible-lg-block"></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 5</p></div>
```

```
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 6</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 7</p></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 8</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="clearfix visible-lg-block"></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 9</p></div>
        <div class="clearfix visible-md-block"></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 10</p></div>
        <div class="clearfix visible-sm-block"></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 11</p></div>
        <div class="col-sm-6 col-md-4 col-lg-3"><p>Box 12</p></div>
    </div>
</div>
```

Tip: According to the above illustration there is no need to customize the layout for extra small devices like mobile phones; since columns on extra small devices are always horizontal and rendered as one column grid layout by default.

Offsetting the Grid Columns

You can also move grid columns to the right for alignment purpose using the column offset classes like `.col-md-offset-*`, `.col-sm-offset-*`, etc.

These classes offset the columns by simply increasing its left margin by specified number of columns. For example, the class `.col-sm-offset-4` on the column `.col-sm-8` moves it to the right over four columns from its original position.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-sm-4"></div>
        <div class="col-sm-8"></div>
    </div>
    <div class="row">
        <div class="col-sm-8 col-sm-offset-4"><!--Column with 4 columns offset--></div>
    </div>
</div>
```

Nesting of Grid Columns

The Bootstrap grid columns are nestable, that means you can put rows and columns inside an existing column. However, the formula for placing the columns will be the same, i.e. the sum of column numbers should be equal to 12 or less within a single row.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-xs-8"><!--Column left--></div>
        <div class="col-xs-4">
            <!--Column right with nested rows and columns-->
            <div class="row">
                <div class="col-xs-12"></div>
            </div>
            <div class="row">
                <div class="col-xs-12"></div>
            </div>
        </div>
    </div>
</div>
```

Bootstrap Responsive Utilities Classes

You can use the following responsive classes to enable the visibility of elements on certain devices that screen sizes falls with the specific range.

As of v3.2.0, the `.visible-*-*` classes for each breakpoint come in three variations, one for each CSS `display` property value: `inline`, `block` and `inline-block`.

Class	Description
`.visible-xs-*`	Makes the element visible only on extra small devices having screen width less than `768px`. Hidden on others.
`.visible-sm-*`	Makes the element visible only on small devices having screen width greater than or equal to `768px` (i.e. `≥768px`) but less than `992px`. Hidden on others.
`.visible-md-*`	Makes the element visible only on medium devices having screen width greater than or equal to `992px` (i.e. `≥992px`) but less than `1200px`. Hidden on others.
`.visible-lg-*`	Makes the element visible only on large devices having screen width greater than or equal to `1200px` (i.e. `≥1200px`). Hidden on others.

Tip: You can also mix these classes to make the elements visible on multiple devices. For example, you can apply the class `.visible-xs-*` and `.visible-md-*` on any element to make it visible on extra small and medium devices.

Example

Try this code »

```
<p class="visible-xs-block">This paragraph is visible only on extra small devices.</p>
<p class="visible-sm-block">This paragraph is visible only on small devices.</p>
<p class="visible-md-block">This paragraph is visible only on medium devices.</p>
<p class="visible-lg-block">This paragraph is visible only on large devices.</p>
```

Similarly you can use these hidden utility classes to hide the elements on certain devices.

Class	Description
`.hidden-xs`	Hide the elements only on extra small devices having screen width less than `768px`. Visible on others.
`.hidden-sm`	Hide the elements only on small devices having screen width greater than or equal to `768px` (i.e. `≥768px`) but less than `992px`. Visible on others.
`.hidden-md`	Hide the elements only on medium devices having screen width greater than or equal to `992px` (i.e. `≥992px`) but less than `1200px`. Visible on others.
`.hidden-lg`	Hide the elements only on large devices having screen width greater than or equal to `1200px` (i.e. `≥1200px`). Visible on others.

Tip: You can also mix these classes to make the elements hidden on multiple devices. For example you can apply the class `.hidden-xs` and `.hidden-md` on any element to make it hidden on extra small and medium devices.

Example

Try this code »

```
<p class="hidden-xs">This paragraph is hidden only on extra small devices.</p>
<p class="hidden-sm">This paragraph is hidden only on small devices.</p>
<p class="hidden-md">This paragraph is hidden only on medium devices.</p>
```

- `<p class="hidden-lg">This paragraph is hidden only on large devices.</p>`

Similar to the regular responsive classes, you can use the following utility classes to show or hide certain elements for printing purpose or devices.

Class	Description
`.visible-print-block`	Hide block elements for browser rendering while visible for print.
`.visible-print-inline`	Hide inline elements for browser rendering while visible for print.
`.visible-print-inline-block`	Hide inline-block elements for browser rendering while visible for print.
`.hidden-print`	Hide elements for printing while visible on browser.

Previous Page Next Page

Bootstrap Fixed Layout

In this tutorial you will learn how to create fixed layouts with Bootstrap.

Creating Fixed Layout with Bootstrap

With Bootstrap 3 you can still create layouts of web pages based on fixed number of pixels, however this time it is responsive from the start as opposed to previous 2.x version where you need to include the responsive style sheet to make it responsive for other devices.

The process of creating the fixed yet responsive layout starts with the `.container` class. After that create rows with the `.row` class to wrap the horizontal groups of columns. Rows must be placed within a `.container` for proper alignment and padding.

Further columns can be created inside the rows using the predefined grid classes like `.col-xs-*`, `.col-sm-*`, `.col-md-*` and `.col-lg-*` where `*` represent grid number and should be from 1 to 12. Learn more about the Bootstrap grid system.

Note: Actual content like text, images, videos, etc. should be placed within columns, and only columns may be the immediate children of rows.

The following code creates a fixed width responsive layout that is `750px` pixels wide on small devices like tablet having screen width `≥768px`, whereas `970px` wide on medium devices like desktop and laptop having screen width `≥992px` and `1170px` wide on large devices like large

desktops having screen width `≥1200px`. However the layout width will be automatically calculated for devices that has screen width `<768px` like cell phones.

Example

Try this code »

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Bootstrap 3 Fixed Layout Example</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="css/bootstrap-theme.min.css">
<script type="text/javascript" src="https://code.jquery.com/jquery-1.11.3.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
</head>
<body>
    <nav id="myNavbar" class="navbar navbar-default navbar-inverse navbar-fixed-top" role="navigation">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbarCollapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Tutorial Republic</a>
            </div>
            <!-- Collect the nav links, forms, and other content for toggling -->
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="http://www.tutorialrepublic.com" target="_blank">Home</a></li>
                    <li><a href="http://www.tutorialrepublic.com/about-us.php" target="_blank">About</a></li>
                    <li><a href="http://www.tutorialrepublic.com/contact-us.php" target="_blank">Contact</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <div class="container">
        <div class="jumbotron">
```

```
            <h1>Learn to Create Websites</h1>
            <p>In today's world internet is the most popular way of connecting with the people. At <a href="http://www.tutorialrepublic.com" target="_blank">tutorialrepublic.com</a> you will learn the essential of web development technologies along with real life practice example, so that you can create your own website to connect with the people around the world.</p>
            <p><a href="http://www.tutorialrepublic.com" target="_blank" class="btn btn-success btn-lg">Get started today</a></p>
        </div>
        <div class="row">
            <div class="col-xs-4">
                <h2>HTML</h2>
                <p>HTML is a markup language that is used for creating web pages. The HTML tutorial section will help you understand the basics of HTML, so that you can create your own web pages or website.</p>
                <p><a href="http://www.tutorialrepublic.com/html-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
            <div class="col-xs-4">
                <h2>CSS</h2>
                <p>CSS is used for describing the presentation of web pages. The CSS tutorial section will help you learn the essentials of CSS, so that you can fine control the style and layout of your HTML document.</p>
                <p><a href="http://www.tutorialrepublic.com/css-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
            <div class="col-xs-4">
                <h2>Bootstrap</h2>
                <p>Bootstrap is a powerful front-end framework for faster and easier web development. The Bootstrap tutorial section will help you learn the techniques of Bootstrap so that you can quickly create your own website.</p>
                <p><a href="http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
        </div>
        <hr>
        <div class="row">
            <div class="col-xs-12">
                <footer>
                    <p>&copy; Copyright 2013 Tutorial Republic</p>
                </footer>
            </div>
        </div>
```

```
    </div>
</body>
</html>
```

— The output of the above example will look something like this:

Previous Page Next Page

Bootstrap Fluid Layout

In this tutorial you will learn how to create fluid layouts with Bootstrap.

Creating Fluid Layout with Bootstrap

In Bootstrap (version 3.2+), you can use the class `.container-fluid` to create the fluid layouts in order to utilize the 100% width of the viewport.

The class `.container-fluid` simply applies the horizontal margin with the value `auto` and left and right padding of `15px` on element to offset the left and right margin of `-15px` (i.e. `margin: 0 -15px;`) used on the `.row`.

The following code creates a fluid layout that covers `100%` width of the screen.

Example

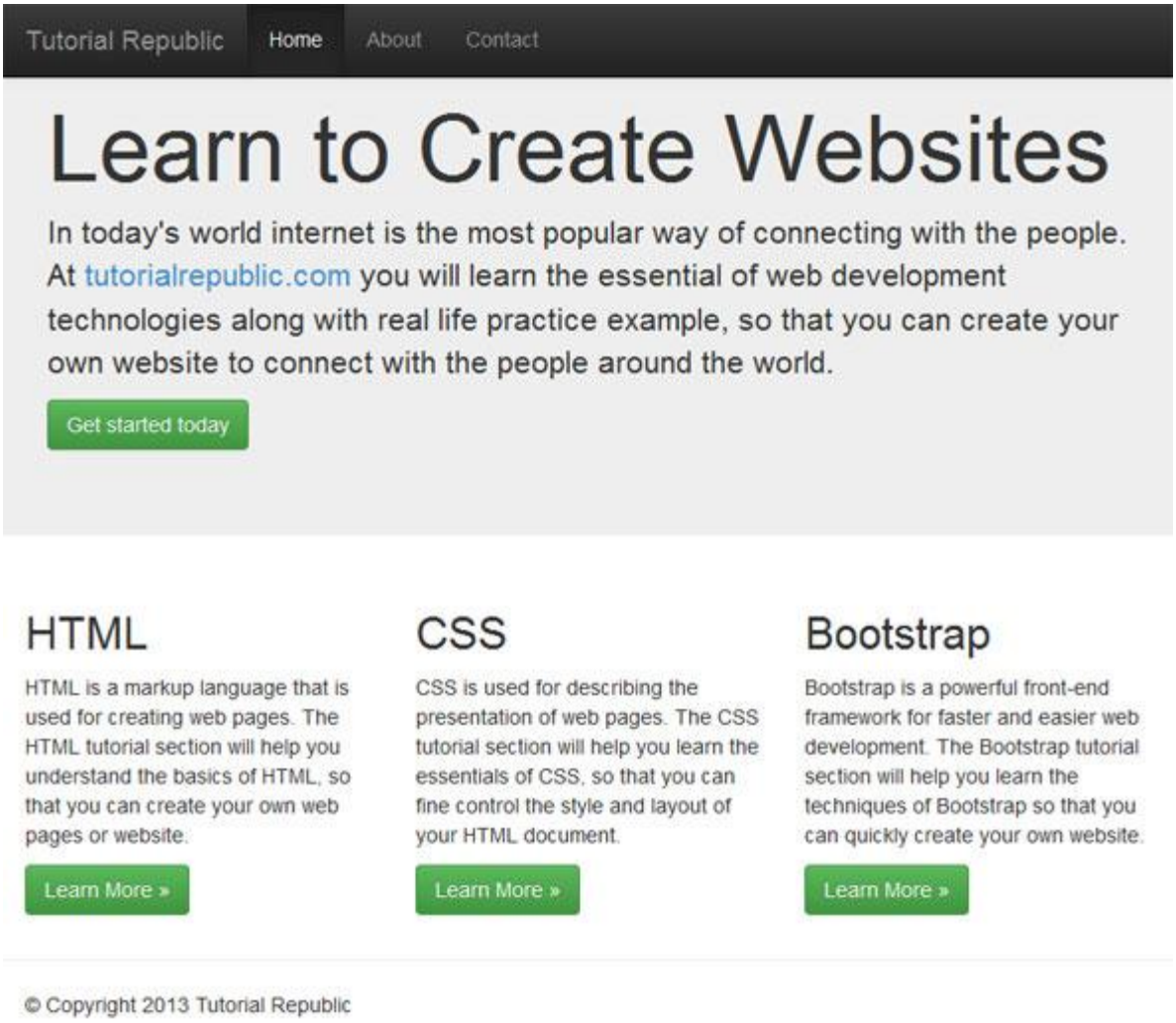
Try this code »

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Bootstrap 3 Fluid Layout Example</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="css/bootstrap-theme.min.css">
<link rel="stylesheet" type="text/css" href="css/custom.css">
<script type="text/javascript" src="https://code.jquery.com/jquery-1.11.3.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
</head>
<body>
    <nav id="myNavbar" class="navbar navbar-default navbar-inverse navbar-fixed-top" role="navigation">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="container-fluid">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbarCollapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Tutorial Republic</a>
            </div>
            <!-- Collect the nav links, forms, and other content for toggling -->
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="http://www.tutorialrepublic.com" target="_blank">Home</a></li>
                    <li><a href="http://www.tutorialrepublic.com/about-us.php" target="_blank">About</a></li>
                    <li><a href="http://www.tutorialrepublic.com/contact-us.php" target="_blank">Contact</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <div class="jumbotron">
        <div class="container-fluid">
            <h1>Learn to Create Websites</h1>
            <p>In today's world internet is the most popular way of connecting with the people. At <a
```

```
href="http://www.tutorialrepublic.com"
target="_blank">tutorialrepublic.com</a> you will learn the essential
of web development technologies along with real life practice example,
so that you can create your own website to connect with the people
around the world.</p>
            <p><a href="http://www.tutorialrepublic.com"
target="_blank" class="btn btn-success btn-lg">Get started
today</a></p>
        </div>
    </div>
    <div class="container-fluid">
        <div class="row">
            <div class="col-xs-4">
                <h2>HTML</h2>
                <p>HTML is a markup language that is used for creating
web pages. The HTML tutorial section will help you understand the
basics of HTML, so that you can create your own web pages or
website.</p>
                <p><a href="http://www.tutorialrepublic.com/html-
tutorial/" target="_blank" class="btn btn-success">Learn More
&raquo;</a></p>
            </div>
            <div class="col-xs-4">
                <h2>CSS</h2>
                <p>CSS is used for describing the presentation of web
pages. The CSS tutorial section will help you learn the essentials of
CSS, so that you can fine control the style and layout of your HTML
document.</p>
                <p><a href="http://www.tutorialrepublic.com/css-
tutorial/" target="_blank" class="btn btn-success">Learn More
&raquo;</a></p>
            </div>
            <div class="col-xs-4">
                <h2>Bootstrap</h2>
                <p>Bootstrap is a powerful front-end framework for
faster and easier web development. The Bootstrap tutorial section will
help you learn the techniques of Bootstrap so that you can quickly
create your own website.</p>
                <p><a href="http://www.tutorialrepublic.com/twitter-
bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More
&raquo;</a></p>
            </div>
        </div>
        <hr>
        <div class="row">
            <div class="col-xs-12">
                <footer>
                    <p>&copy; Copyright 2013 Tutorial Republic</p>
                </footer>
            </div>
        </div>
    </div>
```

- </body>
- </html>

— The output of the above example will look something like this:



Previous Page Next Page

Bootstrap Responsive Layout

In this tutorial you will learn how to create responsive web designs or layouts with Bootstrap framework.

What is Responsive Web Design or Layout

Responsive web design is a process of designing and building websites to provide better accessibility and optimal viewing experience to the user.

With the growing trend of smart phones and tablets, it has become almost unavoidable to ignore the optimization of sites for mobile devices. Responsive web design is a preferable alternative and an efficient way to target a wide range of devices with much less efforts.

Responsive layouts automatically adjust and adapts to any device screen size, whether it is a desktop, a laptop, a tablet, or a mobile phone.

Creating Responsive Layout with Bootstrap

With the new Bootstrap 3 mobile first grid system creating the responsive and mobile friendly websites has become much easier. As opposed to previous version you don't need to include any additional style sheet to enable responsive feature. Bootstrap 3 is responsive and mobile friendly from the start. Its four tiers grids classes provides better control over the layout as well as how it will be rendered on different types of devices like cell phones, tablets, desktop and laptops, large screen devices etc.

Example

Try this code »

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Bootstrap 3 Responsive Layout Example</title>
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href="css/bootstrap-
theme.min.css">
<script type="text/javascript" src="https://code.jquery.com/jquery-
1.11.3.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>
</head>
<body>
    <nav id="myNavbar" class="navbar navbar-default navbar-inverse
navbar-fixed-top" role="navigation">
```

```
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbarCollapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Tutorial Republic</a>
            </div>
            <!-- Collect the nav links, forms, and other content for toggling -->
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="http://www.tutorialrepublic.com" target="_blank">Home</a></li>
                    <li><a href="http://www.tutorialrepublic.com/about-us.php" target="_blank">About</a></li>
                    <li><a href="http://www.tutorialrepublic.com/contact-us.php" target="_blank">Contact</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <div class="container">
        <div class="jumbotron">
            <h1>Learn to Create Websites</h1>
            <p>In today's world internet is the most popular way of connecting with the people. At <a href="http://www.tutorialrepublic.com" target="_blank">tutorialrepublic.com</a> you will learn the essential of web development technologies along with real life practice example, so that you can create your own website to connect with the people around the world.</p>
            <p><a href="http://www.tutorialrepublic.com" target="_blank" class="btn btn-success btn-lg">Get started today</a></p>
        </div>
        <div class="row">
            <div class="col-sm-6 col-md-4 col-lg-2">
                <h2>HTML</h2>
                <p>HTML is a markup language that is used for creating web pages. The HTML tutorial section will help you understand the basics of HTML, so that you can create your own web pages or website.</p>
                <p><a href="http://www.tutorialrepublic.com/html-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
```

```
            <div class="col-sm-6 col-md-4 col-lg-2">
                <h2>CSS</h2>
                <p>CSS is used for describing the presentation of web pages. The CSS tutorial section will help you learn the essentials of CSS, so that you can fine control the style and layout of your HTML document.</p>
                <p><a href="http://www.tutorialrepublic.com/css-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
            <div class="clearfix visible-sm-block"></div>
            <div class="col-sm-6 col-md-4 col-lg-2">
                <h2>Bootstrap</h2>
                <p>Bootstrap is a powerful front-end framework for faster and easier web development. The Bootstrap tutorial section will help you learn the techniques of Bootstrap so that you can create web your own website with much less efforts.</p>
                <p><a href="http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
            <div class="clearfix visible-md-block"></div>
            <div class="col-sm-6 col-md-4 col-lg-2">
                <h2>References</h2>
                <p>The references section outlines all the standard HTML tags and CSS properties along with other useful references such as color names and values, symbols and character entities, web safe fonts, language codes, HTTP messages and much more.</p>
                <p><a href="http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
            <div class="clearfix visible-sm-block"></div>
            <div class="col-sm-6 col-md-4 col-lg-2">
                <h2>Examples</h2>
                <p>The examples section encloses an extensive collection of examples on various topic that you can try and test yourself using online HTML editor.</p>
                <p><a href="http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
            <div class="col-sm-6 col-md-4 col-lg-2">
                <h2>FAQ</h2>
                <p>The collection of Frequently Asked Questions (FAQ) provides brief answers to many common questions related to web design and development.</p>
                <p><a href="http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank" class="btn btn-success">Learn More &raquo;</a></p>
            </div>
        </div>
```

```
        <hr>
        <div class="row">
            <div class="col-sm-12">
                <footer>
                    <p>&copy; Copyright 2013 Tutorial Republic</p>
                </footer>
            </div>
        </div>
    </div>
</body>
</html>
```

Tip: Open the output of this example in new window and resize the screen you will see the orientation of content boxes changes when viewport width crossing or approaching the certain limit (i.e. breakpoints).

Previous Page Next Page

Bootstrap Typography

In this tutorial you will learn about the styling and formatting of text content like headings, paragraphs, blockquotes, etc. with Bootstrap.

Working with Headings

You can define all HTML headings, `<h1>` through `<h6>` — In the same way you define in simple HTML document. You can also utilize the heading classes `.h1` through `.h6` on other elements, if you want to apply the style on element's text same as headings.

Example

Try this code »

```
<h1>h1. Bootstrap heading</h1>
<h2>h2. Bootstrap heading</h2>
<h3>h3. Bootstrap heading</h3>
<h4>h4. Bootstrap heading</h4>
<h5>h5. Bootstrap heading</h5>
<h6>h6. Bootstrap heading</h6>
```

— The output of the above example will look something like this:

h1. Bootstrap heading

h2. Bootstrap heading

h3. Bootstrap heading

h4. Bootstrap heading

h5. Bootstrap heading

h6. Bootstrap heading

Moreover you can use the `<small>` tag or `<span>` tag with `.small` class to display the secondary text of any heading in a smaller and lighter variation.

Example

Try this code »

```
<h1>h1. Bootstrap heading <small>Secondary text</small></h1>
<h2>h2. Bootstrap heading <small>Secondary text</small></h2>
<h3>h3. Bootstrap heading <small>Secondary text</small></h3>
<h4>h4. Bootstrap heading <small>Secondary text</small></h4>
<h5>h5. Bootstrap heading <small>Secondary text</small></h5>
<h6>h6. Bootstrap heading <small>Secondary text</small></h6>
```

— The output of the above example will look something like this:

h1. Bootstrap heading Secondary text

h2. Bootstrap heading Secondary text

h3. Bootstrap heading Secondary text

h4. Bootstrap heading Secondary text

h5. Bootstrap heading Secondary text

h6. Bootstrap heading Secondary text

Creating Page Headers

You can make your `<h1>` heading appear differently than rest of the headings on a page using the page header component. You can also utilize `<small>` tag to mark header subtext.

Example

Try this code »

```
<div class="page-header">
    <h1>Bootstrap <small>An intuitive front-end framework</small></h1>
</div>
```

— The output of the above example will look something like this:

Working with Paragraphs

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 to that a bottom margin of half their `line-height` (`10px` by default) is applied to the all paragraphs (`<p>`).

You can also make a paragraph stand out by just adding the class `.lead`.

Example

Try this code »

```
<p>This is how a normal paragraph looks like in Bootstrap.</p>
<p class="lead">This is how a paragraph stands out in Bootstrap.</p>
```

— The HTML code in the above examples will produce the following result:

This is how a normal paragraph looks like in Twitter Bootstrap.

This is how a paragraph stands out in Twitter Bootstrap.

You can easily align text inside a paragraph and other elements using text alignment classes.

Example

Try this code »

```
<p class="text-left">Left aligned text.</p>
<p class="text-center">Center aligned text.</p>
<p class="text-right">Right aligned text.</p>
<p class="text-justify">Justified text.</p>
<p class="text-nowrap">No wrap text.</p>
```

Bootstrap Text Formatting

You are free to use text formatting tags like `<strong>`, `<i>`, `<small>` to make your text bold, italic, small and so on, in the same way you do in simple HTML.

Example

Try this code »

```
<p><b>This is bold text</b></p>
<p><big>This is big text</big></p>
<p><code>This is computer code</code></p>
<p><em>This is emphasized text</em></p>
<p><i>This is italic text</i></p>
<p><mark>This is highlighted text</mark></p>
<p><small>This is small text</small></p>
<p><strong>This is strongly emphasized text</strong></p>
<p>This is <sub>subscript</sub> and <sup>superscript</sup></p>
<p><ins>This text is inserted to the document</ins></p>
<p><del>This text is deleted from the document</del></p>
```

— The output of the above example will look something like this:

This is bold text

This is big text

`This is computer code`

This is emphasized text

This is italic text

This is highlighted text

This is small text

This is strongly emphasized text

This is subscript and superscript

This text is inserted to the document

~~This text is deleted from the document~~

Text Transformation Classes

You can also transform the text to lowercase, uppercase or make them capitalize.

Example

Try this code »

```
<p class="text-lowercase">The quick brown fox jumps over the lazy dog.</p>
<p class="text-uppercase">The quick brown fox jumps over the lazy dog.</p>
<p class="text-capitalize">The quick brown fox jumps over the lazy dog.</p>
```

— The output of the above example will look something like this:

the quick brown fox jumps over the lazy dog.

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG.

The Quick Brown Fox Jumps Over The Lazy Dog.

Text Emphasis Classes

Colors are the powerful method of conveying important information in website design.

Bootstrap has handful of emphasis utility classes that can be used for this purpose such as showing success message in green color, warning or error message in red color, etc.

Example

Try this code »

```
<p class="text-muted">Muted: This text is grayed out.</p>
<p class="text-primary">Important: Please read the instructions carefully before proceeding.</p>
<p class="text-success">Success: Your message has been sent successfully.</p>
<p class="text-info">Note: You must agree with the terms and conditions to complete the sign up process.</p>
<p class="text-warning">Warning: There was a problem with your network connection.</p>
<p class="text-danger">Error: An error has been occurred while submitting your data.</p>
```

— The output of the above example will look something like this:

Muted: This text is grayed out.

Important: Please read the instructions carefully before proceeding.

Success: Your message has been sent successfully.

Note: You must agree with the terms and conditions to complete the sign up process.

Warning: There was a problem with your network connection.

Error: An error has been occurred while submitting your data.

Styling Blockquotes

You can also give pretty look to your blockquotes — Just define the blockquotes using the standard `<blockquote>` element and bootstrap's style sheet will do the rest.

Example

Try this code »

```
<blockquote>
    <p>The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.</p>
    <small>by <cite>Albert Einstein</cite></small>
</blockquote>
```

— The output of the above example will look something like this:

> The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.
>
> — by Albert Einstein

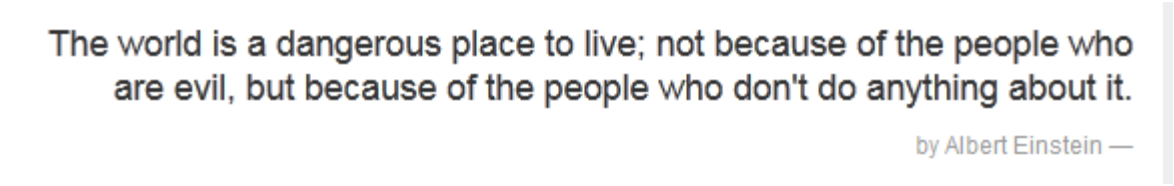
Alternatively, you can right-align the blockquote through floating it to right by simply applying the class `.pull-right` on the `<blockquote>` element.

Example

Try this code »

```
<blockquote class="pull-right">
    <p>The world is a dangerous place to live; not because of the people who are evil, but because of the people who don't do anything about it.</p>
    <small>by <cite>Albert Einstein</cite></small>
</blockquote>
```

— The output of the above example will look something like this:



Previous Page Next Page

Bootstrap Tables

In this tutorial you will learn how to create elegant tables with Bootstrap.

What is Table

The HTML tables are used to present data in grid manner like row and columns. Using Bootstrap you can greatly improve the appearance of table in a simple way.

See the tutorial on HTML Tables to learn more about tables.

Creating a Simple Table with Bootstrap

You can create tables with basic styling that has horizontal dividers and small cell padding (`8px` by default), by just adding the Bootstrap's class `.table` to the `<table>` element.

Example

Try this code »

```
<table class="table">
    <thead>
        <tr>
            <th>Row</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>John</td>
            <td>Carter</td>
            <td>johncarter@mail.com</td>
        </tr>
        <tr>
```

```
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
        </tr>
        <tr>
            <td>3</td>
            <td>John</td>
            <td>Rambo</td>
            <td>johnrambo@mail.com</td>
        </tr>
    </tbody>
</table>
```

— The output of the above example will look something like this:

Row	First Name	Last Name	Email
1	John	Carter	johncarter@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Rambo	johnrambo@mail.com

Tables with Striped Rows

You can create table with alternate background like zebra-stripes by simply adding the Bootstrap's class `.table-striped` to the `.table` base class.

This is achieved by adding the background-color to the table row within `<tbody>` element via the `:nth-child` CSS selector (not supported in IE7-8).

Example

Try this code »

```
<table class="table table-striped">
    <thead>
        <tr>
            <th>Row</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
```

```
            <tr>
                <td>1</td>
                <td>John</td>
                <td>Carter</td>
                <td>johncarter@mail.com</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Peter</td>
                <td>Parker</td>
                <td>peterparker@mail.com</td>
            </tr>
            <tr>
                <td>3</td>
                <td>John</td>
                <td>Rambo</td>
                <td>johnrambo@mail.com</td>
            </tr>
        </tbody>
    </table>
```

— The output of the above example will look something like this:

Row	First Name	Last Name	Email
1	John	Carter	johncarter@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Rambo	johnrambo@mail.com

Table with Borders on All Sides

You can also add borders to the all table cells by adding an extra Bootstrap's class `.table-bordered` to the `.table` base class.

Example

Try this code »

```
<table class="table table-bordered">
    <thead>
        <tr>
            <th>Row</th>
            <th>First Name</th>
            <th>Last Name</th>
```

```
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>John</td>
            <td>Carter</td>
            <td>johncarter@mail.com</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
        </tr>
        <tr>
            <td>3</td>
            <td>John</td>
            <td>Rambo</td>
            <td>johnrambo@mail.com</td>
        </tr>
    </tbody>
</table>
```

— The output of the above example will look something like this:

Row	First Name	Last Name	Email
1	John	Carter	johncarter@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Rambo	johnrambo@mail.com

Enable Hover State on Table Rows

You can also enable a hover state on table rows within a `<tbody>` element by adding the Bootstrap's class `.table-hover` to the `.table` base class.

Example

Try this code »

```
<table class="table table-hover">
    <thead>
```

```
            <tr>
                <th>Row</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Email</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>John</td>
                <td>Carter</td>
                <td>johncarter@mail.com</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Peter</td>
                <td>Parker</td>
                <td>peterparker@mail.com</td>
            </tr>
            <tr>
                <td>3</td>
                <td>John</td>
                <td>Rambo</td>
                <td>johnrambo@mail.com</td>
            </tr>
        </tbody>
    </table>
```

— The output of the above example will look something like this:

Row	First Name	Last Name	Email
1	John	Carter	johncarter@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Rambo	johnrambo@mail.com

Condensed or Compact Table

You can also make your tables more compact and save the space through adding an extra class `.table-condensed` to the `.table` base class. The class `.table-condensed` makes the table compact by cutting the cell padding in half.

Example

Try this code »

```
<table class="table table-condensed">
    <thead>
        <tr>
            <th>Row</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>1</td>
            <td>John</td>
            <td>Carter</td>
            <td>johncarter@mail.com</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Peter</td>
            <td>Parker</td>
            <td>peterparker@mail.com</td>
        </tr>
        <tr>
            <td>3</td>
            <td>John</td>
            <td>Rambo</td>
            <td>johnrambo@mail.com</td>
        </tr>
    </tbody>
</table>
```

— The output of the above example will look something like this:

Row	First Name	Last Name	Email
1	John	Carter	johncarter@mail.com
2	Peter	Parker	peterparker@mail.com
3	John	Rambo	johnrambo@mail.com

Optional Emphasis Classes for Table Rows

There are some contextual classes to emphasize the row or individual cells data like success, warning, danger, etc. through coloring its background.

Example

Try this code »

```
<table class="table">
    <thead>
        <tr>
            <th>Row</th>
            <th>Bill</th>
            <th>Payment Date</th>
            <th>Payment Status</th>
        </tr>
    </thead>
    <tbody>
        <tr class="active">
            <td>1</td>
            <td>Credit Card</td>
            <td>04/07/2014</td>
            <td>Call in to confirm</td>
        </tr>
        <tr class="success">
            <td>2</td>
            <td>Water</td>
            <td>01/07/2014</td>
            <td>Paid</td>
        </tr>
        <tr class="info">
            <td>3</td>
            <td>Internet</td>
            <td>05/07/2014</td>
            <td>Change plan</td>
        </tr>
        <tr class="warning">
            <td>4</td>
            <td>Electricity</td>
            <td>03/07/2014</td>
            <td>Pending</td>
        </tr>
        <tr class="danger">
            <td>5</td>
            <td>Telephone</td>
            <td>06/07/2014</td>
            <td>Due</td>
        </tr>
    </tbody>
</table>
```

— The output of the above example will look something like this:

Row	Bill	Payment Date	Payment Status
1	Credit Card	04/07/2014	Call in to confirm
2	Water	01/07/2014	Paid
3	Internet	05/07/2014	Change plan
4	Electricity	03/07/2014	Pending
5	Telephone	06/07/2014	Due

Creating Responsive Tables with Bootstrap

With Bootstrap 3 you can also create responsive tables to enable horizontal scrolling on small devices (screen width under `768px`). However, viewing responsive tables on other devices having screen width larger than `768px`, you will not see any difference.

To make any table responsive just place the table inside a `<div>` element and apply the class `.table-responsive` on it, as demonstrated in the example below:

Example

Try this code »

```
<div class="table-responsive">
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Row</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Email</th>
                <th>Biography</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>John</td>
                <td>Carter</td>
                <td>johncarter@mail.com</td>
                <td>Lorem ipsum dolor sit amet…</td>
```

```
            </tr>
            <tr>
                <td>2</td>
                <td>Peter</td>
                <td>Parker</td>
                <td>peterparker@mail.com</td>
                <td>Vestibulum consectetur scelerisque…</td>
            </tr>
            <tr>
                <td>3</td>
                <td>John</td>
                <td>Rambo</td>
                <td>johnrambo@mail.com</td>
                <td>Integer pulvinar leo id risus…</td>
            </tr>
        </tbody>
    </table>
</div>
```

Supported Table Elements in Bootstrap

The following table lists the supported HTML table elements and how they should be used.

Tag	Description
`<table>`	Wrapper element for displaying data in a tabular format.
`<caption>`	The title or summary of what the table holds.
`<thead>`	Container element for table header rows (`<tr>`) defines headings for table columns.
`<tbody>`	Container element for table rows (`<tr>`) that defines the body of a table.
`<tr>`	Container element for a set of table cells (`<td>` or `<th>`) that appears in a single row.
`<th>`	Special table cell for column headings.
`<td>`	Default table cell for placing data.

Previous Page Next Page

Bootstrap Lists

In this tutorial you will learn how to style HTML lists with Bootstrap.

Creating Lists With Bootstrap

You can define the three different types of lists:

- **Unordered lists —** A list of items in which the order does not explicitly matter. The list items in unordered lists are marked with bullets.
- **Ordered lists —** A list of items in which the order does explicitly matter. The list items in ordered lists are marked with numbers.
- **Definition list —** A list of terms with their associated descriptions.

See the tutorial on HTML Lists, to learn more about different lists types.

Unstyled Ordered and Unordered Lists

Sometimes you might need to remove the default styling form the list items. You can do this by simply applying the class `.list-unstyled` to the respective `<ul>` or `<ol>` elements.

Example

Try this code »

```
<ul class="list-unstyled">
  <li>Home</li>
  <li>Products
    <ul>
      <li>Gadgets</li>
      <li>Accessories</li>
    </ul>
  </li>
  <li>About Us</li>
  <li>Contact</li>
</ul>
```

— The output of the above example will look something like this:

Home
Products
- Gadgets
- Accessories

About Us
Contact

Note: The `.list-unstyled` class removes the default `list-style` and left `padding` only from the list items which are immediate children of the `<ul>` or `<ol>` element.

Placing Ordered and Unordered List Items Inline

If you want to create a horizontal menu using ordered or unordered list you need to place all list items in a single line i.e. side by side. You can do this by simply applying the Bootstrap's class `.list-inline` to the respective `<ul>` or `<ol>` elements. The `.list-inline` class also adds some left and right padding (`5px` by default) to the all list items.

Example

Try this code »

```
<ul class="list-inline">
    <li>Home</li>
    <li>Products</li>
    <li>About Us</li>
    <li>Contact</li>
</ul>
```

— The output of the above example will look something like this:

Home Products About Us Contact

Creating Horizontal Definition Lists

The terms and descriptions in definition lists can also be placed side-by-side using the Bootstrap's class `.dl-horizontal`. The terms in horizontal definition lists will be truncated if is too long to fit in the left column (`160px` by default), however in narrower viewports they will change to the default stacked layout.

Example

Try this code »

```
<dl class="dl-horizontal">
    <dt>User Agent</dt>
    <dd>An HTML user agent is any device that interprets HTML documents.</dd>
    <dt>Client-side Scripting</dt>
    <dd>Client-side scripting generally refers to the category of computer programs on the web that are executed client-side i.e. by the user's web browser.</dd>
    <dt>Document Tree</dt>
```

```
    <dd>The tree of elements encoded in the source document.</dd>
</dl>
```

— The output of the above example will look something like this:

User Agent An HTML user agent is any device that interprets HTML documents.
Client-side Scripting Client-side scripting generally refers to the category of computer programs on the web that are executed client-side i.e. by the user's web browser.
Document Tree The tree of elements encoded in the source document.

Note: The terms clipped in horizontal definition lists will be indicated by an ellipsis (…) using the `text-overflow` CSS property.

In the next chapter you will learn how to create even more flexible and complex list of elements using the Bootstrap list group component.

Previous Page Next Page

Bootstrap List Groups

In this tutorial you will learn how to use Bootstrap list group component.

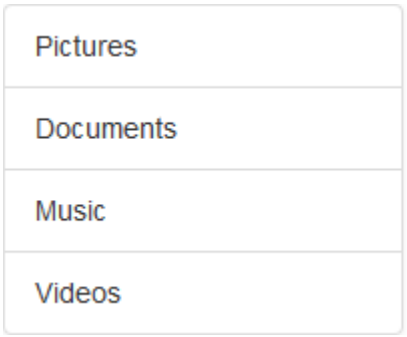
Creating List Groups with Bootstrap

The list groups are very useful and flexible component for displaying lists of elements in a beautiful manner. In most basic form a list group is simply an unordered list with the class `.list-group` whereas, the list items having the class `.list-group-item`.

Example
Try this code »

```
<ul class="list-group">
    <li class="list-group-item">Pictures</li>
    <li class="list-group-item">Documents</li>
    <li class="list-group-item">Music</li>
    <li class="list-group-item">Videos</li>
</ul>
```

— The output of the above example will look something like this:



List Group with Linked Items

You can also hyperlink list group items with the little change in HTML markup.

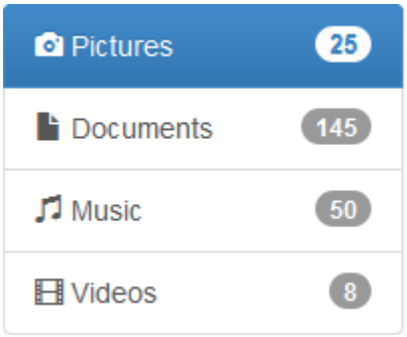
Just replace the `<li>` with `<a>` tag and use `<div>` element as a parent instead of `<ul>`. You can also add icons and badges to this list group to make it more elegant. The badges component will automatically be positioned on the right.

Example

Try this code »

```
<div class="list-group">
    <a href="#" class="list-group-item active">
        <span class="glyphicon glyphicon-camera"></span> Pictures <span class="badge">25</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-file"></span> Documents <span class="badge">145</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-music"></span> Music <span class="badge">50</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-film"></span> Videos <span class="badge">8</span>
    </a>
</div>
```

— The output of the above example will look something like this:



Tip: You can use the Bootstrap list group component for creating the sidebar navigation menu, e.g. displaying the list of products or categories on your website.

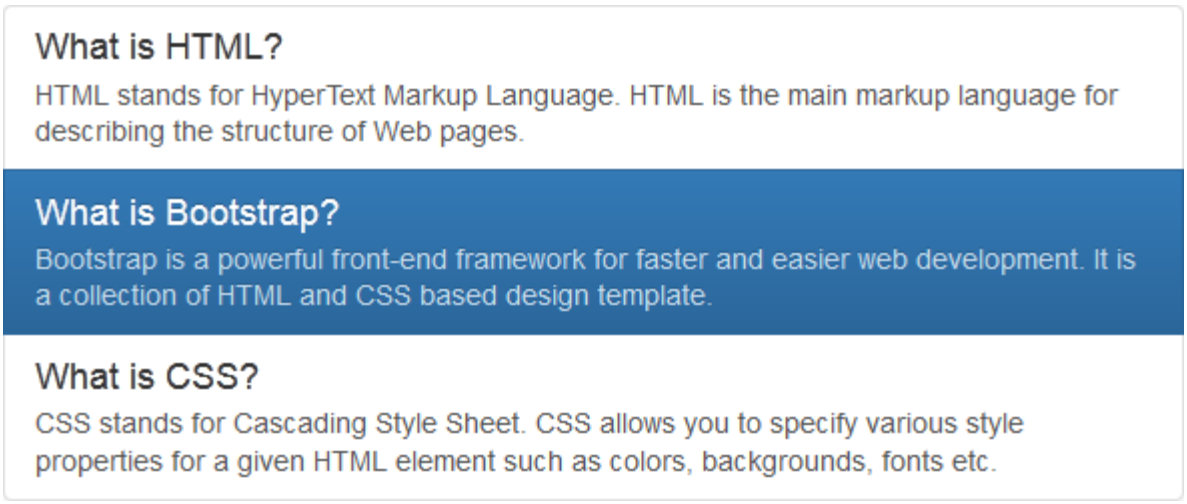
You can also add other HTML elements like headings and paragraph within these list groups.

Example

Try this code »

```
<div class="list-group">
    <a href="#" class="list-group-item">
        <h4 class="list-group-item-heading">What is HTML?</h4>
        <p class="list-group-item-text">HTML stands for HyperText Markup Language. HTML is the main markup language for describing the structure of Web pages.</p>
    </a>
    <a href="#" class="list-group-item active">
        <h4 class="list-group-item-heading">What is Bootstrap?</h4>
        <p class="list-group-item-text">Bootstrap is a powerful front-end framework for faster and easier web development. It is a collection of HTML and CSS based design template.</p>
    </a>
    <a href="#" class="list-group-item">
        <h4 class="list-group-item-heading">What is CSS?</h4>
        <p class="list-group-item-text">CSS stands for Cascading Style Sheet. CSS allows you to specify various style properties for a given HTML element such as colors, backgrounds, fonts etc.</p>
    </a>
</div>
```

— The output of the above example will look something like this:



List Group with Contextual States

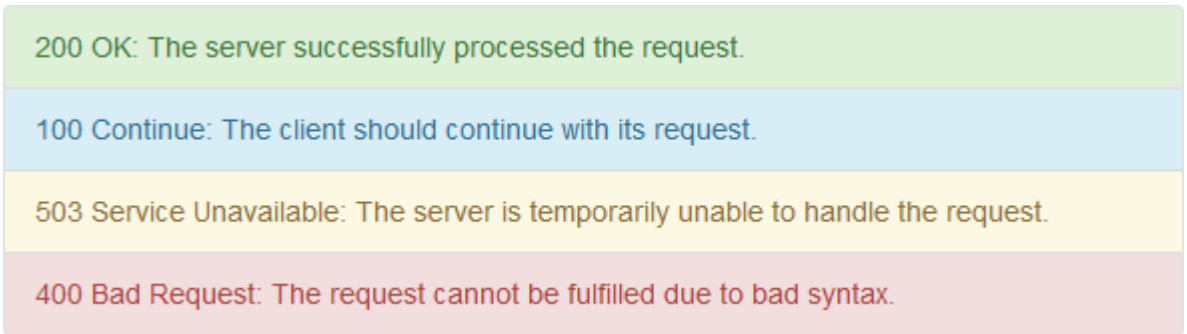
Like most of the other components you can also use contextual classes on the list group items to apply extra emphasis on them, like this:

Example

Try this code »

```
<ul class="list-group">
    <li class="list-group-item list-group-item-success">200 OK: The server successfully processed the request.</li>
    <li class="list-group-item list-group-item-info">100 Continue: The client should continue with its request.</li>
    <li class="list-group-item list-group-item-warning">503 Service Unavailable: The server is temporarily unable to handle the request.</li>
    <li class="list-group-item list-group-item-danger">400 Bad Request: The request cannot be fulfilled due to bad syntax.</li>
</ul>
```

— The output of the above example will look something like this:



Similarly, you can use these contextual classes to the linked list group items. You can also use the class `.active` to specify the active list group item.

Example

Try this code »

```
<div class="list-group">
    <a href="#" class="list-group-item list-group-item-success">200 OK: The server successfully processed the request.</a>
    <a href="#" class="list-group-item list-group-item-info active">100 Continue: The client should continue with its request.</a>
    <a href="#" class="list-group-item list-group-item-warning">503 Service Unavailable: The server is temporarily unable to handle the request.</a>
    <a href="#" class="list-group-item list-group-item-danger">400 Bad Request: The request cannot be fulfilled due to bad syntax.</a>
</div>
```

Previous Page Next Page

Bootstrap Forms

In this tutorial you will learn how to create elegant forms with Bootstrap.

Creating Forms with Bootstrap

HTML forms are the integral part of the web pages and applications, but styling the form controls manually one by one with CSS are often boring and tedious. Bootstrap greatly simplifies the process of styling and alignment of form controls like labels, input fields, selectboxes, textareas, buttons, etc. through predefined set of classes.

Bootstrap provides three different types of form layouts:

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

The following section will give you the detailed overview of these form layouts as well as the various form related Bootstrap components one by one.

Creating Vertical Form Layout

This is the default Bootstrap form layout in which styles are applied to form controls without adding any base class to the `<form>` element or any large changes in the markup.

The form controls in this layout are stacked with left-aligned labels on the top.

Example

Try this code »

```
<form>
    <div class="form-group">
        <label for="inputEmail">Email</label>
        <input type="email" class="form-control" id="inputEmail"
placeholder="Email">
    </div>
    <div class="form-group">
        <label for="inputPassword">Password</label>
        <input type="password" class="form-control" id="inputPassword"
placeholder="Password">
    </div>
    <div class="checkbox">
        <label><input type="checkbox"> Remember me</label>
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
</form>
```

— The output of the above example will look something like this:

Note: In Bootstrap 3 all textual elements like `<input>`, `<textarea>`, and `<select>` with the class `.form-control` are `100%` wide by default. To use them inline, you'll have to set a width on the element the form controls used within.

Creating Horizontal Form Layout

In horizontal form layout labels are right aligned and floated to left to make them appear on the same line as form controls. The horizontal form layout requires the various markup changes from a default form layout. Steps to achieve this layout are listed below:

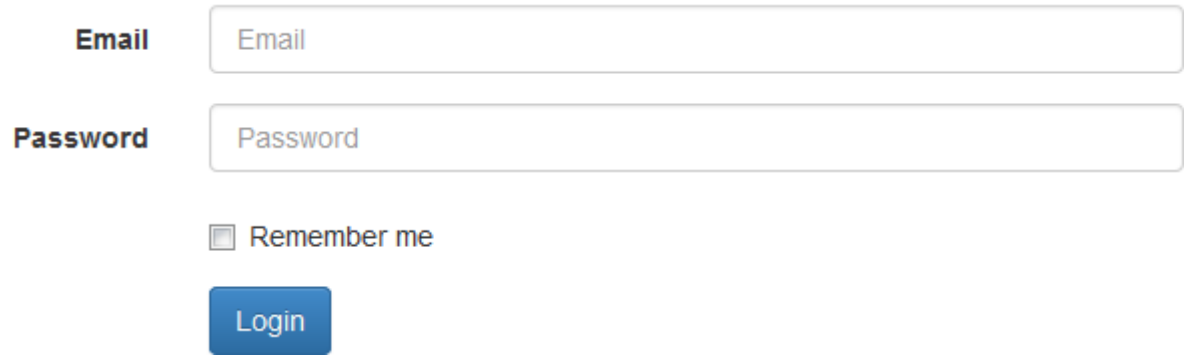
- Add the class `.form-horizontal` to the `<form>` element.
- Wrap labels and form controls in a `<div>` element and apply the class `.form-group`.
- Use Bootstrap's predefined grid classes to align labels and form controls.
- Add the class `.control-label` to the `<label>` element.

Example

Try this code »

```
<form class="form-horizontal">
    <div class="form-group">
        <label for="inputEmail" class="control-label col-xs-2">Email</label>
        <div class="col-xs-10">
            <input type="email" class="form-control" id="inputEmail" placeholder="Email">
        </div>
    </div>
    <div class="form-group">
        <label for="inputPassword" class="control-label col-xs-2">Password</label>
        <div class="col-xs-10">
            <input type="password" class="form-control" id="inputPassword" placeholder="Password">
        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-offset-2 col-xs-10">
            <div class="checkbox">
                <label><input type="checkbox"> Remember me</label>
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-offset-2 col-xs-10">
            <button type="submit" class="btn btn-primary">Login</button>
        </div>
    </div>
</form>
```

— The output of the above example will look something like this:



Creating Inline Form Layout

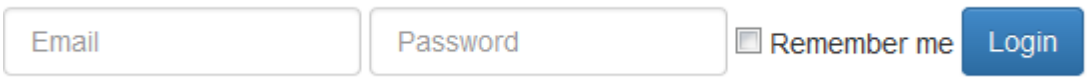
Sometimes you might require to place the form controls side-by-side to compact the layout. You can do this easily by adding the Bootstrap class `.form-inline` to the `<form>` element.

Example

Try this code »

```
<form class="form-inline">
    <div class="form-group">
        <label class="sr-only" for="inputEmail">Email</label>
        <input type="email" class="form-control" id="inputEmail" placeholder="Email">
    </div>
    <div class="form-group">
        <label class="sr-only" for="inputPassword">Password</label>
        <input type="password" class="form-control" id="inputPassword" placeholder="Password">
    </div>
    <div class="checkbox">
        <label><input type="checkbox"> Remember me</label>
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
</form>
```

— The output of the above example will look something like this:



Note: It is recommended to include a label for every form inputs otherwise screen readers will have trouble with your forms. However in case of inline form layout you can hide these labels using the `.sr-only` class.

Static Form Control

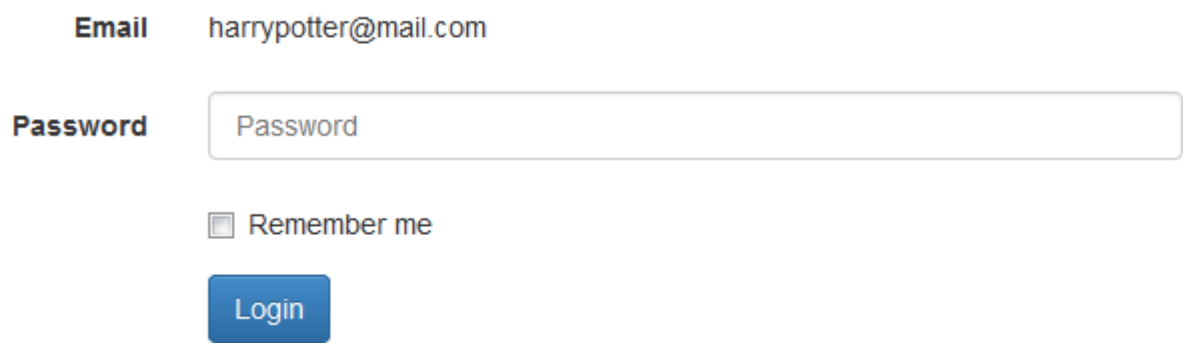
There might be a situation when you need to place just plain text next to a form label instead of a form control. You can do this within a horizontal form layout by using the `.form-control-static` class on a `<p>` element, like this:

Example

Try this code »

```
<form class="form-horizontal">
    <div class="form-group">
        <label for="inputEmail" class="control-label col-xs-2">Email</label>
        <div class="col-xs-10">
            <p class="form-control-static">harrypotter@mail.com</p>
        </div>
    </div>
    <div class="form-group">
        <label for="inputPassword" class="control-label col-xs-2">Password</label>
        <div class="col-xs-10">
            <input type="password" class="form-control" id="inputPassword" placeholder="Password">
        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-offset-2 col-xs-10">
            <div class="checkbox">
                <label><input type="checkbox"> Remember me</label>
            </div>
        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-offset-2 col-xs-10">
            <button type="submit" class="btn btn-primary">Login</button>
        </div>
    </div>
</form>
```

— The output of the above example will look something like this:



Height Sizing of Inputs and Select Boxes

You can easily control the height of your input and select boxes to match the button sizes. The Bootstrap's relative sizing classes like `.input-lg`, `.input-sm` can be used both on `<input>` and `<select>` boxes to create it's larger or smaller sizes.

Example

Try this code »

```
<form>
    <div class="row">
        <div class="col-xs-6">
            <input type="text" class="form-control input-lg"
placeholder="Large input">
        </div>
        <div class="col-xs-6">
            <select class="form-control input-lg">
                <option>Large select</option>
            </select>
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-6">
            <input type="text" class="form-control"
placeholder="Default input">
        </div>
        <div class="col-xs-6">
            <select class="form-control">
                <option>Default select</option>
            </select>
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-6">
```

```
            <input type="text" class="form-control input-sm"
placeholder="Small input">
        </div>
        <div class="col-xs-6">
            <select class="form-control input-sm">
                <option>Small select</option>
            </select>
        </div>
    </div>
</form>
```

Column Sizing of Inputs, Textareas and Select Boxes

You can also match the sizes of your form controls to the Bootstrap grid column sizes. Just wrap your form controls (i.e. `<input>`, `<textarea>`, and `<select>`) in grid columns, or any custom element and apply the grid classes on it.

Example

Try this code »

```
<form>
    <div class="row">
        <div class="col-xs-3">
            <input type="text" class="form-control">
        </div>
        <div class="col-xs-4">
            <input type="text" class="form-control">
        </div>
        <div class="col-xs-5">
            <input type="text" class="form-control">
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-3">
            <textarea class="form-control"></textarea>
        </div>
        <div class="col-xs-4">
            <textarea class="form-control"></textarea>
        </div>
        <div class="col-xs-5">
            <textarea class="form-control"></textarea>
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-3">
```

```
            <select class="form-control">
                <option>Select</option>
            </select>
        </div>
        <div class="col-xs-4">
            <select class="form-control">
                <option>Select</option>
            </select>
        </div>
        <div class="col-xs-5">
            <select class="form-control">
                <option>Select</option>
            </select>
        </div>
    </div>
</form>
```

Height Sizing of Form Groups

You can add the relative form sizing classes such as `.form-group-lg` or `.form-group-sm` to the `.form-group` itself to make both the labels and form controls larger or smaller at the same time within the horizontal form layouts.

Example

Try this code »

```
<form class="form-horizontal">
    <div class="form-group form-group-lg">
        <label class="col-sm-3 control-label" for="inputLarge">Large label</label>
        <div class="col-sm-9">
            <input type="text" class="form-control" id="inputLarge" placeholder="Large input">
        </div>
    </div>
    <div class="form-group">
        <label class="col-sm-3 control-label" for="inputDefault">Default label</label>
        <div class="col-sm-9">
            <input type="text" class="form-control" id="inputDefault" placeholder="Default input">
        </div>
    </div>
    <div class="form-group form-group-sm">
        <label class="col-sm-3 control-label" for="inputSmall">Small label</label>
        <div class="col-sm-9">
```

```
            <input type="text" class="form-control" id="inputSmall"
placeholder="Small input">
        </div>
    </div>
</form>
```

Creating Disabled and Readonly Inputs

To create disabled inputs just add the attributes `disabled` to the `<input>` element and Bootstrap will do the rest.

Example

Try this code »

```
<form>
    <input type="text" class="form-control" placeholder="Disabled
input" disabled="disabled">
</form>
```

— The output of the above example will look something like this:

Similarly, you can add the `readonly` attribute on input elements to create read only input controls that prevent user inputs and give the style same as disabled.

Example

Try this code »

```
<form>
    <input type="text" class="form-control" placeholder="Readonly
input" readonly="readonly">
</form>
```

Creating Disabled Fieldsets

Rather than disabling the form controls individually, you can also disable all form controls within a fieldset at once by adding the `disabled` attribute to the `<fieldset>` element.

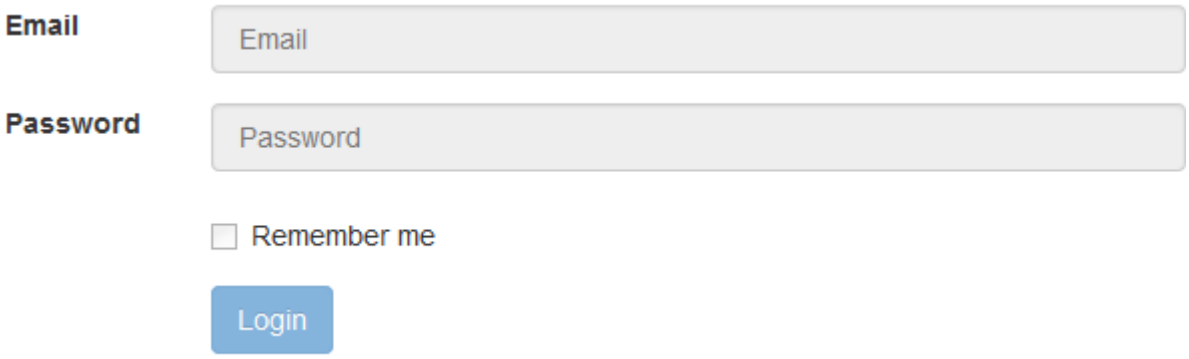
Example

Try this code »

```
<form class="form-horizontal">
    <fieldset disabled="disabled">
```

```
        <div class="form-group">
            <label for="inputEmail" class="control-label col-xs-2">Email</label>
            <div class="col-xs-10">
                <input type="email" class="form-control" id="inputEmail" placeholder="Email">
            </div>
        </div>
        <div class="form-group">
            <label for="inputPassword" class="control-label col-xs-2">Password</label>
            <div class="col-xs-10">
                <input type="password" class="form-control" id="inputPassword" placeholder="Password">
            </div>
        </div>
        <div class="form-group">
            <div class="col-xs-offset-2 col-xs-10">
                <div class="checkbox">
                    <label><input type="checkbox"> Remember me</label>
                </div>
            </div>
        </div>
        <div class="form-group">
            <div class="col-xs-offset-2 col-xs-10">
                <button type="submit" class="btn btn-primary">Login</button>
            </div>
        </div>
    </fieldset>
</form>
```

— The output of the above example will look something like this:



Placing Help Text around Form Controls

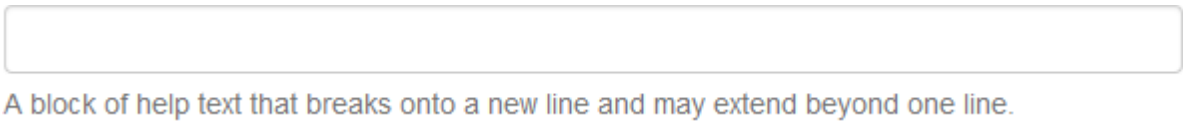
Placing help text for the form controls in an efficient way to guide users to enter the correct data in a form. You can place block level help text for the form controls using the class `.help-block`. The help text is typically displayed at the bottom of the control.

Example

Try this code »

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

— The output of the above example will look something like this:



Bootstrap Form Validation States

Bootstrap provides easy to use and powerful mechanism for styling input controls to present different validation states. Bootstrap includes validation styles for error, warning, and success messages. To use, just add the appropriate class to the surrounding `.form-group`.

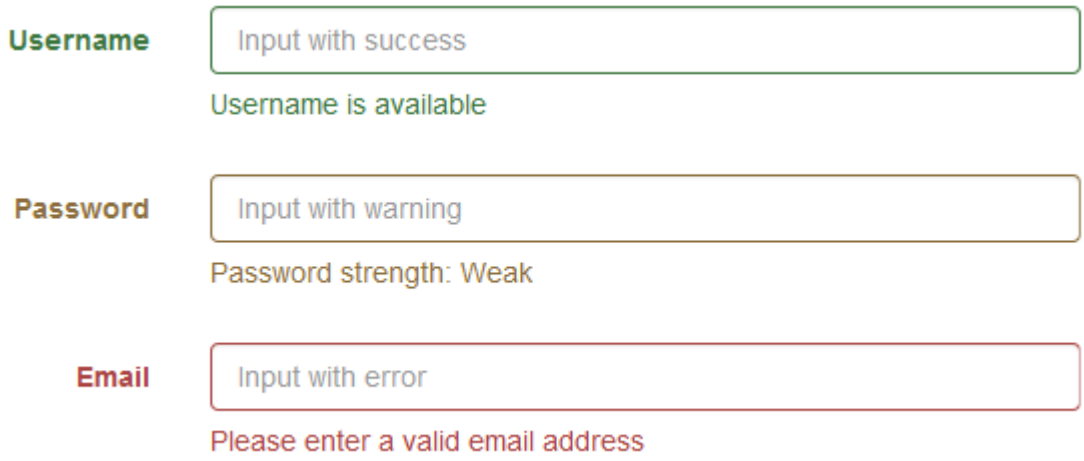
Example

Try this code »

```
<form class="form-horizontal">
    <div class="form-group has-success">
        <label class="col-xs-2 control-label" for="inputSuccess">Username</label>
        <div class="col-xs-10">
            <input type="text" id="inputSuccess" class="form-control" placeholder="Input with success">
            <span class="help-block">Username is available</span>
        </div>
    </div>
    <div class="form-group has-warning">
        <label class="col-xs-2 control-label" for="inputWarning">Password</label>
        <div class="col-xs-10">
            <input type="password" id="inputWarning" class="form-control" placeholder="Input with warning">
            <span class="help-block">Password strength: Weak</span>
        </div>
```

```
    </div>
    <div class="form-group has-error">
        <label class="col-xs-2 control-label" for="inputError">Email</label>
        <div class="col-xs-10">
            <input type="email" id="inputError" class="form-control" placeholder="Input with error">
            <span class="help-block">Please enter a valid email address</span>
        </div>
    </div>
</form>
```

— The output of the above example will look something like this:



You can also add optional feedback icons to your inputs using the class `.has-feedback` on `.form-group` and the right glyphicon, like this:

Example

Try this code »

```
<form class="form-horizontal">
    <div class="form-group has-success has-feedback">
        <label class="col-xs-2 control-label" for="inputSuccess">Username</label>
        <div class="col-xs-10">
            <input type="text" id="inputSuccess" class="form-control" placeholder="Input with success">
            <span class="glyphicon glyphicon-ok form-control-feedback"></span>
            <span class="help-block">Username is available</span>
        </div>
    </div>
    <div class="form-group has-warning has-feedback">
```

```
        <label class="col-xs-2 control-label" for="inputWarning">Password</label>
        <div class="col-xs-10">
            <input type="password" id="inputWarning" class="form-control" placeholder="Input with warning">
            <span class="glyphicon glyphicon-warning-sign form-control-feedback"></span>
            <span class="help-block">Password strength: Weak</span>
        </div>
    </div>
    <div class="form-group has-error has-feedback">
        <label class="col-xs-2 control-label" for="inputError">Email</label>
        <div class="col-xs-10">
            <input type="email" id="inputError" class="form-control" placeholder="Input with error">
            <span class="glyphicon glyphicon-remove form-control-feedback"></span>
            <span class="help-block">Please enter a valid email address</span>
        </div>
    </div>
</form>
```

Supported Form Controls in Bootstrap

Bootstrap includes support for all standard form controls as well as new HTML5 input types such as datetime, number, email, url, search, tel, color etc. The following example will show you the usages of standard form controls with Bootstrap.

Example

Try this code »

```
<form class="form-horizontal">
    <div class="form-group">
        <label class="control-label col-xs-3" for="inputEmail">Email:</label>
        <div class="col-xs-9">
            <input type="email" class="form-control" id="inputEmail" placeholder="Email">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3" for="inputPassword">Password:</label>
        <div class="col-xs-9">
            <input type="password" class="form-control" id="inputPassword" placeholder="Password">
        </div>
```

```
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3" for="confirmPassword">Confirm Password:</label>
        <div class="col-xs-9">
            <input type="password" class="form-control" id="confirmPassword" placeholder="Confirm Password">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3" for="firstName">First Name:</label>
        <div class="col-xs-9">
            <input type="text" class="form-control" id="firstName" placeholder="First Name">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3" for="lastName">Last Name:</label>
        <div class="col-xs-9">
            <input type="text" class="form-control" id="lastName" placeholder="Last Name">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3" for="phoneNumber">Phone:</label>
        <div class="col-xs-9">
            <input type="tel" class="form-control" id="phoneNumber" placeholder="Phone Number">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3">Date of Birth:</label>
        <div class="col-xs-3">
            <select class="form-control">
                <option>Date</option>
            </select>
        </div>
        <div class="col-xs-3">
            <select class="form-control">
                <option>Month</option>
            </select>
        </div>
        <div class="col-xs-3">
            <select class="form-control">
                <option>Year</option>
            </select>
        </div>
```

```
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3" for="postalAddress">Address:</label>
        <div class="col-xs-9">
            <textarea rows="3" class="form-control" id="postalAddress" placeholder="Postal Address"></textarea>
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3" for="ZipCode">Zip Code:</label>
        <div class="col-xs-9">
            <input type="text" class="form-control" id="ZipCode" placeholder="Zip Code">
        </div>
    </div>
    <div class="form-group">
        <label class="control-label col-xs-3">Gender:</label>
        <div class="col-xs-2">
            <label class="radio-inline">
                <input type="radio" name="genderRadios" value="male"> Male
            </label>
        </div>
        <div class="col-xs-2">
            <label class="radio-inline">
                <input type="radio" name="genderRadios" value="female"> Female
            </label>
        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-offset-3 col-xs-9">
            <label class="checkbox-inline">
                <input type="checkbox" value="news"> Send me latest news and updates.
            </label>
        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-offset-3 col-xs-9">
            <label class="checkbox-inline">
                <input type="checkbox" value="agree">  I agree to the <a href="#">Terms and Conditions</a>.
            </label>
        </div>
    </div>
    <br>
    <div class="form-group">
```

```
        <div class="col-xs-offset-3 col-xs-9">
            <input type="submit" class="btn btn-primary"
value="Submit">
            <input type="reset" class="btn btn-default" value="Reset">
        </div>
    </div>
</form>
```

In the next chapter you will learn how to create interactive text-based input controls for your forms using the Bootstrap input group component.

Previous Page Next Page

Bootstrap Input Groups

In this tutorial you will learn how to use the Bootstrap input group component.

Extending Form Controls with Bootstrap

Bootstrap input group component is very flexible and powerful component for creating the interactive form controls, however it is limited to textual input only.

In the following sections you'll see how to extend the text based `<input>` by adding the text or buttons before, after, or on both sides of it to make your form more attractive.

Creating Prepended and Appended Inputs

You can add text and icons or buttons before or after any text-based input.

To prepend or append text and icons to an input:

- Wrap the text or icon within a `<span>` element having the class `.input-group-addon` and place it before or after the `<input>` element.
- Wrap both the `<span>` and text-based `<input>` element within a `<div>` element and apply the class `.input-group` on it.

Note: Bootstrap's prepending or appending feature is only available to text-based inputs. It does not support `<select>` or `<textarea>` elements.

Example
Try this code »

```
<form>
```

```
    <div class="row">
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><span class="glyphicon glyphicon-user"></span></span>
                <input type="text" class="form-control" placeholder="Username">
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group">
                <input type="text" class="form-control" placeholder="Amount">
                <span class="input-group-addon">.00</span>
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon">$</span>
                <input type="text" class="form-control" placeholder="US Dollar">
                <span class="input-group-addon">.00</span>
            </div>
        </div>
    </div>
</form>
```

— The output of the above example will look something like this:

Checkboxes and Radio Buttons Addons

Similarly, you can place checkbox or radio button within input group's addon instead of text.

Example

Try this code »

```
<form>
    <div class="row">
        <div class="col-xs-6">
          <div class="input-group">
            <span class="input-group-addon">
              <input type="checkbox">
            </span>
            <input type="text" class="form-control">
```

```
            </div>
          </div>
          <div class="col-xs-6">
            <div class="input-group">
              <span class="input-group-addon">
                <input type="radio">
              </span>
              <input type="text" class="form-control">
            </div>
          </div>
        </div>
</form>
```

— The output of the above example will look something like this:

Buttons Addons for Text Inputs

You can also prepend or append buttons instead of text. Wrap buttons within the `<span>` element and apply the class `.input-group-btn`, instead of `.input-group-addon`.

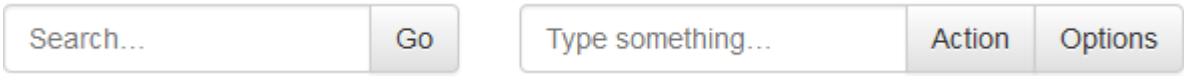
Example

Try this code »

```
<form>
    <div class="row">
        <div class="col-xs-5">
            <div class="input-group">
                <input type="text" class="form-control"
placeholder="Search&hellip;">
                <span class="input-group-btn">
                    <button type="button" class="btn btn-
default">Go</button>
                </span>
            </div>
        </div>
        <div class="col-xs-7">
            <div class="input-group">
                <input type="text" class="form-control"
placeholder="Type something&hellip;">
                <span class="input-group-btn">
                    <button type="button" class="btn btn-
default">Action</button>
                    <button type="button" class="btn btn-
default">Options</button>
```

```
                </span>
            </div>
        </div>
    </div>
</form>
```

— The output of the above example will look something like this:



Adding Button Dropdowns to Text Inputs

You can also create dropdown buttons if you want to do more than one action from a button.

Example

Try this code »

```
<form>
    <div class="row">
        <div class="col-xs-6">
            <div class="input-group">
                <div class="input-group-btn">
                    <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown">Action <span class="caret"></span></button>
                    <ul class="dropdown-menu">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li class="divider"></li>
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </div>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-6">
            <div class="input-group">
                <input type="text" class="form-control">
                <div class="input-group-btn">
                    <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown">Action <span class="caret"></span></button>
                    <ul class="dropdown-menu pull-right">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li class="divider"></li>
```

```
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
</form>
```

— The output of the above example will look something like this:

Adding Segmented Dropdown Button Groups

Similarly, you can define the segmented dropdown button group where dropdown button is placed besides the other buttons.

Example

Try this code »

```
<form>
    <div class="row">
        <div class="col-xs-6">
            <div class="input-group">
                <div class="input-group-btn">
                    <button tabindex="-1" class="btn btn-default" type="button">Action</button>
                    <button tabindex="-1" data-toggle="dropdown" class="btn btn-default dropdown-toggle" type="button">
                        <span class="caret"></span>
                        <span class="sr-only">Toggle Dropdown</span>
                    </button>
                    <ul class="dropdown-menu">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li class="divider"></li>
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </div>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-6">
            <div class="input-group">
                <input type="text" class="form-control">
                <div class="input-group-btn">
```

```
                    <button tabindex="-1" class="btn btn-default"
type="button">Action</button>
                    <button tabindex="-1" data-toggle="dropdown"
class="btn btn-default dropdown-toggle" type="button">
                        <span class="caret"></span>
                        <span class="sr-only">Toggle Dropdown</span>
                    </button>
                    <ul class="dropdown-menu pull-right">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li class="divider"></li>
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </div>
            </div>
        </div>
    </div>
</form>
```

— The output of the above example will look something like this:

Height Sizing of Input Groups

You can also add the relative form sizing classes such as `.input-group-lg` or `.input-group-sm` to the `.input-group` itself to make it larger or smaller.

The contents within `.input-group` will automatically resize — there is no need for repeating the form control size classes on each element.

Example
Try this code »

```
<form>
    <div class="row">
        <div class="col-xs-4">
            <div class="input-group input-group-lg">
                <span class="input-group-addon"><span class="glyphicon
glyphicon-user"></span></span>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group input-group-lg">
```

```
                <span class="input-group-addon">
                    <input type="checkbox">
                </span>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group input-group-lg">
                <div class="input-group-btn">
                    <button tabindex="-1" class="btn btn-default"
type="button">Action</button>
                    <button tabindex="-1" data-toggle="dropdown"
class="btn btn-default dropdown-toggle" type="button">
                        <span class="caret"></span>
                        <span class="sr-only">Toggle Dropdown</span>
                    </button>
                    <ul class="dropdown-menu">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li class="divider"></li>
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </div>
                <input type="text" class="form-control">
            </div>
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon"><span class="glyphicon
glyphicon-user"></span></span>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group">
                <span class="input-group-addon">
                    <input type="checkbox">
                </span>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group">
                <div class="input-group-btn">
                    <button tabindex="-1" class="btn btn-default"
type="button">Action</button>
```

```
                    <button tabindex="-1" data-toggle="dropdown"
class="btn btn-default dropdown-toggle" type="button">
                        <span class="caret"></span>
                        <span class="sr-only">Toggle Dropdown</span>
                    </button>
                    <ul class="dropdown-menu">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li class="divider"></li>
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </div>
                <input type="text" class="form-control">
            </div>
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-4">
            <div class="input-group input-group-sm">
                <span class="input-group-addon"><span class="glyphicon
glyphicon-user"></span></span>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group input-group-sm">
                <span class="input-group-addon">
                    <input type="checkbox">
                </span>
                <input type="text" class="form-control">
            </div>
        </div>
        <div class="col-xs-4">
            <div class="input-group input-group-sm">
                <div class="input-group-btn">
                    <button tabindex="-1" class="btn btn-default"
type="button">Action</button>
                    <button tabindex="-1" data-toggle="dropdown"
class="btn btn-default dropdown-toggle" type="button">
                        <span class="caret"></span>
                        <span class="sr-only">Toggle Dropdown</span>
                    </button>
                    <ul class="dropdown-menu">
                        <li><a href="#">Action</a></li>
                        <li><a href="#">Another action</a></li>
                        <li class="divider"></li>
                        <li><a href="#">Separated link</a></li>
                    </ul>
                </div>
```

```
                    <input type="text" class="form-control">
                </div>
            </div>
        </div>
    </form>
```

Previous Page Next Page

Bootstrap Buttons

In this tutorial you will learn how to create and modify buttons with Bootstrap.

Creating Buttons with Bootstrap

Buttons are the integral part of a website and application. They are used for various purposes like, submit or reset an HTML form, performing interactive actions such as showing or hiding something on a web page on click of the button, etc. The Bootstrap button CSS provides the quick and easy way to create and customize the buttons.

Bootstrap Button Styles

Different classes are available in Bootstrap for styling the buttons as well as to indicate the different states. Button styles can be applied to any element. However, it is applied normally to the `<a>`, `<input>`, and `<button>` elements for the best rendering.

The following table lists the different buttons which are available in the Bootstrap:

Button	Class	Description
Default	`btn btn-default`	Default gray button with gradient.
Primary	`btn btn-primary`	Provides extra visual weight to indicate primary action button in a set of buttons.
Info	`btn btn-info`	Can be used as an alternative to the default button.
Success	`btn btn-success`	Indicates a successful or positive action.
Warning	`btn btn-warning`	Indicates caution should be taken with this action.

Button	Class	Description
Danger	`btn btn-danger`	Indicates a dangerous or potentially negative action.
Link	`btn btn-link`	Deemphasize a button by making it look like a link while maintaining button behavior.

The following example will show you these buttons in action.

Example
Try this code »

```
<button type="button" class="btn btn-default">Default</button>
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-success">Success</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>
```

Warning: Internet Explorer 9 doesn't crop background gradients on rounded corners, so gradient is removed from buttons.

Changing the Sizes of Buttons

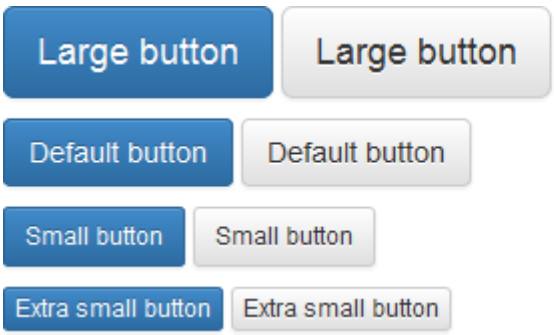
Bootstrap gives you option further to scaling a button up or down. You can make buttons larger or smaller through adding an extra class `.btn-lg`, `.btn-sm`, or `.btn-xs`.

Example
Try this code »

```
<p>
    <button type="button" class="btn btn-primary btn-lg">Large button</button>
    <button type="button" class="btn btn-default btn-lg">Large button</button>
</p>
<p>
    <button type="button" class="btn btn-primary">Default button</button>
    <button type="button" class="btn btn-default">Default button</button>
</p>
```

```
<p>
    <button type="button" class="btn btn-primary btn-sm">Small button</button>
    <button type="button" class="btn btn-default btn-sm">Small button</button>
</p>
<p>
    <button type="button" class="btn btn-primary btn-xs">Extra small button</button>
    <button type="button" class="btn btn-default btn-xs">Extra small button</button>
</p>
```

— The output of the above example will look something like this:



You can also create block level buttons (buttons that covers the full width of the parent elements) by adding an extra class `.btn-block`.

Example

Try this code »

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

— The output of the above example will look something like this:

Bootstrap Disabled Buttons

Sometimes we need to disable a button for certain reasons like, a user in case is not eligible to perform this particular action, or we want to ensure that user should performed all other required actions before proceed to this particular action.

Creating Disabled Buttons Using Anchor Elements

Buttons created through `<a>` tag can be disabled by adding the class `.disabled`.

Example

Try this code »

```
<a href="#" class="btn btn-primary btn-lg disabled">Primary link</a>
<a href="#" class="btn btn-default btn-lg disabled">Link</a>
```

— The output of the above example will look something like this:

Note: The `.disabled` class only changes the visual appearance of the link by making it gray and removing the hover effect, however the link will remain clickable unless you remove the `"href"` attribute. Alternatively, you could implement custom JavaScript to prevent those clicks.

Creating Disabled Buttons Using Button and Input Element

Buttons created through `<button>` or `<input>` tag can be disabled by adding the `"disabled"` attribute to the respective element.

Example

Try this code »

```
<button type="button" class="btn btn-primary btn-lg disabled"
disabled="disabled">Primary button</button>
<button type="button" class="btn btn-default btn-lg"
disabled="disabled">Button</button>
```

— The output of the above example will look something like this:

In the next chapter you will learn how to combine multiple buttons horizontally or vertically in a line like toolbars using the Bootstrap button groups component.

Bootstrap Button Groups

In this tutorial you will learn how to use Bootstrap button group component.

Creating Button Group with Bootstrap

In the previous chapter you've learnt how to create different types of individual buttons and modify them with predefined classes. Bootstrap however, also allows you to group a series of buttons together in a single line through the button group component.

To create a button groups just wrap a series of buttons in a `<div>` element and apply the class `.btn-group` on it, like shown in the example below:

Example

Try this code »

```
<div class="btn-group">
    <button type="button" class="btn btn-primary">Left</button>
    <button type="button" class="btn btn-primary">Middle</button>
    <button type="button" class="btn btn-primary">Right</button>
</div>
```

— The output of the above example will look something like this:

You can also make the button groups appear vertically stacked rather than horizontally. To do this just replace the class `.btn-group` with the `.btn-group-vertical`.

Example

Try this code »

```
<div class="btn-group-vertical">
    <button type="button" class="btn btn-primary">Top</button>
    <button type="button" class="btn btn-primary">Middle</button>
    <button type="button" class="btn btn-primary">Bottom</button>
</div>
```

Creating Button Toolbar

You can also combine sets of button groups together for creating more complex components like button toolbar. To create button toolbar just wrap sets of button groups in a `<div>` element and apply the class `.btn-toolbar` on it.

Example

Try this code »

```
<div class="btn-toolbar">
    <div class="btn-group">
        <button type="button" class="btn btn-primary">1</button>
        <button type="button" class="btn btn-primary">2</button>
        <button type="button" class="btn btn-primary">3</button>
        <button type="button" class="btn btn-primary">4</button>
    </div>
    <div class="btn-group">
        <button type="button" class="btn btn-primary">5</button>
        <button type="button" class="btn btn-primary">6</button>
        <button type="button" class="btn btn-primary">7</button>
    </div>
    <div class="btn-group">
        <button type="button" class="btn btn-primary">8</button>
    </div>
</div>
```

— The output of the above example will look something like this:

Height Sizing of Button Groups

You can also apply relative sizing classes like `.btn-group-lg`, `.btn-group-sm` or `.btn-group-xs` on button groups to create larger or smaller button groups. Just add these button sizing classes directly to the `.btn-group`, instead of applying to every button.

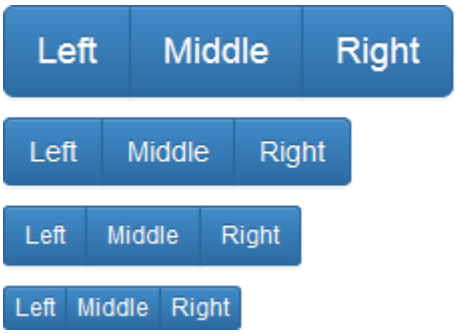
Example

Try this code »

```
<div class="btn-toolbar">
    <div class="btn-group btn-group-lg">
        <button type="button" class="btn btn-primary">Left</button>
        <button type="button" class="btn btn-primary">Middle</button>
        <button type="button" class="btn btn-primary">Right</button>
    </div>
</div>
<div class="btn-toolbar">
```

```
    <div class="btn-group">
        <button type="button" class="btn btn-primary">Left</button>
        <button type="button" class="btn btn-primary">Middle</button>
        <button type="button" class="btn btn-primary">Right</button>
    </div>
</div>
<div class="btn-toolbar">
    <div class="btn-group btn-group-sm">
        <button type="button" class="btn btn-primary">Left</button>
        <button type="button" class="btn btn-primary">Middle</button>
        <button type="button" class="btn btn-primary">Right</button>
    </div>
</div>
<div class="btn-toolbar">
    <div class="btn-group btn-group-xs">
        <button type="button" class="btn btn-primary">Left</button>
        <button type="button" class="btn btn-primary">Middle</button>
        <button type="button" class="btn btn-primary">Right</button>
    </div>
</div>
```

— The output of the above example will look something like this:



Creating Justified Button Groups

You can also make a group of buttons stretch at the same size to span the entire width of its parent by appling an extra class `.btn-group-justified` to the `.btn-group` base class.

The following example will create a justified button group using the `<a>` element.

Example

Try this code »

```
<div class="btn-group btn-group-justified">
    <a href="#" class="btn btn-primary">Left</a>
    <a href="#" class="btn btn-primary">Middle</a>
```

```
    <a href="#" class="btn btn-primary">Right</a>
</div>
```

— The output of the above example will look something like this:

However, to create justified button groups using the `<button>` elements, you must wrap each button individually in a `.btn-group` class, otherwise it will not work.

Example

Try this code »

```
<div class="btn-group btn-group-justified">
    <div class="btn-group">
        <button type="button" class="btn btn-primary">Left</button>
    </div>
    <div class="btn-group">
        <button type="button" class="btn btn-primary">Middle</button>
    </div>
    <div class="btn-group">
        <button type="button" class="btn btn-primary">Right</button>
    </div>
</div>
```

The story doesn't ends here; in the advanced section you'll learn how to create button dropdowns as well as stateful buttons through activating the toggling and loading state on simple buttons and the button groups component.

Previous Page Next Page

Bootstrap Images

In this tutorial you will learn how to style images, creating thumbnails, grids of images and videos, and so on using Bootstrap.

Styling Images with Bootstrap

Images are very common in modern web design. So styling images and placing it properly on the web pages is very important for improving the user experience.

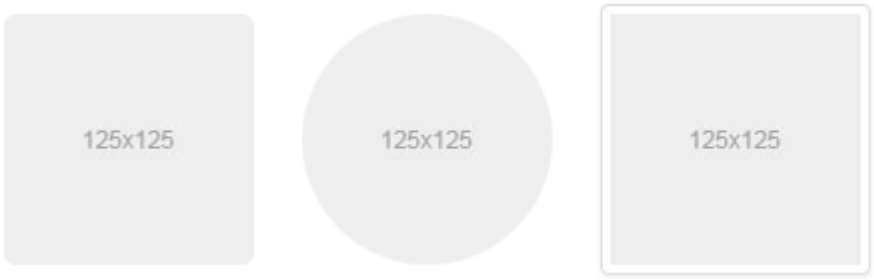
Using the Bootstrap built-in classes you can easily style images such as making the round cornered or circular images, or give them effect like thumbnails.

Example

Try this code »

```
<img src="125x125.jpg" class="img-rounded" alt="Rounded Image">
<img src="125x125.jpg" class="img-circle" alt="Circular Image">
<img src="125x125.jpg" class="img-thumbnail" alt="Thumbnail Image">
```

— The output of the above example will look something like this:



Warning: The classes `.img-rounded` and `.img-circle` do not work in IE8 and lower versions due to lack of support for CSS `border-radius` property.

Creating Responsive Images and Videos

In Bootstrap you can make the images responsive too. Just add the class `.img-responsive` to the `<img>` tag. This class mainly applies the styles `max-width: 100%;` and `height: auto;` to the image so that it scales nicely to fit the containing element — in case if the width of the image is larger than the containing element itself.

Example

Try this code »

```
<img src="kites.jpg" class="img-responsive" alt="Flying Kites">
<img src="sky.jpg" class="img-responsive" alt="Cloudy Sky">
<img src="balloons.jpg" class="img-responsive" alt="Hot Air Balloons">
```

Note: When making the responsive layouts consider making your images responsive too, otherwise if an image width is larger than the parent element's width in any case it will overflow and may break your layout.

You can also make the video or slideshow embedded in a web page responsive without affecting its original aspect ratio. The Bootstrap responsive embed classes can be applied directly to the `<iframe>`, `<embed>`, `<video>`, and `<object>` elements.

Example

Try this code »

```
<!-- 16:9 aspect ratio -->
<div class="embed-responsive embed-responsive-16by9">
    <iframe class="embed-responsive-item"
src="//www.youtube.com/embed/YE7VzlLtp-4"></iframe>
</div>

<!-- 4:3 aspect ratio -->
<div class="embed-responsive embed-responsive-4by3">
    <iframe class="embed-responsive-item"
src="//www.youtube.com/embed/YE7VzlLtp-4"></iframe>
</div>
```

In the above example, we've created the two responsive videos with two different aspect ratios (**16:9** and **4:3**) by adding the classes `.embed-responsive-16by9` and `.embed-responsive-4by3` to their containing blocks respectively and the class `.embed-responsive-item` to the descendant `<iframe>` element.

Tip: The aspect ratio of an image describes the proportional relationship between its width and its height. Two common videographic aspect ratios are 16:9 and 4:3.

Bootstrap Thumbnails

The Bootstrap thumbnail component is very useful for creating grids of images or videos, lists of products, portfolios, and much more. The following example will show you how to create thumbnails to showcase linked images.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-xs-3">
            <a href="#" class="thumbnail">
                <img src="125x125.jpg" alt="125x125">
            </a>
        </div>
        <div class="col-xs-3">
            <a href="#" class="thumbnail">
                <img src="125x125.jpg" alt="125x125">
            </a>
        </div>
```

```
        <div class="col-xs-3">
            <a href="#" class="thumbnail">
                <img src="125x125.jpg" alt="125x125">
            </a>
        </div>
        <div class="col-xs-3">
            <a href="#" class="thumbnail">
                <img src="125x125.jpg" alt="125x125">
            </a>
        </div>
    </div>
</div>
```

— The output of the above example will look something like this:

You can also insert HTML content like headings, paragraphs, or buttons into thumbnails.

Example

Try this code »

```
<div class="container">
    <div class="row">
        <div class="col-xs-6">
            <div class="thumbnail">
                <img src="avatar.jpg" alt="Sample Image">
                <div class="caption">
                    <h3>Thumbnail label</h3>
                    <p>Thumbnail description…</p>
                    <p><a href="#" class="btn btn-primary">Share</a> <a
href="#" class="btn btn-default">Download</a></p>
                </div>
            </div>
        </div>
        <div class="col-xs-6">
            <div class="thumbnail">
                <img src="avatar.jpg" alt="Sample Image">
                <div class="caption">
                    <h3>Thumbnail label</h3>
                    <p>Thumbnail description…</p>
                    <p><a href="#" class="btn btn-primary">Share</a> <a
href="#" class="btn btn-default">Download</a></p>
```

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

Tip: The thumbnails component uses existing grid classes like `.col-xs-*`, `.col-sm-*`, `.col-md-*`, `.col-lg-*`, etc. for controlling the dimensions of thumbnails.

Previous Page Next Page

Bootstrap Media Objects

In this tutorial you will learn how to use the Bootstrap media object component.

Using the Bootstrap Media Object

If you want to create a layout that contains left- or right-aligned media object like images or video alongside the textual content such as blog comments, Tweets, etc. you can do this easily using the newly introduced Bootstrap media component, like this:

Example

Try this code »

```
<div class="media">
    <div class="media-left">
        <a href="#">
            <img src="avatar-tiny.jpg" class="media-object" alt="Sample Image">
        </a>
    </div>
    <div class="media-body">
        <h4 class="media-heading">Jhon Carter <small><i>Posted on January 10, 2014</i></small></h4>
        <p>Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.</p>
    </div>
</div>
```

— The output of the above example will look something like this:

Jhon Carter *Posted on January 10, 2014*

Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.

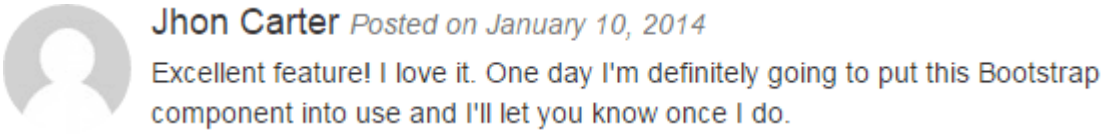
You can also apply image modifier classes like `.img-rounded`, `.img-circle` etc. to the embedded image to create other variation of the media component.

Example

Try this code »

```
<div class="media">
    <div class="media-left">
        <a href="#">
            <img src="avatar-tiny.jpg" class="media-object img-circle" alt="Sample Image">
        </a>
    </div>
    <div class="media-body">
        <h4 class="media-heading">Jhon Carter <small><i>Posted on January 10, 2014</i></small></h4>
        <p>Excellent feature! I love it. One day I'm definitely going to put this Bootstrap component into use and I'll let you know once I do.</p>
    </div>
</div>
```

— The output of the above example will look something like this:



Alignment of Media Objects

You can also align the images or other media at the middle or bottom of the content block using the class `.media-middle` or `.media-bottom`. By default it is top aligned.

Example

Try this code »

```
<!--Top aligned media-->
<div class="media">
    <div class="media-left">
        <a href="#">
            <img src="../images/avatar-tiny.jpg" class="media-object" alt="Sample Image">
        </a>
    </div>
    <div class="media-body">
```

```
        <h4 class="media-heading">Top aligned media <small><i>This is Default</i></small></h4>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit…</p>
    </div>
</div>

<!--Middle aligned media-->
<div class="media">
    <div class="media-left media-middle">
        <a href="#">
            <img src="../images/avatar-tiny.jpg" class="media-object" alt="Sample Image">
        </a>
    </div>
    <div class="media-body">
        <h4 class="media-heading">Middle Aligned Media</h4>
        <p>Vestibulum quis quam ut magna consequat faucibus…</p>
    </div>
</div>

<!--Bottom aligned media-->
<div class="media">
    <div class="media-left media-bottom">
        <a href="#">
            <img src="../images/avatar-tiny.jpg" class="media-object" alt="Sample Image">
        </a>
    </div>
    <div class="media-body">
        <h4 class="media-heading">Bottom Aligned Media</h4>
        <p>Amet nibh libero, in gravida nulla. Nulla vel metus…</p>
    </div>
</div>
```

Creating Nested Media List

You can also create a list of media object or nested media object using the media list component. It can be useful for comment threads or article lists.

Example

Try this code »

```
<ul class="media-list">
    <li class="media">
        <div class="media-left">
            <a href="#">
                <img src="avatar-tiny.jpg" class="media-object" alt="Sample Image">
```

```
            </a>
        </div>
        <div class="media-body">
            <h4 class="media-heading">Media Heading</h4>
            <p>Lorem ipsum dolor sit amet, consectetur adipis elit…</p>
            <!-- Nested media object -->
            <div class="media">
                <div class="media-left">
                    <a href="#">
                        <img src="avatar-tiny.jpg" class="media-object"
alt="Sample Image">
                    </a>
                </div>
                <div class="media-body">
                    <h4 class="media-heading">Nested Media Heading</h4>
                    <p>Vestibulum consectetur scelerisque faucibus…</p>
                    <!-- Nested media object -->
                    <div class="media">
                        <div class="media-left">
                            <a href="#">
                                <img src="avatar-tiny.jpg"
class="media-object" alt="Sample Image">
                            </a>
                        </div>
                        <div class="media-body">
                            <h4 class="media-heading">Nested Media
Heading</h4>
                            <p>Integer pulvinar leo id risus
tempor…</p>
                        </div>
                    </div>
                </div>
            </div>
            <!-- Nested media object -->
            <div class="media">
                <div class="media-left">
                    <a href="#">
                        <img src="avatar-tiny.jpg" class="media-object"
alt="Sample Image">
                    </a>
                </div>
                <div class="media-body">
                    <h4 class="media-heading">Nested Media Heading</h4>
                    <p>Phasellus vitae suscipit justo mauris…</p>
                </div>
            </div>
        </div>
    </li>
    <li class="media">
        <div class="media-left">
```

```
            <a href="#">
                <img src="avatar-tiny.jpg" class="media-object"
alt="Sample Image">
            </a>
        </div>
        <div class="media-body">
            <h4 class="media-heading">Media Heading</h4>
            <p>Quisque pharetra velit id velit iaculis pretium…</p>
        </div>
    </li>
</ul>
```

Previous Page Next Page

Bootstrap Icons

In this tutorial you will learn how to use Bootstrap font icons.

Icons by Glyphicons

Bootstrap 3.3.6 includes more than 250 Glyphicons. As opposed to previous sprite version these icons are available in font format for better usability and scalability.

Since these Glyphicons are font based now you can create icons of any color just applying the CSS `color` property on the specific element.

- glyphicon glyphicon-asterisk
- glyphicon glyphicon-plus
- glyphicon glyphicon-euro
- glyphicon glyphicon-eur
- glyphicon glyphicon-minus
- glyphicon glyphicon-cloud
- glyphicon glyphicon-envelope
- glyphicon glyphicon-pencil
- glyphicon glyphicon-glass
- glyphicon glyphicon-music
- glyphicon glyphicon-search
- glyphicon glyphicon-heart
- glyphicon glyphicon-star
- glyphicon glyphicon-star-empty
- glyphicon glyphicon-user
- glyphicon glyphicon-film
- glyphicon glyphicon-th-large
- glyphicon glyphicon-th
- glyphicon glyphicon-th-list
- glyphicon glyphicon-ok
- glyphicon glyphicon-remove

- glyphicon glyphicon-zoom-in
- glyphicon glyphicon-zoom-out
- glyphicon glyphicon-off
- glyphicon glyphicon-signal
- glyphicon glyphicon-cog
- glyphicon glyphicon-trash
- glyphicon glyphicon-home
- glyphicon glyphicon-file
- glyphicon glyphicon-time
- glyphicon glyphicon-road
- glyphicon glyphicon-download-alt
- glyphicon glyphicon-download
- glyphicon glyphicon-upload
- glyphicon glyphicon-inbox
- glyphicon glyphicon-play-circle
- glyphicon glyphicon-repeat
- glyphicon glyphicon-refresh
- glyphicon glyphicon-list-alt
- glyphicon glyphicon-lock
- glyphicon glyphicon-flag
- glyphicon glyphicon-headphones
- glyphicon glyphicon-volume-off
- glyphicon glyphicon-volume-down
- glyphicon glyphicon-volume-up
- glyphicon glyphicon-qrcode
- glyphicon glyphicon-barcode
- glyphicon glyphicon-tag
- glyphicon glyphicon-tags
- glyphicon glyphicon-book
- glyphicon glyphicon-bookmark
- glyphicon glyphicon-print
- glyphicon glyphicon-camera
- glyphicon glyphicon-font
- glyphicon glyphicon-bold
- glyphicon glyphicon-italic
- glyphicon glyphicon-text-height
- glyphicon glyphicon-text-width
- glyphicon glyphicon-align-left
- glyphicon glyphicon-align-center
- glyphicon glyphicon-align-right
- glyphicon glyphicon-align-justify
- glyphicon glyphicon-list
- glyphicon glyphicon-indent-left
- glyphicon glyphicon-indent-right
- glyphicon glyphicon-facetime-video
- glyphicon glyphicon-picture
- glyphicon glyphicon-map-marker
- glyphicon glyphicon-adjust

- glyphicon glyphicon-tint
- glyphicon glyphicon-edit
- glyphicon glyphicon-share
- glyphicon glyphicon-check
- glyphicon glyphicon-move
- glyphicon glyphicon-step-backward
- glyphicon glyphicon-fast-backward
- glyphicon glyphicon-backward
- glyphicon glyphicon-play
- glyphicon glyphicon-pause
- glyphicon glyphicon-stop
- glyphicon glyphicon-forward
- glyphicon glyphicon-fast-forward
- glyphicon glyphicon-step-forward
- glyphicon glyphicon-eject
- glyphicon glyphicon-chevron-left
- glyphicon glyphicon-chevron-right
- glyphicon glyphicon-plus-sign
- glyphicon glyphicon-minus-sign
- glyphicon glyphicon-remove-sign
- glyphicon glyphicon-ok-sign
- glyphicon glyphicon-question-sign
- glyphicon glyphicon-info-sign
- glyphicon glyphicon-screenshot
- glyphicon glyphicon-remove-circle
- glyphicon glyphicon-ok-circle
- glyphicon glyphicon-ban-circle
- glyphicon glyphicon-arrow-left
- glyphicon glyphicon-arrow-right
- glyphicon glyphicon-arrow-up
- glyphicon glyphicon-arrow-down
- glyphicon glyphicon-share-alt
- glyphicon glyphicon-resize-full
- glyphicon glyphicon-resize-small
- glyphicon glyphicon-exclamation-sign
- glyphicon glyphicon-gift
- glyphicon glyphicon-leaf
- glyphicon glyphicon-fire
- glyphicon glyphicon-eye-open
- glyphicon glyphicon-eye-close
- glyphicon glyphicon-warning-sign
- glyphicon glyphicon-plane
- glyphicon glyphicon-calendar
- glyphicon glyphicon-random
- glyphicon glyphicon-comment
- glyphicon glyphicon-magnet
- glyphicon glyphicon-chevron-up
- glyphicon glyphicon-chevron-down

- glyphicon glyphicon-retweet
- glyphicon glyphicon-shopping-cart
- glyphicon glyphicon-folder-close
- glyphicon glyphicon-folder-open
- glyphicon glyphicon-resize-vertical
- glyphicon glyphicon-resize-horizontal
- glyphicon glyphicon-hdd
- glyphicon glyphicon-bullhorn
- glyphicon glyphicon-bell
- glyphicon glyphicon-certificate
- glyphicon glyphicon-thumbs-up
- glyphicon glyphicon-thumbs-down
- glyphicon glyphicon-hand-right
- glyphicon glyphicon-hand-left
- glyphicon glyphicon-hand-up
- glyphicon glyphicon-hand-down
- glyphicon glyphicon-circle-arrow-right
- glyphicon glyphicon-circle-arrow-left
- glyphicon glyphicon-circle-arrow-up
- glyphicon glyphicon-circle-arrow-down
- glyphicon glyphicon-globe
- glyphicon glyphicon-wrench
- glyphicon glyphicon-tasks
- glyphicon glyphicon-filter
- glyphicon glyphicon-briefcase
- glyphicon glyphicon-fullscreen
- glyphicon glyphicon-dashboard
- glyphicon glyphicon-paperclip
- glyphicon glyphicon-heart-empty
- glyphicon glyphicon-link
- glyphicon glyphicon-phone
- glyphicon glyphicon-pushpin
- glyphicon glyphicon-usd
- glyphicon glyphicon-gbp
- glyphicon glyphicon-sort
- glyphicon glyphicon-sort-by-alphabet
- glyphicon glyphicon-sort-by-alphabet-alt
- glyphicon glyphicon-sort-by-order
- glyphicon glyphicon-sort-by-order-alt
- glyphicon glyphicon-sort-by-attributes
- glyphicon glyphicon-sort-by-attributes-alt
- glyphicon glyphicon-unchecked
- glyphicon glyphicon-expand
- glyphicon glyphicon-collapse-down
- glyphicon glyphicon-collapse-up
- glyphicon glyphicon-log-in
- glyphicon glyphicon-flash
- glyphicon glyphicon-log-out

- glyphicon glyphicon-new-window
- glyphicon glyphicon-record
- glyphicon glyphicon-save
- glyphicon glyphicon-open
- glyphicon glyphicon-saved
- glyphicon glyphicon-import
- glyphicon glyphicon-export
- glyphicon glyphicon-send
- glyphicon glyphicon-floppy-disk
- glyphicon glyphicon-floppy-saved
- glyphicon glyphicon-floppy-remove
- glyphicon glyphicon-floppy-save
- glyphicon glyphicon-floppy-open
- glyphicon glyphicon-credit-card
- glyphicon glyphicon-transfer
- glyphicon glyphicon-cutlery
- glyphicon glyphicon-header
- glyphicon glyphicon-compressed
- glyphicon glyphicon-earphone
- glyphicon glyphicon-phone-alt
- glyphicon glyphicon-tower
- glyphicon glyphicon-stats
- glyphicon glyphicon-sd-video
- glyphicon glyphicon-hd-video
- glyphicon glyphicon-subtitles
- glyphicon glyphicon-sound-stereo
- glyphicon glyphicon-sound-dolby
- glyphicon glyphicon-sound-5-1
- glyphicon glyphicon-sound-6-1
- glyphicon glyphicon-sound-7-1
- glyphicon glyphicon-copyright-mark
- glyphicon glyphicon-registration-mark
- glyphicon glyphicon-cloud-download
- glyphicon glyphicon-cloud-upload
- glyphicon glyphicon-tree-conifer
- glyphicon glyphicon-tree-deciduous
- glyphicon glyphicon-cd
- glyphicon glyphicon-save-file
- glyphicon glyphicon-open-file
- glyphicon glyphicon-level-up
- glyphicon glyphicon-copy
- glyphicon glyphicon-paste
- glyphicon glyphicon-alert
- glyphicon glyphicon-equalizer
- glyphicon glyphicon-king
- glyphicon glyphicon-queen
- glyphicon glyphicon-pawn
- glyphicon glyphicon-bishop

- glyphicon glyphicon-knight
- glyphicon glyphicon-baby-formula
- glyphicon glyphicon-tent
- glyphicon glyphicon-blackboard
- glyphicon glyphicon-bed
- glyphicon glyphicon-apple
- glyphicon glyphicon-erase
- glyphicon glyphicon-hourglass
- glyphicon glyphicon-lamp
- glyphicon glyphicon-duplicate
- glyphicon glyphicon-piggy-bank
- glyphicon glyphicon-scissors
- glyphicon glyphicon-bitcoin
- glyphicon glyphicon-btc
- glyphicon glyphicon-xbt
- glyphicon glyphicon-yen
- glyphicon glyphicon-jpy
- glyphicon glyphicon-ruble
- glyphicon glyphicon-rub
- glyphicon glyphicon-scale
- glyphicon glyphicon-ice-lolly
- glyphicon glyphicon-ice-lolly-tasted
- glyphicon glyphicon-education
- glyphicon glyphicon-option-horizontal
- glyphicon glyphicon-option-vertical
- glyphicon glyphicon-menu-hamburger
- glyphicon glyphicon-modal-window
- glyphicon glyphicon-oil
- glyphicon glyphicon-grain
- glyphicon glyphicon-sunglasses
- glyphicon glyphicon-text-size
- glyphicon glyphicon-text-color
- glyphicon glyphicon-text-background
- glyphicon glyphicon-object-align-top
- glyphicon glyphicon-object-align-bottom
- glyphicon glyphicon-object-align-horizontal
- glyphicon glyphicon-object-align-left
- glyphicon glyphicon-object-align-vertical
- glyphicon glyphicon-object-align-right
- glyphicon glyphicon-triangle-right
- glyphicon glyphicon-triangle-left
- glyphicon glyphicon-triangle-bottom
- glyphicon glyphicon-triangle-top
- glyphicon glyphicon-console
- glyphicon glyphicon-superscript
- glyphicon glyphicon-subscript
- glyphicon glyphicon-menu-left
- glyphicon glyphicon-menu-right

- glyphicon glyphicon-menu-down
- glyphicon glyphicon-menu-up
-
-

Tip: Bootstrap icons are provided by Glyphicons. However you are free to use these icons in your project but as a courtesy you can consider to include a link back to Glyphicons whenever practical.

How to Use Icons in Your Code

To use the Bootstrap icons you will require an `<span>` tag along with a base class `glyphicon` and an individual icon class `glyphicon-*`. The general syntax for using Bootstrap icons is:

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

Where `"glyphicon-class-name"` is the name of the particular icon class (e.g. `glyphicon-search`, `glyphicon-user`, `glyphicon-star` etc.) defined in `bootstrap.css`.

For example, to use search icon you can place the following code just about anywhere:

Example

Try this code »

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

— The output of the above example will look something like this:

Check out the next chapter to see how to use these icons in Bootstrap nav components.

Note: Remember to leave a space after the closing tag of icon element (i.e. after `</span>` tag) when using icons along with the strings of text such as inside buttons or navigation links, to ensure proper spacing.

Bootstrap Nav: Tabs and Pills

In this tutorial you will learn how to create Bootstrap nav components.

Bootstrap Nav Components

Bootstrap provides an easy and quick way to create basic nav components like tabs and pills which are very flexible and elegant. All the Bootstrap's nav components—tabs and pills—share the same base markup and styles through the base `.nav` class.

Creating Basic Tabs with Bootstrap

The following example will show you how to create a basic tab component using Bootstrap.

Example

Try this code »

```
<ul class="nav nav-tabs">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages</a></li>
</ul>
```

— The output of the above example will look something like this:

Adding Icons to Tabs

You can also add icons to your tabs to make it more attractive.

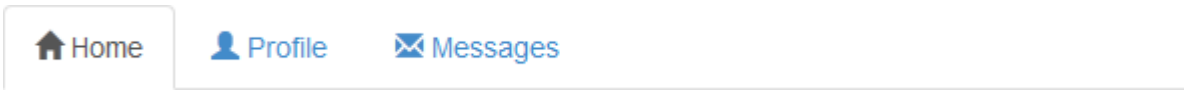
Example

Try this code »

```
<ul class="nav nav-tabs">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span>  Home</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-user"></span> Profile</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-envelope"></span> Messages</a></li>
```

```
</ul>
```

— The output of the above example will look something like this:



Creating Basic Pills Nav with Bootstrap

Similarly you can create a basic pill based navigation using the base class `.nav-pills` instead of `.nav-tabs`, without any further change in markup.

Example

Try this code »

```
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages</a></li>
</ul>
```

— The output of the above example will look something like this:

Adding Icons to Pills Nav

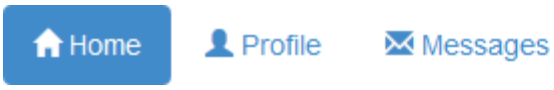
You can also add icons to your pills nav to make it more attractive.

Example

Try this code »

```
<ul class="nav nav-pills">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-
home"></span>  Home</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-user"></span>
Profile</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-envelope"></span>
Messages</a></li>
</ul>
```

— The output of the above example will look something like this:



Stacked Pills Nav

Pills navigations are horizontal by default. To make them appear vertically stacked, just add an extra class `.nav-stacked` to the `<ul>` element.

Example

Try this code »

```
<ul class="nav nav-pills nav-stacked">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages</a></li>
</ul>
```

— The output of the above example will look something like this:

Bootstrap Tabs and Pills Nav with Dropdown Menus

You can add dropdown menus to a link inside tabs and pills navigation with a little extra markup and including the JavaScript files `jquery.js` and `bootstrap.js` in your HTML file.

The four CSS classes `.dropdown`, `.dropdown-toggle`, `.dropdown-menu` and `.caret` are required in addition to the `.nav`, `.nav-tabs`, `.nav-pills` to create a simple dropdown menu.

Creating Tabs with Dropdowns

The following example will show you how to add simple dropdown menus to a tab.

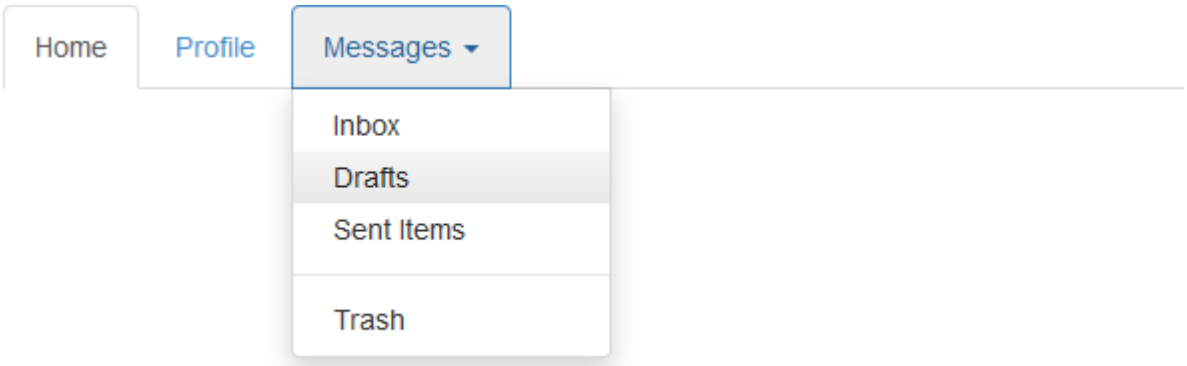
Example

Try this code »

```
<ul class="nav nav-tabs">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
```

```
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-
toggle">Messages <b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#">Inbox</a></li>
            <li><a href="#">Drafts</a></li>
            <li><a href="#">Sent Items</a></li>
            <li class="divider"></li>
            <li><a href="#">Trash</a></li>
        </ul>
    </li>
</ul>
```

— The output of the above example will look something like this:



Creating Pills with Dropdowns

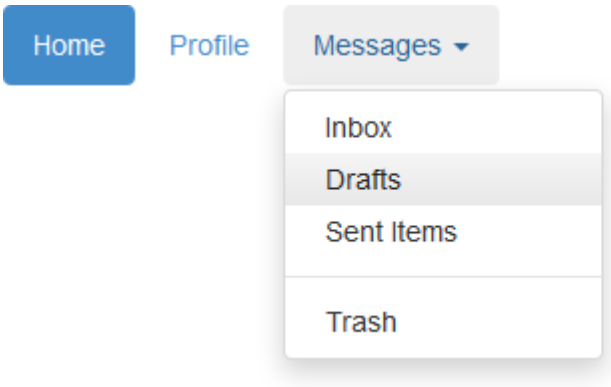
The following example will show you how to add simple dropdown menus to a pill nav.

Example

Try this code »

```
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-
toggle">Messages <b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#">Inbox</a></li>
            <li><a href="#">Drafts</a></li>
            <li><a href="#">Sent Items</a></li>
            <li class="divider"></li>
            <li><a href="#">Trash</a></li>
        </ul>
    </li>
</ul>
```

— The output of the above example will look something like this:



Justified Tabs and Pills

You can easily make your tabs and pills nav having widths equal to their parent element by simply adding an extra class `.nav-justified`. However it works only on screens which are wider than `768px`, on smaller screens, the nav links are stacked.

Example

Try this code »

```
<!-- Justified tabs -->
<ul class="nav nav-tabs nav-justified">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages</a></li>
</ul>
<!-- Justified pills -->
<ul class="nav nav-pills nav-justified">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages</a></li>
</ul>
```

Disable Links inside Nav Components

You can also make any link inside the navs to appear as disabled using the class `.disabled`.

Example

Try this code »

```
<ul class="nav">
    <li class="active"><a href="#">Home</a></li>
```

```
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages</a></li>
    <li class="disabled"><a href="#">Admin</a></li>
</ul>
```

Note: The Bootstrap `.disabled` class only changes the visual appearance of the link by making it gray and removing the hover effect, however the link will remain clickable unless you remove the `"href"` attribute.

Previous Page Next Page

Bootstrap Navbar

In this tutorial you will learn how to create the static and fixed navigation headers using the Bootstrap navbar component.

Creating a Simple Navbar with Bootstrap

You can use the Bootstrap navbar component to create responsive navigation header for your website or application. These responsive navbar initially collapsed on devices having small viewports like cell-phones but expand when user click the toggle button. However, it will be horizontal as normal on the medium and large devices like laptop or desktop.

You can also create different variations of the navbar such as navbars with dropdown menus and search boxes as well as fixed positioned navbar with much less effort. The following example will show you how to create a simple static navbar with navigation links.

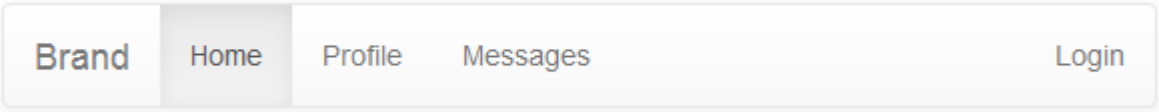
Example

Try this code »

```
<nav role="navigation" class="navbar navbar-default">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-
toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="#" class="navbar-brand">Brand</a>
    </div>
    <!-- Collection of nav links and other content for toggling -->
    <div id="navbarCollapse" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
```

```
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#">Profile</a></li>
            <li><a href="#">Messages</a></li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
            <li><a href="#">Login</a></li>
        </ul>
    </div>
</nav>
```

— The output of the above example will look something like this:



Note: Use the classes `.navbar-left` or `.navbar-right` instead of `.pull-left` or `.pull-right` to align the nav links, forms, buttons or text inside the navbar.

You can also include dropdowns and search form within navbars.

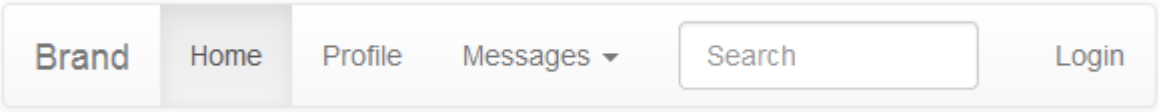
Example

Try this code »

```
<nav role="navigation" class="navbar navbar-default">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-
toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="#" class="navbar-brand">Brand</a>
    </div>
    <!-- Collection of nav links, forms, and other content for toggling
-->
    <div id="navbarCollapse" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#">Profile</a></li>
            <li class="dropdown">
                <a data-toggle="dropdown" class="dropdown-toggle"
href="#">Messages <b class="caret"></b></a>
                <ul role="menu" class="dropdown-menu">
```

```
                    <li><a href="#">Inbox</a></li>
                    <li><a href="#">Drafts</a></li>
                    <li><a href="#">Sent Items</a></li>
                    <li class="divider"></li>
                    <li><a href="#">Trash</a></li>
                </ul>
            </li>
        </ul>
        <form role="search" class="navbar-form navbar-left">
            <div class="form-group">
                <input type="text" placeholder="Search" class="form-
control">
            </div>
        </form>
        <ul class="nav navbar-nav navbar-right">
            <li><a href="#">Login</a></li>
        </ul>
    </div>
</nav>
```

— The output of the above example will look something like this:



Tip: To create navbars that is not fixed on the top or bottom, place it anywhere within a `.container`, which sets the width of your site and content.

Bootstrap Fixed Navbar

Bootstrap also provides mechanism to create navbar that is fixed on the top or bottom of the viewport and will scroll with the content on the page.

Creating Navbar Fixed to Top

Add an extra class `.navbar-fixed-top` in addition to the `.navbar` and `.navbar-default` base class to create navbars that is fixed on the top.

Example

Try this code »

```
<nav role="navigation" class="navbar navbar-default navbar-fixed-top">
```

```
    <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a href="#" class="navbar-brand">Brand</a>
        </div>
        <!-- Collection of nav links and other content for toggling -->
        <div id="navbarCollapse" class="collapse navbar-collapse">
            <ul class="nav navbar-nav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#">Profile</a></li>
                <li><a href="#">Messages</a></li>
            </ul>
            <ul class="nav navbar-nav navbar-right">
                <li><a href="#">Login</a></li>
            </ul>
        </div>
    </div>
</nav>
```

Tip: Place the fixed `.navbar` content inside the `.container` or `.container-fluid` for proper padding and alignment with the rest of the content.

Creating Navbar Fixed to Bottom

Similarly to create navbars that is fixed at the bottom add the class `.navbar-fixed-bottom`.

Example

Try this code »

```
<nav role="navigation" class="navbar navbar-default navbar-fixed-bottom">
    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
```

```
            <a href="#" class="navbar-brand">Brand</a>
        </div>
        <!-- Collection of nav links and other content for toggling -->
        <div id="navbarCollapse" class="collapse navbar-collapse">
            <ul class="nav navbar-nav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#">Profile</a></li>
                <li><a href="#">Messages</a></li>
            </ul>
            <ul class="nav navbar-nav navbar-right">
                <li><a href="#">Login</a></li>
            </ul>
        </div>
    </div>
</nav>
```

Note: Remember to add `padding` (at least `70px`) to the top or bottom of the `<body>` element to avoid the content to go underneath the navbar while implementing fixed top or bottom navbar. Also be sure to add your custom style sheet after the core Bootstrap CSS file, otherwise it may not work.

Bootstrap Static Top Navbar

You can also create full-width navbar that appears on the top but scrolls away with the page by adding the class `.navbar-static-top`. Unlike the `.navbar-fixed-top` class, you do not need to change any padding on the `<body>` element.

Example

Try this code »

```
<nav role="navigation" class="navbar navbar-default navbar-static-top">
    <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" data-target="#navbarCollapse" data-
toggle="collapse" class="navbar-toggle">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a href="#" class="navbar-brand">Brand</a>
        </div>
        <!-- Collection of nav links and other content for toggling -->
        <div id="navbarCollapse" class="collapse navbar-collapse">
```

```
            <ul class="nav navbar-nav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="#">Profile</a></li>
                <li><a href="#">Messages</a></li>
            </ul>
            <ul class="nav navbar-nav navbar-right">
                <li><a href="#">Login</a></li>
            </ul>
        </div>
    </div>
</nav>
```

Bootstrap Navbar with Search Form

Search form is very common component of the navbars and you have seen it on various website quite often. Search form can be placed inside the navbar using the class `.navbar-form` on the `<form>` element.

Example

Try this code »

```
<nav role="navigation" class="navbar navbar-default">
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="#" class="navbar-brand">Brand</a>
    </div>
    <div id="navbarCollapse" class="collapse navbar-collapse">
        <form role="search" class="navbar-form navbar-left">
            <div class="form-group">
                <input type="text" placeholder="Search" class="form-control">
            </div>
            <button type="submit" class="btn btn-default">Submit</button>
        </form>
    </div>
</nav>
```

Creating the Inverted Variation of a Navbar

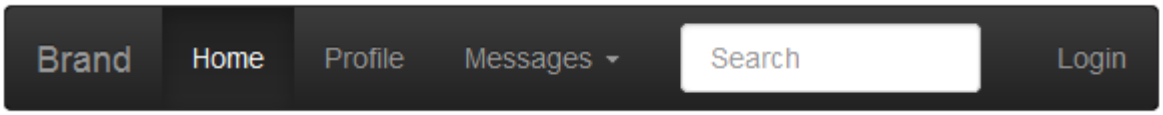
You can also create inverted variation of the Bootstrap navbar by adding an extra class `.navbar-inverse` to the `.navbar` base class, without any further change in markup.

Example

Try this code »

```
<nav role="navigation" class="navbar navbar-inverse">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="#" class="navbar-brand">Brand</a>
    </div>
    <!-- Collection of nav links, forms, and other content for toggling -->
    <div id="navbarCollapse" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#">Profile</a></li>
            <li class="dropdown">
                <a data-toggle="dropdown" class="dropdown-toggle" href="#">Messages <b class="caret"></b></a>
                <ul role="menu" class="dropdown-menu">
                    <li><a href="#">Inbox</a></li>
                    <li><a href="#">Drafts</a></li>
                    <li><a href="#">Sent Items</a></li>
                    <li class="divider"></li>
                    <li><a href="#">Trash</a></li>
                </ul>
            </li>
        </ul>
        <form role="search" class="navbar-form navbar-left">
            <div class="form-group">
                <input type="text" placeholder="Search" class="form-control">
            </div>
        </form>
        <ul class="nav navbar-nav navbar-right">
            <li><a href="#">Login</a></li>
        </ul>
    </div>
</nav>
```

— The output of the above example will look something like this:



Previous Page Next Page

Bootstrap Panels

In this tutorial you will learn how to create panels with Bootstrap.

Introducing Bootstrap 3 Panels

Sometimes you might require to place your content in box for better presentation. In such situation the Bootstrap panel component can be very useful. In most basic form the panel component applies some `border` and `padding` around the content.

Example

Try this code »

```
<div class="panel panel-default">
    <div class="panel-body">Look, I'm in a panel!</div>
</div>
```

— The output of the above example will look something like this:

Look, I'm in a panel!

Panels with Heading

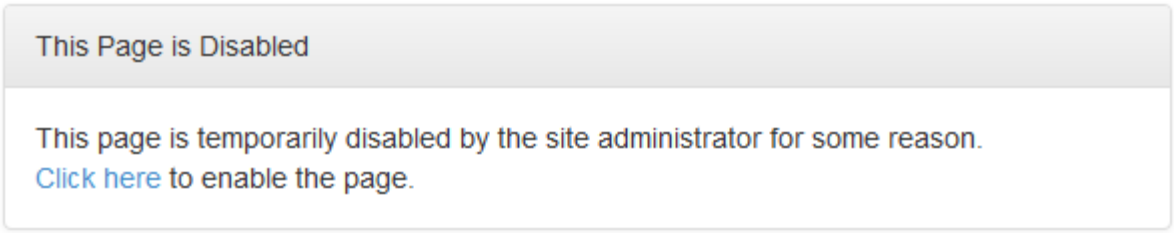
You can also add a heading to your panel with `.panel-heading` class.

Example

Try this code »

```
<div class="panel panel-default">
    <div class="panel-heading">This Page is Disabled</div>
    <div class="panel-body">This page is temporarily disabled by the site administrator for some reason.<br> <a href="#">Click here</a> to enable the page.</div>
</div>
```

— The output of the above example will look something like this:



You can also include heading elements from `<h1>` to `<h6>` with a `.panel-title` class.

Example

Try this code »

```
<div class="panel panel-default">
    <div class="panel-heading">
        <h1 class="panel-title">Panel Title</h1>
    </div>
    <div class="panel-body">Panel content…</div>
</div>
```

Panels with Footer

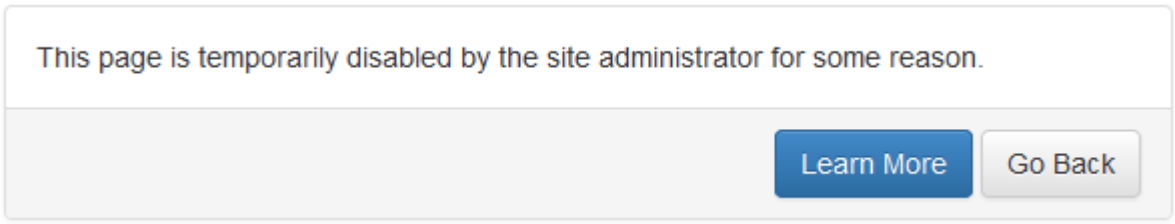
Like heading you can also add footer section to your panels using the `.panel-footer` class. The panel's footer can be used to wrap buttons or secondary text.

Example

Try this code »

```
<div class="panel panel-default">
    <div class="panel-body">This page is temporarily disabled by the
site administrator for some reason.</div>
    <div class="panel-footer clearfix">
        <div class="pull-right">
            <a href="#" class="btn btn-primary">Learn More</a>
            <a href="#" class="btn btn-default">Go Back</a>
        </div>
    </div>
</div>
```

— The output of the above example will look something like this:



Panels with Contextual States

Like other component you also add contextual state classes like `.panel-primary`, `.panel-success`, `.panel-info`, `.panel-warning`, or `.panel-danger` on the panel components to make it more meaningful and drawing attention of the user.

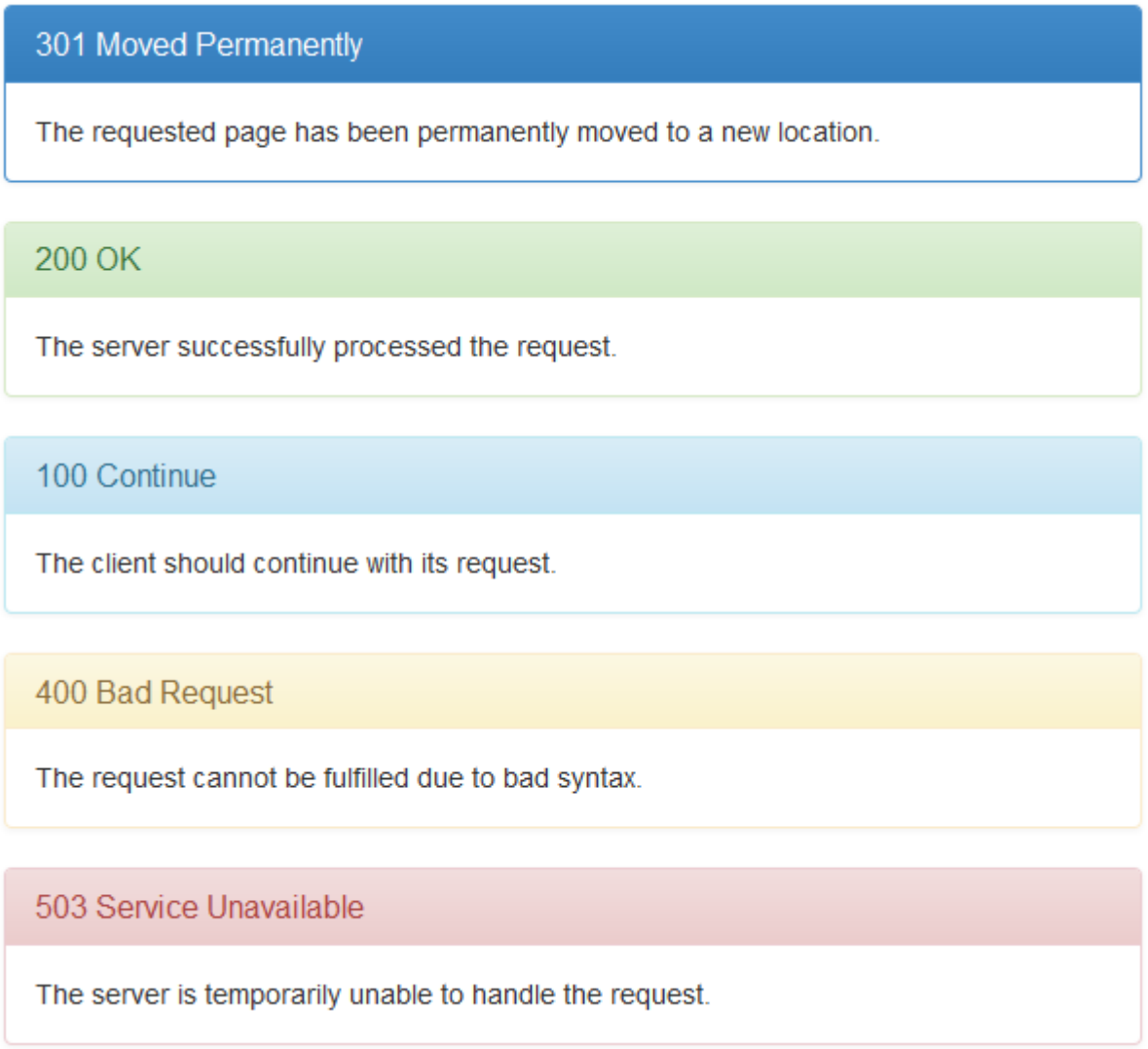
Example

Try this code »

```
<div class="panel panel-primary">
    <div class="panel-heading">
        <h3 class="panel-title">301 Moved Permanently</h3>
    </div>
    <div class="panel-body">The requested page has been permanently moved to a new location.</div>
</div>
<div class="panel panel-success">
    <div class="panel-heading">
        <h3 class="panel-title">200 OK</h3>
    </div>
    <div class="panel-body">The server successfully processed the request.</div>
</div>
<div class="panel panel-info">
    <div class="panel-heading">
        <h3 class="panel-title">100 Continue</h3>
    </div>
    <div class="panel-body">The client should continue with its request.</div>
</div>
<div class="panel panel-warning">
    <div class="panel-heading">
        <h3 class="panel-title">400 Bad Request</h3>
    </div>
    <div class="panel-body">The request cannot be fulfilled due to bad syntax.</div>
</div>
<div class="panel panel-danger">
    <div class="panel-heading">
```

```
        <h3 class="panel-title">503 Service Unavailable</h3>
    </div>
    <div class="panel-body">The server is temporarily unable to handle the request.</div>
</div>
```

— The output of the above example will look something like this:



Placing Tables and List Groups inside Panels

You can add any non-bordered table within a panel to create more informative tables.

Example

Try this code »

```
<div class="panel panel-default">
    <!-- Default panel contents -->
    <div class="panel-heading">User Information</div>
    <div class="panel-body">
        <p>The following table contains some personal information about users.</p>
    </div>
    <!-- Table -->
    <table class="table">
        <thead>
            <tr>
                <th>Row</th>
                <th>First Name</th>
                <th>Last Name</th>
                <th>Email</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>1</td>
                <td>John</td>
                <td>Carter</td>
                <td>johncarter@mail.com</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Peter</td>
                <td>Parker</td>
                <td>peterparker@mail.com</td>
            </tr>
            <tr>
                <td>3</td>
                <td>John</td>
                <td>Rambo</td>
                <td>johnrambo@mail.com</td>
            </tr>
        </tbody>
    </table>
</div>
```

Similarly you can include full-width list groups within any panel.

Example

Try this code »

```
<div class="panel panel-default">
    <!-- Default panel contents -->
    <div class="panel-heading">Products</div>
    <div class="panel-body">
```

```
        <p>The following products are currently available on our store.</p>
    </div>
    <!-- List group -->
    <div class="list-group">
        <a href="#" class="list-group-item">Mobile Phones <span class="badge">50</span>
        </a>
        <a href="#" class="list-group-item">Laptops &amp;amp; Desktops <span class="badge">145</span>
        </a>
        <a href="#" class="list-group-item">Tablets <span class="badge">30</span>
        </a>
        <a href="#" class="list-group-item">Audio &amp;amp; Video Players <span class="badge">65</span>
        </a>
        <a href="#" class="list-group-item">Camera <span class="badge">8</span>
        </a>
    </div>
</div>
```

Previous Page Next Page

Bootstrap Breadcrumbs

In this tutorial you will learn how to create breadcrumbs with Bootstrap.

Creating Breadcrumbs with Bootstrap

A breadcrumb is a navigation scheme that indicates current page's location to the user within a website or application. Breadcrumb navigation can greatly enhance the accessibility of the websites having a large number of pages.

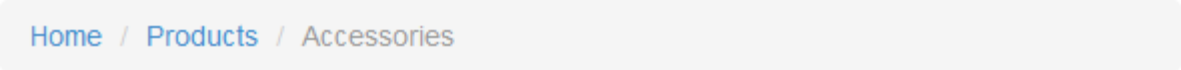
You can create static breadcrumbs layouts with Bootstrap simply using the class `.breadcrumb` on the unordered lists, like this:

Example

Try this code »

```
<ul class="breadcrumb">
    <li><a href="#">Home</a></li>
    <li><a href="#">Products</a></li>
    <li class="active">Accessories</li>
</ul>
```

— The output of the above example will look something like this:



Tip: Separators inside the breadcrumbs components are added automatically via CSS through `::before` pseudo-element and `content` property.

Previous Page Next Page

Bootstrap Pagination

In this tutorial you will learn how to create pagination with Bootstrap.

Creating Pagination with Bootstrap

Pagination is the process of organizing content by dividing it into separate pages.

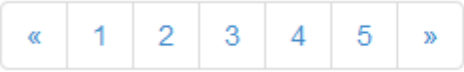
Pagination is used in some or other form quite often in almost every web application, for instance it is used by search engines for displaying a limited number of results on search results pages, or showing a limited number of posts for every page on a blog or forum.

Example

Try this code »

```
<ul class="pagination">
    <li><a href="#">&laquo;</a></li>
    <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>
    <li><a href="#">&raquo;</a></li>
</ul>
```

— The output of the above example will look something like this:



Pagination with Disabled and Active States

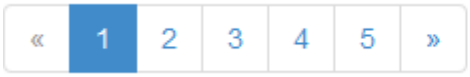
Links inside Bootstrap pagination can further be customized for different circumstances such as when user approaches to an end or start, or indicating current page number to the user. Use the class `.disabled` for making the links unclickable and `.active` to indicate the current page.

Example

Try this code »

```
<ul class="pagination">
    <li class="disabled"><a href="#">&laquo;</a></li>
    <li class="active"><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>
    <li><a href="#">&raquo;</a></li>
</ul>
```

— The output of the above example will look something like this:



Note: The `.disabled` class only displays links as it disabled it doesn't remove the click functionality, to do this you can swap active or disabled anchors with spans.

Example

Try this code »

```
<ul class="pagination">
    <li class="disabled"><span>&laquo;</span></li>
    <li class="active"><span>1</span></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>
    <li><a href="#">&raquo;</a></li>
</ul>
```

Changing the Sizes of Pagination

You can also change the sizes of pagination to increase or decrease the clickable area.

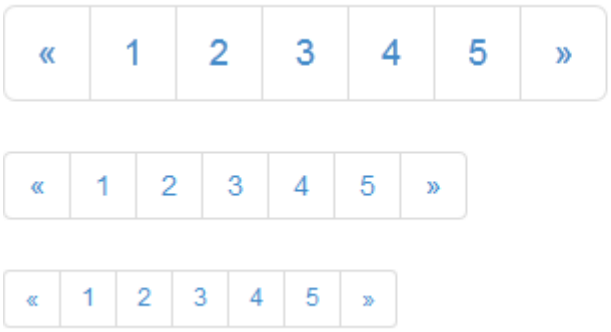
Add the relative sizing classes like `.pagination-lg`, or `.pagination-sm` to the `.pagination` base class for creating larger or smaller pagination.

Example

Try this code »

```
<!-- Larger pagination -->
<ul class="pagination pagination-lg">
    <li><a href="#">&laquo;</a></li>
    <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>
    <li><a href="#">&raquo;</a></li>
</ul>
<!-- Default pagination -->
<ul class="pagination">
    <li><a href="#">&laquo;</a></li>
    <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>
    <li><a href="#">&raquo;</a></li>
</ul>
<!-- Smaller pagination -->
<ul class="pagination pagination-sm">
    <li><a href="#">&laquo;</a></li>
    <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>
    <li><a href="#">&raquo;</a></li>
</ul>
```

— The output of the above example will look something like this:



Bootstrap Pager

Sometimes you may simply require previous and next links on your website to provide simple and quick navigation to the user instead of the complex pagination as we discussed above.

This can be done using the Bootstrap class `.pager`.

Example
Try this code »

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

— The output of the above example will look something like this:

Alignment of Pager

By default pager are aligned horizontally center but you align previous link to left and next link right using the class `.previous` and `.next` respectively.

Example
Try this code »

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

— The output of the above example will look something like this:

You can also apply the same pagination classes `.disabled` and `.active` to the pager.

Example
Try this code »

```
<ul class="pager">
```

```
    <li class="previous disabled"><a href="#">&larr; Previous</a></li>
    <li class="next"><a href="#">Next &rarr;</a></li>
</ul>
```

— The output of the above example will look something like this:

Previous Page Next Page

Bootstrap Labels and Badges

In this tutorial you will learn how to create inline labels and badges with Bootstrap.

Creating Inline Labels

Inline labels are generally used to indicate some valuable information on the web pages such as important notes, warning messages, etc.

The following example will show you how to create inline labels using the Bootstrap.

Example

Try this code »

```
<h1>Bootstrap heading <span class="label label-default">New</span></h1>
<h2>Bootstrap heading <span class="label label-default">New</span></h2>
<h3>Bootstrap heading <span class="label label-default">New</span></h3>
<h4>Bootstrap heading <span class="label label-default">New</span></h4>
<h5>Bootstrap heading <span class="label label-default">New</span></h5>
<h6>Bootstrap heading <span class="label label-default">New</span></h6>
```

— The output of the above example will look something like this:

There are some contextual classes to emphasize these inline labels.

Example

Try this code »

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

— The output of the above example will look something like this:

Creating Inline Badges

Similarly you can create inline badges to provide important notification to the user such as number received or unread messages, number of friend requests etc. They're most commonly found in email client and social networking websites.

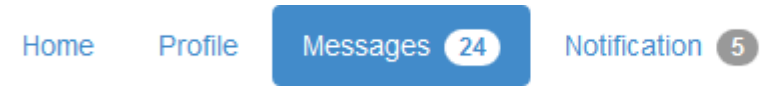
Example

Try this code »

```
<ul class="nav nav-pills">
    <li><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li class="active"><a href="#">Messages <span
class="badge">24</span></a></li>

    <li><a href="#">Notification  <span class="badge">5</span></a></li>
</ul>
```

— The output of the above example will look something like this:



Previous Page Next Page

Bootstrap Progress Bars

In this tutorial you will learn how to create progress bars using Bootstrap.

Creating Progress Bar with Bootstrap

Progress bars can be used for showing the progress of a task or action to the users. The following example will show you how to create a simple progress bar with vertical gradient.

Example

Try this code »

```
<div class="progress">
    <div class="progress-bar" style="width: 60%;">
        <span class="sr-only">60% Complete</span>
    </div>
</div>
```

— The output of the above example will look something like this:

Creating Progress Bar with Label

To show to the progress status as a percentage label just remove the `<span>` with `.sr-only` class from within the progress bar as demonstrated in example above.

Example

Try this code »

```
<div class="progress">
    <div class="progress-bar" style="width: 60%;">
        60%
    </div>
</div>
```

— The output of the above example will look something like this:

If you are showing percentage label you should also add a `min-width` to the progress bar to ensure that the label text remains readable even for low percentage, like this.

Example

Try this code »

```
<div class="progress">
    <div class="progress-bar" style="min-width: 20px;">
        0%
```

```
    </div>
</div>
<div class="progress">
    <div class="progress-bar" style="min-width: 20px; width: 2%;">
        2%
    </div>
</div>
```

Creating Stripped Progress Bar

To create the stripped progress bar just add an extra class `.progress-striped` to the `.progress` base class.

Example

Try this code »

```
<div class="progress progress-striped">
    <div class="progress-bar" style="width: 60%;">
        <span class="sr-only">60% Complete</span>
    </div>
</div>
```

— The output of the above example will look something like this:

Warning: The stripped progress bar uses a gradient to create the striped effect. The stripped progress bar is not supported in IE7-8.

Similarly you can create the animated progress bar — just add the `.active` class to `.progress-stripped`. The `.active` class animates the stripes from right to left.

Example

Try this code »

```
<div class="progress progress-striped active">
    <div class="progress-bar" style="width: 60%;">
        <span class="sr-only">60% Complete</span>
    </div>
</div>
```

Warning: Due to lack of support for CSS3 animation properties the animated progress bar is not supported in any versions of IE upto IE9.

Creating Stacked Progress Bar

You can also place multiple bars into the same progress bar to stack them.

Example

Try this code »

```
<div class="progress">
    <div class="progress-bar progress-bar-success" style="width: 40%">
        <span class="sr-only">Program Files (40%)</span>
    </div>
    <div class="progress-bar progress-bar-warning" style="width: 25%">
        <span class="sr-only">Residual Files (25%)</span>
    </div>
    <div class="progress-bar progress-bar-danger" style="width: 15%">
        <span class="sr-only">Junk Files (15%)</span>
    </div>
</div>
```

— The output of the above example will look something like this:

Progress Bars with Emphasis Classes

Bootstrap also provides some emphasis utility classes for progress bars that can be further used to convey meaning through color.

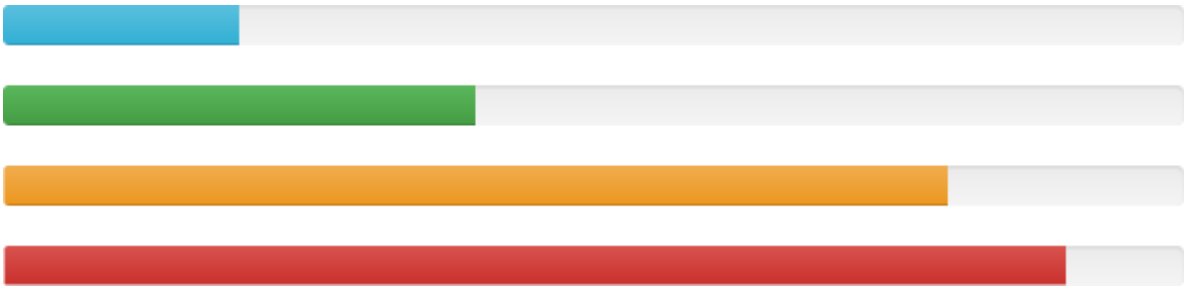
Example

Try this code »

```
<div class="progress">
    <div class="progress-bar progress-bar-info" style="width: 20%">
        <span class="sr-only">20% Used</span>
    </div>
</div>
<div class="progress">
    <div class="progress-bar progress-bar-success" style="width: 40%">
        <span class="sr-only">40% Used</span>
    </div>
</div>
<div class="progress">
    <div class="progress-bar progress-bar-warning" style="width: 80%">
        <span class="sr-only">80% Used</span>
```

```
    </div>
</div>
<div class="progress">
    <div class="progress-bar progress-bar-danger" style="width: 90%">
        <span class="sr-only">90% Used</span>
    </div>
</div>
```

— The output of the above example will look something like this:



Striped Progress Bars with Emphasis Classes

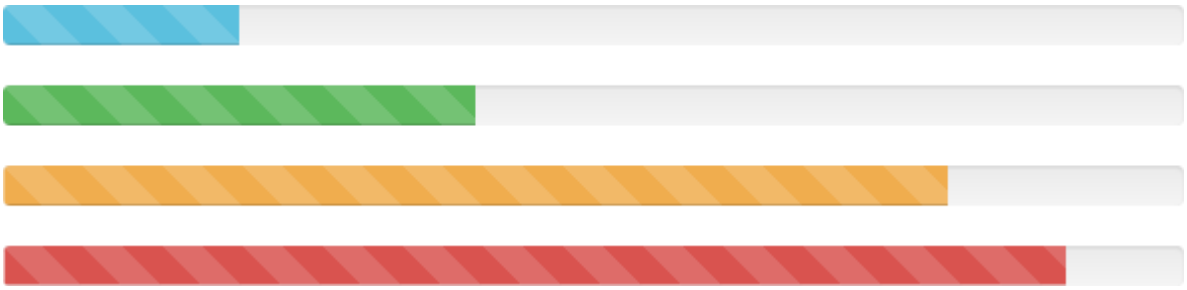
Similar to the solid colors, you can also create varied striped progress bars.

Example

Try this code »

```
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-info" style="width: 20%">
        <span class="sr-only">20% Used</span>
    </div>
</div>
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-success" style="width: 40%">
        <span class="sr-only">40% Used</span>
    </div>
</div>
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-warning" style="width: 80%">
        <span class="sr-only">80% Used</span>
    </div>
</div>
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-danger" style="width: 90%">
        <span class="sr-only">90% Used</span>
    </div>
</div>
```

— The output of the above example will look something like this:



Previous Page Next Page

Bootstrap Jumbotron

In this tutorial you will learn how to use Bootstrap jumbotron component.

Showcasing Contents with Jumbotron

The Bootstrap jumbotron component provides an excellent way to showcase the key content or information on a web page. Just wrap your featured content like heading, descriptions etc. in a `<div>` element and apply the class `.jumbotron` on it.

Example

Try this code »

```
<div class="jumbotron">
    <h1>Learn to Create Websites</h1>
    <p>In today's world internet is the most popular way…</p>
    <p><a href="#" class="btn btn-primary btn-lg">Learn more</a></p>
</div>
```

— The output of the above example will look something like this:

Creating Full Page Width Jumbotron

To create a jumbotron without rounded corners and that covers the full width of the viewport, place it outside all the containers and add the `.container` within like this.

Example

Try this code »

```
<div class="jumbotron">
    <div class="container">
        <h1>Learn to Create Websites</h1>
        <p>In today's world internet is the most popular way…</p>
        <p><a href="#" class="btn btn-primary btn-lg">Learn more</a></p>
    </div>
</div>
```

Previous Page Next Page

Bootstrap Wells

In this tutorial you will learn how to use Bootstrap well component.

Placing Content inside Wells

The Bootstrap well component provides a quick way to apply a simple inset effect to an element. It will be very helpful if you just want to place some content inside a box to make it look like different from rest of the contents. To use this just enclose the contents with a `<div>` element and apply the class `.well` on it, like this:

Example

Try this code »

```
<div class="well">
    Look, I'm in a well!
</div>
```

— The output of the above example will look something like this:

Look, I'm in a well!

Varying Well Sizes

You can further control the padding and rounded corners of the wells using the two optional modifier classes `.well-lg` and `.well-sm`, like this:

Example

Try this code »

```
<div class="well well-lg">
    Look, I'm in a large well!
</div>
<div class="well well-sm">
    Look, I'm in a small well!
</div>
```

Tip: If you're going to place the floated content inside a well, make sure to add the class `.clearfix` to the `.well` element to prevent parent collapsing.

Previous Page Next Page

Bootstrap Helper Classes

In this tutorial you will learn about the Bootstrap helper classes.

Bootstrap Helper Classes

Bootstrap provides a handful of CSS classes for common usages.

Contextual Colors

You can use the contextual color classes like `.text-success`, `.text-info`, `.text-warning` etc. to emphasize the text. See the tutorial on Bootstrap typography for more detail.

Contextual Backgrounds

Similar to the contextual text color classes, you can use the contextual background color classes to set the `background-color` of an element to apply extra emphasis on them.

Example

Try this code »

```
<p class="bg-primary">Important: Please read the instructions carefully before proceeding.</p>
<p class="bg-success">Success: Your message has been sent successfully.</p>
<p class="bg-info">Note: You must agree with the terms and conditions to complete the sign up process.</p>
<p class="bg-warning">Warning: There was a problem with your network connection.</p>
<p class="bg-danger">Error: An error has been occurred while submitting your data.</p>
```

— The output of the above example will look something like this:

Important: Please read the instructions carefully before proceeding.

Success: Your message has been sent successfully.

Note: You must agree with the terms and conditions to complete the sign up process.

Warning: There was a problem with your network connection.

Error: An error has been occurred while submitting your data.

Note: The contextual background classes only apply the CSS `background-color` property on the element. To adjust the space between content and border-box you have to set the CSS `padding` property yourself.

Close Icon

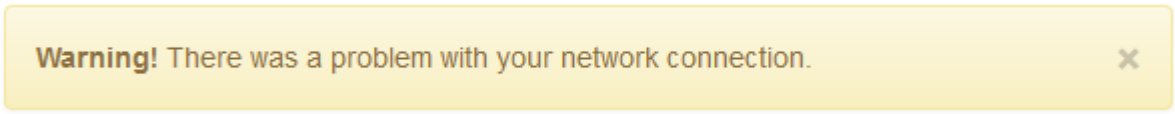
Bootstrap provides a generic close icon that can be used for dismissing modals and alerts.

Example

Try this code »

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

— The output of the above example will look something like this:



Caret Icon

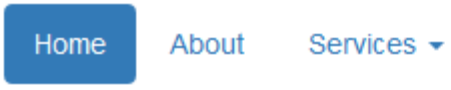
Bootstrap provides a generic caret icon to indicate the dropdown functionality. The direction of the caret icon will reverse automatically in dropup menus.

Example

Try this code »

```
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-toggle">Services <span class="caret"></span></a>
        <ul class="dropdown-menu">
            <li><a href="#">Design</a></li>
            <li><a href="#">Development</a></li>
        </ul>
    </li>
</ul>
```

— The output of the above example will look something like this:



Center Alignment of Content Blocks

You can use the Bootstrap class `.center-block` to align the content block horizontally center. However, to see the effect of this class you have to set the width of the content block yourself and it should be less than the containing parent element.

Example

Try this code »

```
<div class="wrapper">
   <div class="center-block">Center Aligned DIV Box</div>
</div>
```

See the tutorial on CSS alignment to learn more about aligning the elements.

Quick Floats

You can quickly float an element to the left or right using the `.pull-left` and `.pull-right` classes. These classes included the CSS `!important` declaration to avoid specificity issues.

Example

Try this code »

```
<div class="pull-left">Floated to left.</div>
<div class="pull-right">Floated to right.</div>
```

Clearfix

The `.clearfix` class clears the float on any element. This class is widely used for fixing the collapsing parent issue, where parent element contains floated boxes.

Example

Try this code »

```
<div class="wrapper clearfix">
   <div class="pull-left">Float to left</div>
   <div class="pull-right">Float to right</div>
</div>
```

See the tutorial on CSS alignment to learn more about clearing floats on an element.

Showing and Hiding Content

You can force an element to be shown or hidden on all the devices using the `.show` and `.hidden` classes. These classes also included the CSS `!important` declaration to avoid specificity conflicts, just like the quick floats.

Furthermore, you can use the class `.invisible` to toggle only the visibility of an element; however the element still occupies the space in the layout.

Example

Try this code »

```
<div class="show">This is visible to the user.</div>
<div class="hidden">This is not visible to the user.</div>
<div class="invisible">This is not visible but affects the layout.</div>
```

Content Specific to Screen Readers

The special `.sr-only` class hides an element to all devices except screen readers.

Example

Try this code »

```
<p>This paragraph is visible to all devices.</p>
<p class="sr-only">This paragraph is only visible to screen readers.</p>
```

Hide Text Only

You can use the class `.text-hide` to hide the text content of an element.

Example

Try this code »

```
<h1 class="text-hide">The text of this heading is not visible</h1>
<p class="text-hide">The text of this paragraph is not visible.</p>
```

Warning: Think twice before using this class, because it uses the `color: transparent` and `font-size: 0px` to hide the text. Search engines consider such techniques as spam that may result in penalty.

Bootstrap Modals

In this tutorial you will learn how to create modals with Bootstrap.

Creating Modals with Bootstrap

Modals are basically a dialog box that is used to provide important information to the user or prompt user to take necessary actions before moving on. Modal windows are widely used to warn users for situations like session time out or to receive their final confirmation before going to perform any critical actions such as saving or deleting important data.

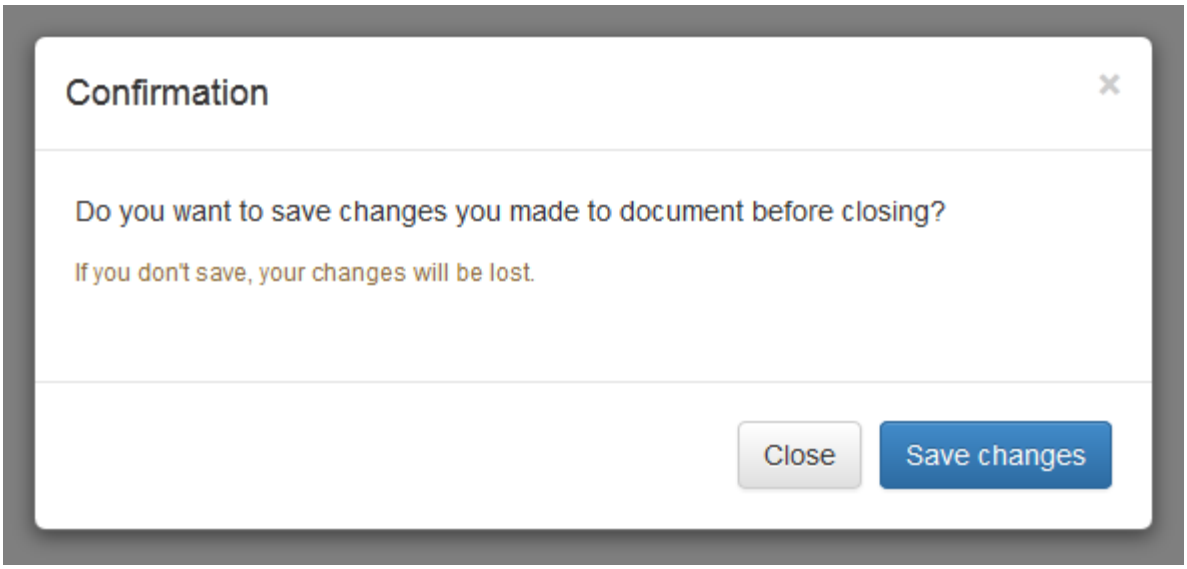
You can easily create very smart and flexible dialog boxes with the Bootstrap modal plugin. The following example will show you how to create a simple modal with a header, message body and the footer containing action buttons for the user.

Example

Try this code »

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

— The above example launches the modal window when the DOM is fully loaded via JavaScript. The output will look something like this:



Activate Modals via Data Attributes

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

Example

Try this code »

```
<!-- Button HTML (to Trigger Modal) -->
<a href="#myModal" role="button" class="btn btn-large btn-primary" data-toggle="modal">Launch Demo Modal</a>

<!-- Modal HTML -->
<div id="myModal" class="modal fade">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
                <h4 class="modal-title">Confirmation</h4>
            </div>
            <div class="modal-body">
                <p>Do you want to save changes you made to document before closing?</p>
                <p class="text-warning"><small>If you don't save, your changes will be lost.</small></p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
```

```
                    <button type="button" class="btn btn-primary">Save changes</button>
                </div>
            </div>
        </div>
    </div>
```

The above example launches the modal window on click of the "Launch Demo Modal" button. Let's go through each part of this modal code one by one for a better understanding.

Explanation of Code

To activate a Bootstrap modal via data attributes we basically need two components — the controller element like a button or link, and the modal element itself.

- The outermost container of every modal in a document must have a unique id (in this case `id="myModal"`), so that it can be targeted via `data-target` (for buttons) or `href` (for hyperlinks) attribute of the controller element.
- The attribute `data-toggle="modal"` is required to add on the controller element, like a button or an anchor, along with a attribute `data-target="#myModal"` or `href="#myModal"` to target a specific modal to toggle.
- The `.modal-dialog` class sets the width as well as horizontal and vertical alignment of the modal box. Whereas the class `.modal-content` sets the styles like text and background color, borders, rounded corners etc.

Rest of the thing is self explanatory, such as the `.modal-header` element defines a header for the modal that usually contains a modal title and a close button, whereas the `.modal-body` element contains the actual content like text, images, forms etc. and the `.modal-footer` element defines the footer that typically contains action buttons for the user.

Note: The `.fade` class on the `.modal` element adds a fading and sliding animation effect while showing and hiding the modal window. If you want the modal that simply appear without any effect you can just remove this class.

Activate Modals via JavaScript

You may also activate a Bootstrap modal window via JavaScript — just call the `modal()` Bootstrap method with the modal `"id"` or `"class"` selector in your JavaScript code.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".btn").click(function(){
        $("#myModal").modal('show');
    });
});
</script>
```

Changing the Sizes of Modals

Bootstrap gives you option further to scaling a modal up or down. You can make modals larger or smaller by adding an extra class `.modal-lg` or `.modal-sm` on `.modal-dialog`.

Example

Try this code »

```
<!-- Large modal -->
<button class="btn btn-primary" data-toggle="modal" data-target="#largeModal">Large modal</button>

<div id="largeModal" class="modal fade bs-example-modal-lg" tabindex="-1" role="dialog">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
                <h4 class="modal-title">Large Modal</h4>
            </div>
            <div class="modal-body">
                <p>Add the <code>.modal-lg</code> class on <code>.modal-dialog</code> to create this large modal.</p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-dismiss="modal">Cancel</button>
                <button type="button" class="btn btn-primary">OK</button>
            </div>
        </div>
    </div>
</div>

<!-- Small modal -->
<button class="btn btn-primary" data-toggle="modal" data-target="#smallModal">Small modal</button>

<div id="smallModal" class="modal fade" tabindex="-1" role="dialog">
```

```
    <div class="modal-dialog modal-sm">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
                <h4 class="modal-title">Small Modal</h4>
            </div>
            <div class="modal-body">
                <p>Add the <code>.modal-sm</code> class on <code>.modal-dialog</code> to create this small modal.</p>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-default" data-dismiss="modal">Cancel</button>
                <button type="button" class="btn btn-primary">OK</button>
            </div>
        </div>
    </div>
</div>
```

Changing Modal Content Based on Trigger Button

Often several modal on a web page has almost same content with minor differences.

You can use the modal events to create slightly different modal windows based on the same modal HTML. The following example will change the title of the modal window according to the trigger button's `data-title` attribute value.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myModal").on('show.bs.modal', function(event){
        // Get button that triggered the modal
        var button = $(event.relatedTarget);
        // Extract value from data-* attributes
        var titleData = button.data('title');
        $(this).find('.modal-title').text(titleData + ' Form');
    });
});
</script>
```

Options

There are certain options which may be passed to `modal()` Bootstrap method to customize the functionality of a modal window.

Name	Type	Default Value	Description
backdrop	boolean or the string `'static'`	true	Includes a modal-backdrop (black overlay area) element. Alternatively, you may specify `static` for a backdrop which doesn't close the modal on click.
keyboard	boolean	true	Closes the modal window on press of escape key.
show	boolean	true	Shows the modal when initialized or activate.
remote	URL	false	Deprecated If a remote url is provided, content will be loaded one time via jQuery's load method and injected into the '`.modal-content`' div.

You may set these options either through the use of data attributes or JavaScript. For setting the modals options via data attributes, just append the option name to `data-`, like `data-backdrop="static"`, `data-keyboard="false"` etc.

However, JavaScript is the more preferable way for setting these options as it prevents you from repetitive work. See the modal's method `.modal(options)` in the section below to know how to set the options for modals using the JavaScript.

If you're using the data api for setting the options for modal window, you may alternatively use the "`href`" attribute to provide the URL of remote source, like this:

Example

Try this code »

```
<!-- Button HTML (to Trigger Modal) -->
<a href="remote.html" role="button" class="btn btn-large btn-primary" data-toggle="modal" data-target="#myModal">Launch Demo Modal</a>

<!-- Modal HTML -->
<div id="myModal" class="modal fade">
    <div class="modal-dialog">
        <div class="modal-content">
            <!-- Content will be loaded here from "remote.php" file -->
        </div>
    </div>
</div>
```

Note: The `remote` option for the Bootstrap modals is deprecated since v3.3.0 and will be removed in v4. Use the client-side templating or a data binding framework instead, or call the `jQuery.load` method yourself.

Methods

These are the standard bootstrap's modals methods:

.modal(options)

This method activates the content as a modal. It also allows you to set options for them.

The jQuery code in the following example will prevent the modal from closing when a user clicks on the backdrop i.e. black overlay area behind the modal.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".launch-modal").click(function(){
        $("#myModal").modal({
            backdrop: 'static'
        });
    });
});
</script>
```

The following jQuery code will prevent the modal from closing on press of the escape key.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".launch-modal").click(function(){
        $("#myModal").modal({
            keyboard: false
        });
    });
});
</script>
```

The jQuery code in the following example will create a modal in which content of the modal will be inserted from a remote file upon activation.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".launch-modal").click(function(){
        $("#myModal").modal({
            remote: '../remote.php'
        });
    });
});
</script>
```

.modal('toggle')

This method toggles a modal window manually.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".toggle-modal").click(function(){
        $("#myModal").modal('toggle');
    });
});
</script>
```

.modal('show')

This method can be used to open a modal window manually.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".open-modal").click(function(){
        $("#myModal").modal('show');
    });
});
</script>
```

.modal('hide')

This method can be used to hide a modal window manually.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".hide-modal").click(function(){
        $("#myModal").modal('hide');
    });
});
</script>
```

.modal('handleUpdate')

This method readjusts the modal's position to counter the jerk that is occurring due to the appearance of the viewport scrollbar in case if the modal height changes in such a way that it becomes higher than the viewport height while it is open.

A common example of this scenario is showing the hidden elements inside the modal via JavaScript or loading content inside the modal using Ajax after activation.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".show-text").click(function(){
        $('#myModal').find(".lots-of-text").toggle();
        $('#myModal').modal('handleUpdate')
    });
});
</script>
```

Events

Bootstrap's modal class includes few events for hooking into modal functionality.

Event	Description
show.bs.modal	This event fires immediately when the show instance method is called.
shown.bs.modal	This event is fired when the modal has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired.
hide.bs.modal	This event is fired immediately when the hide instance method has been called.

Event	Description
hidden.bs.modal	This event is fired when the modal has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired.
loaded.bs.modal	This event is fired when the modal has loaded content using the `remote` option.

The following example displays an alert message to the user when fade out transition of the modal window has been fully completed.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myModal").on('hidden.bs.modal', function(){
        alert("Modal window has been completely closed.");
    });
});
</script>
```

Tip: See also the Bootstrap FAQ section for more examples on modals, like setting vertical alignment, changing default width, embedding video, etc.

Previous Page Next Page

Bootstrap Dropdowns

In this tutorial you will learn how to add dropdown menus to various components using the Bootstrap dropdown plugin.

Creating Dropdown Menus with Bootstrap

The dropdown menu is typically used inside the navigation header to display a list of related links when a user mouse hover or click on the trigger element.

You can use the Bootstrap dropdown plugin to add toggleable dropdown menus (i.e. open and close on click) to almost anything such as links, buttons or button groups, navbar, tabs and pills nav etc. without even writing a single line of JavaScript code.

Adding Dropdowns via Data Attributes

Bootstrap provides an easy and elegant mechanism for adding the dropdown menu to an element via data attributes. The following example will show you the minimum markup required for adding a dropdown menu to the hyperlink via data attributes.

Example

Try this code »

```
<div class="dropdown">
    <a href="#" data-toggle="dropdown" class="dropdown-toggle">Dropdown <b class="caret"></b></a>
    <ul class="dropdown-menu">
        <li><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
    </ul>
</div>
```

The above example demonstrates the most basic form of the Bootstrap dropdowns. Let's understand each part of the Bootstrap dropdown component one by one.

Explanation of Code

The Bootstrap dropdown has basically two components — the dropdown trigger element which can be a hyperlink or button, and the dropdown menu itself.

- The `.dropdown` class specifies a dropdown menu.
- The `.dropdown-toggle` class defines the trigger element, which is a hyperlink in our case, whereas the attribute `data-toggle="dropdown"` is required on the trigger element to toggle the dropdown menu.
- The `.caret` element inside the trigger anchor element creates a small down triangle icon which indicates that the link contains a dropdown menu.
- The unordered list with the class `.dropdown-menu` is actually building the dropdown menu that typically contains the related links or actions.

The previous example code has one small problem. If you click the dropdown link it will add a `"#"` character to the URL while showing the dropdowns. If you want to keep your URLs intact use the `data-target` attribute instead of `href="#"`, like this:

Example

Try this code »

```
<div class="dropdown">
    <a data-target="#" href="page.html" data-toggle="dropdown" class="dropdown-toggle">Dropdown <b class="caret"></b></a>
    <ul class="dropdown-menu">
        <li><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
    </ul>
</div>
```

Similarly, you can add the dropdowns to the buttons and nav components. The following section will show you some common implementation of the Bootstrap dropdown.

Dropdowns within a Navbar

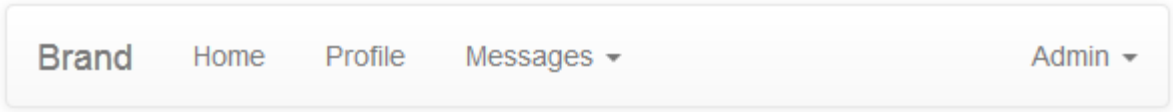
The following examples will show you how to add dropdowns to navbar.

Example

Try this code »

```
<div class="navbar navbar-static">
    <div class="navbar-inner">
        <a href="#" class="brand">Brand</a>
        <ul role="navigation" class="nav">
            <li><a href="#">Home</a></li>
            <li><a href="#">Profile</a></li>
            <li class="dropdown">
                <a href="#" data-toggle="dropdown" class="dropdown-
toggle">Messages <b class="caret"></b></a>
                <ul class="dropdown-menu">
                    <li><a href="#">Inbox</a></li>
                    <li><a href="#">Drafts</a></li>
                    <li><a href="#">Sent Items</a></li>
                    <li class="divider"></li>
                    <li><a href="#">Trash</a></li>
                </ul>
            </li>
        </ul>
        <ul class="nav pull-right">
            <li class="dropdown">
                <a href="#" data-toggle="dropdown" class="dropdown-
toggle">Admin <b class="caret"></b></a>
                <ul class="dropdown-menu">
                    <li><a href="#">Action</a></li>
                    <li><a href="#">Another action</a></li>
                    <li class="divider"></li>
                    <li><a href="#">Settings</a></li>
                </ul>
            </li>
        </ul>
    </div>
</div>
```

— The output of the above example will look something like this:



Tip: You can draw a divider line to separate the links inside a dropdown menu by adding the class `.divider` on a blank list element.

Dropdowns within Navs

The following example will show you how to add dropdowns to pills navs.

Example

Try this code »

```
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-
toggle">Messages <b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#">Inbox</a></li>
            <li><a href="#">Drafts</a></li>
            <li><a href="#">Sent Items</a></li>
            <li class="divider"></li>
            <li><a href="#">Trash</a></li>
        </ul>
    </li>
    <li class="dropdown pull-right">
        <a href="#" data-toggle="dropdown" class="dropdown-
toggle">Admin <b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li class="divider"></li>
            <li><a href="#">Settings</a></li>
        </ul>
    </li>
</ul>
```

— The output of the above example will look something like this:

You can simply convert the above example to a tab dropdown by replacing the class `.nav-pills` with the `.nav-tabs`, without any further change in markup.

Dropdowns within Buttons

The following examples will show you how to add dropdowns to buttons.

Example

Try this code »

```
<div class="btn-group">
    <button type="button" data-toggle="dropdown" class="btn btn-default dropdown-toggle">Action <span class="caret"></span></button>
    <ul class="dropdown-menu">
        <li><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
        <li class="divider"></li>
        <li><a href="#">Separated link</a></li>
    </ul>
</div>
```

— The output of the above example will look something like this:

Bootstrap Split Button Dropdowns

The following examples will show you how to add dropdowns to split buttons.

Example

Try this code »

```
<div class="btn-group">
    <button type="button" class="btn btn-default">Action</button>
    <button type="button" data-toggle="dropdown" class="btn btn-default dropdown-toggle"><span class="caret"></span></button>
    <ul class="dropdown-menu">
        <li><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
        <li class="divider"></li>
        <li><a href="#">Separated link</a></li>
    </ul>
```

```
</div>
```

— The output of the above example will look something like this:

Tip: You can use the Bootstrap's button relative sizing classes like `.btn-lg`, `.btn-sm` and `.btn-xs` to further resizing the buttons dropdowns.

Dropdowns Inside Button Groups

To create dropdown menus inside a button group just place a `.btn-group` along with the dropdown markup within another `.btn-group`.

Example

Try this code »

```
<div class="btn-group">
    <button type="button" class="btn btn-primary">Button</button>
    <button type="button" class="btn btn-primary">Another Button</button>
    <div class="btn-group">
        <button type="button" data-toggle="dropdown" class="btn btn-primary dropdown-toggle">Dropdown <span class="caret"></span></button>
        <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li class="divider"></li>
            <li><a href="#">Separated link</a></li>
        </ul>
    </div>
</div>
```

— The output of the above example will look something like this:

Similarly, you can crate dropdown inside vertically stacked button groups, like this:

Example

Try this code »

```
<div class="btn-group-vertical">
    <button type="button" class="btn btn-primary">Button</button>
    <button type="button" class="btn btn-primary">Another Button</button>
    <div class="btn-group">
        <button type="button" data-toggle="dropdown" class="btn btn-primary dropdown-toggle">Dropdown <span class="caret"></span></button>
        <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li class="divider"></li>
            <li><a href="#">Separated link</a></li>
        </ul>
    </div>
</div>
```

Creating the Dropup Menus

You can even trigger the dropdown menus above the elements by adding an extra class `.dropup` to the parent, as given in the example below.

Example

Try this code »

```
<div class="btn-group dropup">
    <button type="button" class="btn btn-primary">Button</button>
    <button type="button" class="btn btn-primary">Another Button</button>
    <div class="btn-group">
        <button type="button" data-toggle="dropdown" class="btn btn-primary dropdown-toggle">Dropdown <span class="caret"></span></button>
        <ul class="dropdown-menu">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li class="divider"></li>
            <li><a href="#">Separated link</a></li>
        </ul>
    </div>
</div>
```

— The output of the above example will look something like this:

Creating the Right Aligned Dropdown Menus

By default, the top-left corner of the dropdown menu is positioned at the bottom-left corner of its parent element i.e. 100% from the top and along the left side. To right align the dropdown menu just add an extra class `.dropdown-menu-right` to the `.dropdown-menu` base class.

Example

Try this code »

```
<div class="btn-group">
    <button type="button" data-toggle="dropdown" class="btn btn-primary dropdown-toggle">Dropdown <span class="caret"></span></button>
    <ul class="dropdown-menu dropdown-menu-right">
        <li><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
        <li class="divider"></li>
        <li><a href="#">Separated link</a></li>
    </ul>
</div>
```

Adding Headers to Dropdown Items

You can optionally add a menu header to label a section of menu items inside a dropdown menu by adding the class `.dropdown-header` to the list element.

Example

Try this code »

```
<div class="btn-group">
    <button type="button" data-toggle="dropdown" class="btn btn-default dropdown-toggle">Products <span class="caret"></span></button>
    <ul class="dropdown-menu">
        <li class="dropdown-header">ELECTRONICS</li>
        <li><a href="#">Mobiles</a></li>
        <li><a href="#">Tablets</a></li>
        <li><a href="#">Laptops</a></li>
        <li class="dropdown-header">FASHION</li>
        <li><a href="#">Clothing</a></li>
        <li><a href="#">Sunglasses</a></li>
    </ul>
</div>
```

Disable Items within a Dropdown

You can apply the class `.disabled` on a list element to make the menu item look like disabled. However, the link is still clickable, to disable this you can typically remove the anchor's `href` attribute either using the JavaScript or manually.

Example

Try this code »

```
<div class="btn-group">
    <button type="button" data-toggle="dropdown" class="btn btn-default dropdown-toggle">Report <span class="caret"></span></button>
    <ul class="dropdown-menu">
        <li><a href="#">View</a></li>
        <li><a href="#">Download</a></li>
        <li class="disabled"><a href="#">Edit / Delete</a></li>
    </ul>
</div>
```

Adding Dropdowns via JavaScript

You may also add dropdowns manually using the JavaScript — just call the `dropdown()` Bootstrap method with the `"id"` or `"class"` selector of the link or button element in your JavaScript code.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".dropdown-toggle").dropdown();
});
</script>
```

Note: The `data-toggle="dropdown"` is still required for the dropdown's trigger element regardless of whether you call the dropdown via JavaScript or data-api.

Options

None

Methods

This is the standard bootstrap's dropdown method:

$().dropdown('toggle')

A programmatic api for toggling menus for a given navbar or tabbed navigation.

Example

Try this code »

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

Events

These are the standard Bootstrap events to enhance the dropdown functionality. All dropdown events are fired at the `.dropdown-menu`'s parent element.

Event	Description
show.bs.dropdown	This event fires immediately when the show instance method is called. You can use the `event.relatedTarget` to target the toggling anchor element.
shown.bs.dropdown	This event is fired when the dropdown has been made visible to the user. It will wait for CSS transitions, to complete. You can use the `event.relatedTarget` to target the toggling anchor element.
hide.bs.dropdown	This event is fired immediately when the hide instance method has been called. You can use the `event.relatedTarget` to target the toggling anchor element.
hidden.bs.dropdown	This event is fired when the dropdown has finished being hidden from the user. It will wait for CSS transitions, to complete. You can use the `event.relatedTarget` to target the toggling anchor element.

The following example displays the text content of dropdown link when the users click on it.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".dropdown").on("show.bs.dropdown", function(e){
        var linkText = $(e.relatedTarget).text(); // Get the link text
```

```
        alert("Click on OK button to view the dropdown menu for - " +
    linkText);
    });
});
</script>
```

Previous Page Next Page

Bootstrap Tabs

In this tutorial you will learn how to create dynamic tabs to toggle between the content using the Bootstrap tabs plugin.

Creating Tabs with Bootstrap

Tab based navigations provides an easy and powerful mechanism to handle huge amount of content within a small area through separating content into different panes where each pane is viewable one at a time. The user can quickly access the content through switching between the panes without leaving the page. The following example will show you how to create the basic tabs using the Bootstrap tab component.

Example

Try this code »

```
<ul class="nav nav-tabs">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Messages</a></li>
</ul>
```

— The output of the above example will look something like this:

Creating Dynamic Tabs via Data Attributes

You can activate a tab component without writing any JavaScript — simply specify the `data-toggle="tab"` on each tab, as well as create a `.tab-pane` with unique ID for every tab and wrap them in `.tab-content`. The following example will show you how to create a basic tabbable tabs via data attributes in Bootstrap.

Example

Try this code »

```
<ul class="nav nav-tabs">
    <li class="active"><a data-toggle="tab" href="#sectionA">Section A</a></li>
    <li><a data-toggle="tab" href="#sectionB">Section B</a></li>
    <li class="dropdown">
        <a data-toggle="dropdown" class="dropdown-toggle" href="#">Dropdown <b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a data-toggle="tab" href="#dropdown1">Dropdown 1</a></li>
            <li><a data-toggle="tab" href="#dropdown2">Dropdown 2</a></li>
        </ul>
    </li>
</ul>
<div class="tab-content">
    <div id="sectionA" class="tab-pane fade in active">
        <p>Section A content…</p>
    </div>
    <div id="sectionB" class="tab-pane fade">
        <p>Section B content…</p>
    </div>
    <div id="dropdown1" class="tab-pane fade">
        <p>Dropdown 1 content…</p>
    </div>
    <div id="dropdown2" class="tab-pane fade">
        <p>Dropdown 2 content…</p>
    </div>
</div>
```

Creating Dynamic Tabs via JavaScript

You may also enable tabs via JavaScript. Each tab needs to be activated individually.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myTab a").click(function(e){
        e.preventDefault();
        $(this).tab('show');
    });
});
</script>
```

You can activate individual tabs in several ways:

Example

Try this code »

```
$('#myTab a[href="#profile"]').tab('show'); // show tab targeted by the selector
$("#myTab a:first").tab('show'); // show first tab
$("#myTab a:last").tab('show'); // show last tab
$("#myTab li:eq(2) a").tab('show'); // show third tab (0-indexed, like an array)
```

Methods

This is the standard bootstrap's tab method:

$().tab('show')

Activates a tab element and the related content container. Tab should have either a data-target or an href for targeting a container node in the DOM.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myTab li:eq(1) a").tab('show');
})
</script>
```

Events

These are the standard Bootstrap events to enhance the tab functionality.

Event	Description
show.bs.tab	This event fires on tab show, but before the new tab has been shown. You can use the `event.target` and `event.relatedTarget` to target the active tab and the previous active tab (if available) respectively.
shown.bs.tab	This event fires on tab show after a tab has been shown. You can use the `event.target` and `event.relatedTarget` to target the active tab and the previous active tab (if available) respectively.

Event	Description
hide.bs.tab	This event fires when the current active tab is to be hidden and thus a new tab is to be shown. You can use the `event.target` and `event.relatedTarget` to target the current active tab and the new tab which is going to be active very soon, respectively.
hidden.bs.tab	This event fires after the previous active tab is hidden and a new tab is shown. You can use the `event.target` and `event.relatedTarget` to target the previous active tab and the new active tab, respectively.

The following example displays the names of active tab and previous tab to the user when transition of a tab has been fully completed.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('a[data-toggle="tab"]').on('shown', function (e) {
        e.target // active tab
        e.relatedTarget // previous tab
    })
});
</script>
```

Previous Page Next Page

Bootstrap Tooltips

In this tutorial you will learn how to create tooltips with Bootstrap.

Creating Tooltips With Bootstrap

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

Tooltips can be very helpful for the new visitors of your website because they enable the user to know the purpose of icons and links by placing the mouse pointer over them.

Triggering the Tooltips

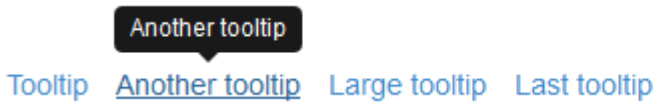
Tooltips can be triggered via JavaScript — just call the `tooltip()` Bootstrap method with the `"id"` or `"class"` selector of the target element in your JavaScript code.

Example

Try this code »

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

— The output of the above example will look something like this:



Note: For performance reasons, the tooltip and popover data-apis are opt in, means to use tooltips and popovers you must initialize them yourself.

Setting the Directions of Tooltips

You can set tooltips to appear on top, right, bottom and left sides of an element.

Positioning of Tooltips via Data Attributes

The following example will show you how to set the direction of tooltips via data attributes.

Example

Try this code »

```
<a href="#" data-toggle="tooltip" data-placement="top" title="Default tooltip">Tooltip</a>
<a href="#" data-toggle="tooltip" data-placement="right" title="Another tooltip">Another tooltip</a>
<a href="#" data-toggle="tooltip" data-placement="bottom" title="A large tooltip.">Large tooltip</a>
<a href="#" data-toggle="tooltip" data-placement="left" title="The last tip!">Last tooltip</a>
```

Positioning of Tooltips via JavaScript

The following example will show you how to set the direction of tooltips via JavaScript.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".tip-top").tooltip({placement : 'top'});
    $(".tip-right").tooltip({placement : 'right'});
    $(".tip-bottom").tooltip({placement : 'bottom'});
    $(".tip-left").tooltip({ placement : 'left'});
});
</script>
```

Options

There are certain options which may be passed to `tooltip()` Bootstrap method to customize the functionality of the tooltip plugin.

Name	Type	Default Value	Description
animation	boolean	true	Apply a CSS fade transition to the tooltip.
html	boolean	false	Insert html into the tooltip. If false, jQuery's `text()` method will be used to insert content into the DOM. Use text if you're worried about XSS attacks.
placement	string | function	'top'	Sets the position of the tooltip — top | bottom | left | right | auto. When "auto" value is specified, it will dynamically reorient the tooltip.

Name	Type	Default Value	Description
			For example, if placement value is "auto top", the tooltip will display on the top when possible, otherwise it will display on the bottom.
selector	string	false	If a selector is provided, tooltip objects will be attached to the specified targets.
title	string | function	''	Sets the default `title` value if title attribute isn't present.
trigger	string	'hover focus'	Specify how tooltip is triggered — click | hover | focus | manual. Note you case pass trigger mutliple, space seperated, trigger types.
delay	number | object	0	Time to delay in showing and hiding the tooltip (ms) — does not apply to manual trigger type. If a number is supplied,

Name	Type	Default Value	Description
			delay is applied to both hide/show Object structure is: `delay: { show: 500, hide: 100 }`
container	string | false	false	Appends the tooltip to a specific element `container: 'body'`
template	string	`'<div class="tooltip" role="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-inner"></div></div>'`	Base HTML to use when creating the tooltip. The tooltip's title will be inserted into the element having the class `.tooltip-inner` and the element with the class `.tooltip-arrow` will become the tooltip's arrow. The outermost wrapper element should have the `.tooltip` class.
viewport	string | object	`{ selector: 'body', padding: 0 }`	Keeps the tooltip within the bounds of this element.

Name	Type	Default Value	Description
			Example: `viewport: '#viewport'` or `{ selector: '#viewport', padding: 0 }`

You may set these options either through the use of data attributes or JavaScript. For setting the tooltips options via data attributes, just append the option name to `data-` along with the correct value, like `data-animation="false"`, `data-placement="bottom"` etc.

However, JavaScript is the more preferable way for setting these options as it prevents you from repetitive work. See the tooltip's method `$().tooltip(options)` in the section below to know how to set the options for tooltips using the JavaScript.

Methods

These are the standard Bootstrap's tooltip methods:

$().tooltip(options)

This method attaches the tooltip handler to a group of elements. It also allows you to set the options for the tooltips, so that you can customize them according to your needs.

The following example will insert the specified text inside the tooltips if the value of the `title` attribute is omitted or missing from the selected elements:

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myTooltips a").tooltip({
        title: 'It works in absence of title attribute.'
    });
});
</script>
```

The following example will show you how to place the HTML content inside a tooltip:

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myTooltip").tooltip({
        title: "<h4><img src='images/smiley.png' alt='Smiley'> Hello, <b>I'm</b> <i>Smiley!</i></h4>",
        html: true
    });
});
</script>
```

The following example will show you how to control the timing of showing and hiding of the tooltip using the tooltip's `delay` option via JavaScript.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    // Showing and hiding tooltip with same speed
    $(".tooltip-tiny").tooltip({
        delay: 100
    });

    // Showing and hiding tooltip with different speed
    $(".tooltip-large").tooltip({
        delay: {show: 0, hide: 500}
    });
});
</script>
```

The following example will show you how you can create your own custom template for the Bootstrap tooltips instead of using the default one.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myTooltips a").tooltip({
        template: '<div class="tooltip"><div class="tooltip-arrow"></div><div class="tooltip-head"><h3><span class="glyphicon glyphicon-info-sign"></span> Tool Info</h3></div><div class="tooltip-inner"></div></div>'
    });
});
</script>
```

The following example will insert the dynamically generated HTML code of the tooltip at the end of a `.wrapper` element instead of the `<body>` element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    // Append tooltip HTML to wrapper element
    $("#myTooltips a").tooltip({container: ".wrapper"});
});
</script>
```

Note: Overriding the tooltip's default `container` option value does not produce any visible difference on the page. To see the actual result you need inspect the resulting DOM using the Firebug or Developer tools.

Similarly, you can set other options for the tooltips using the `$().tooltip(options)` method. Let's check out the other methods of the Bootstrap tooltip plugin.

.tooltip('show')

This method reveals an element's tooltip.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".show-tooltip").click(function(){
        $("#myTooltip").tooltip('show');
    });
});
</script>
```

.tooltip('hide')

This method hides an element's tooltip.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".hide-tooltip").click(function(){
        $("#myTooltip").tooltip('hide');
```

```
    });
});
</script>
```

.tooltip('toggle')

This method toggles an element's tooltip.

Example

Try this code »

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

.tooltip('destroy')

This method hides and destroys an element's tooltip.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".destroy-tooltip").click(function(){
        $("#myTooltip").tooltip('destroy');
    });
});
</script>
```

Events

Bootstrap's tooltip class includes few events for hooking into tooltip functionality.

Event	Description
show.bs.tooltip	This event fires immediately when the show instance method is called.
shown.bs.tooltip	This event is fired when the tooltip has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired.

Event	Description
hide.bs.tooltip	This event is fired immediately when the hide instance method has been called.
hidden.bs.tooltip	This event is fired when the tooltip has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired.
inserted.bs.tooltip	This event is fired after the `show.bs.tooltip` event when the tooltip template has been added to the DOM.

The following example will display an alert message to the user when the fade out transition of the tooltip has been fully completed.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('[data-toggle="tooltip"]').on('hidden.bs.tooltip', function(){
        alert("Tooltip has been completely closed.");
    });
});
</script>
```

Previous Page Next Page

Bootstrap Popovers

In this tutorial you will learn how to create popovers with Bootstrap.

Creating Popovers with Bootstrap

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

Triggering the Popovers

Popovers can be triggered via JavaScript — just call the `popover()` Bootstrap method with the `"id"` or `"class"` selector of the required element in your JavaScript code.

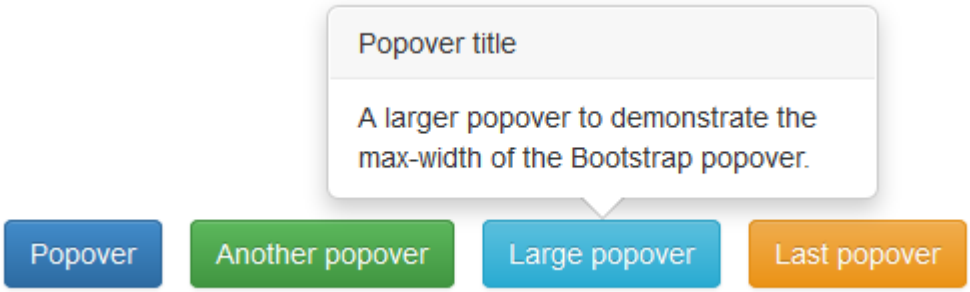
Example

Try this code »

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

- });
- </script>

— The output of the above example will look something like this:



Note: For performance reasons, the tooltip and popover data-apis are opt in, means to use tooltips and popovers you must initialize them yourself.

Setting the Directions of Popovers

You can set popovers to appear on top, right, bottom and left sides of an element.

Positioning of Popovers via Data Attributes

The following example will show you how to set the direction of popovers via data attributes.

Example

Try this code »

- <button type="button" class="btn btn-primary" data-toggle="popover" data-placement="top" title="Popover title" data-content="Default popover">Popover on top</button>
- <button type="button" class="btn btn-success" data-toggle="popover" data-placement="right" title="Popover title" data-content="Popover on right.">Popover on right</button>
- <button type="button" class="btn btn-info" data-toggle="popover" data-placement="bottom" title="Popover title" data-content="Popover on bottom.">Popover on bottom</button>
- <button type="button" class="btn btn-warning" data-toggle="popover" data-placement="left" title="Popover title" data-content="Popover on left.">Popover on left</button>

Positioning of Popovers via JavaScript

The following example will show you how to set the direction of popovers via JavaScript.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".pop-top").popover({placement : 'top'});
    $(".pop-right").popover({placement : 'right'});
    $(".pop-bottom").popover({placement : 'bottom'});
    $(".pop-left").popover({ placement : 'left'});
});
</script>
```

Hiding the Popovers on Next Click

By default popovers are not hiding until you click the trigger element once again. You can use the focus trigger to hide the popovers when the user makes the next click.

Example

Try this code »

```
<a href="#" class="btn btn-primary btn-lg" data-toggle="popover"
tabindex="0" data-trigger="focus" title="Popover title" data-
content="Here's some amazing content.">Dismissible popover</a>
```

Note: To make this feature work properly across the browsers, you must use the `<a>` tag, not the `<button>` tag, and you also must include a `tabindex` attribute.

Options

There are certain options which may be passed to `popover()` Bootstrap method to customize the functionality of the tooltip plugin.

Name	Type	Default Value	Description
animation	boolean	true	Apply a CSS fade transition to the popover.
html	boolean	false	Insert html into the popover. If false, jQuery's `text()`

Name	Type	Default Value	Description
			method will be used to insert content into the DOM. Use text if you're worried about XSS attacks.
placement	string | function	'right'	Sets the position of the popover — top | bottom | left | right | auto. When "auto" value is specified, it will dynamically reorient the popover. For example, if placement value is "auto left", the popover will display to the left when possible, otherwise it will display to right.
selector	string	false	If a selector is provided, popover objects will be attached to the specified targets.
title	string | function	''	Sets the default `title` value if

Name	Type	Default Value	Description
			title attribute isn't present.
trigger	string	'click'	Specify how popover is triggered — click | hover | focus | manual.
content	string | function	''	Sets the default content value if `'data-content'` attribute isn't present.
delay	number | object	0	Time to delay in showing and hiding the popover (ms) — does not apply to manual trigger type. If a number is supplied, delay is applied to both hide/show Object structure is: `delay: { show: 500, hide: 100 }`
container	string | false	false	Appends the popover to a specific element `container: 'body'`
template	string	`'<div class="popover"> <div class="arrow"></div><h3 class="popover-title"></h3><div class="popover-content"></div></div>'`	Base HTML to use when creating the

Name	Type	Default Value	Description
			popover. The popover's title and content will be inserted into the elements having the class `.popover-title` and `.popover-content` respectively. Whereas the element with the class `.arrow` will become the popover's arrow. The outermost wrapper element should have the `.popover` class.
viewport	string | object	`{ selector: 'body', padding: 0 }`	Keeps the popover within the bounds of this element. Example: `viewport: '#viewport'` or `{ selector: '#viewport', padding: 0 }`

You may set these options either through the use of data attributes or JavaScript. For setting the popovers options via data attributes, just append the option name to `data-` along with the correct value, like `data-animation="false"`, `data-placement="bottom"` etc.

However, JavaScript is the more preferable way for setting these options as it saves you from doing the repetitive work. See the popover's method `$().popover(options)` in the section below to know how to set the options for popovers using the JavaScript.

Methods

These are the standard Bootstrap's popover methods:

$().popover(options)

This method attaches the popover handler to a group of elements. It also allows you to set the options for the popovers so that you can customize them according to your needs.

The following example will insert the specified text inside the popovers title if the value of the `title` attribute is omitted or missing from the selected elements:

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myPopovers a").popover({
        title: 'Default title value'
    });
});
</script>
```

The following jQuery code will trigger the popovers on mouse hover instead of click:

Example
Try this code »

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

The following example will show you how to place the HTML content inside a popover:

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myPopover").popover({
        title: '<h3 class="custom-title"><span class="glyphicon glyphicon-info-sign"></span> Popover Info</h3>',
        content: '<p>This is a <em>simple example</em> demonstrating how to insert HTML code inside <mark><strong>Bootstrap popover</strong></mark>.</p>',
        html: true
    });
});
</script>
```

The following example will show you how to control the timing of showing and hiding of the popover using the popover's `delay` option via JavaScript.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    // Showing and hiding popover with same speed
    $(".popover-tiny").tooltip({
        delay: 500
    });

    // Showing and hiding popover with different speed
    $(".popover-large").tooltip({
        delay: {show: 0, hide: 2000}
    });
});
</script>
```

The following example will show you how you can create your own custom template for the Bootstrap popovers instead of using the default one.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('[data-toggle="popover"]').popover({
        html: true,
        template: '<div class="popover"><div class="arrow"></div><h3 class="popover-title"></h3><div class="popover-content"></div><div class="popover-footer"><a href="#" class="btn btn-info btn-sm">Close</a></div></div>'
    });

    // Custom jQuery to hide popover on click of the close button
```

```
    $(document).on("click", ".popover-footer .btn" , function(){
        $(this).parents(".popover").popover('hide');
    });
});
</script>
```

The following example will insert the dynamically generated HTML code of the popover at the end of a `.wrapper` element instead of the `<body>` element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    // Append popover HTML to wrapper element
    $("#myPopovers a").popover({container: '.wrapper'});
});
</script>
```

Note: Overriding the popover's default `container` option value does not produce any visible difference on the page. To see the actual result you need inspect the resulting DOM using the Firebug or Developer tools.

Similarly, you can set other options for the popovers using the `$().popover(options)` method. Let's check out the other methods of the Bootstrap popover plugin.

.popover('show')

This method reveals an element's popover.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".show-popover").click(function(){
        $("#myPopover").popover('show');
    });
});
</script>
```

.popover('hide')

This method hides an element's popover.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".hide-popover").click(function(){
        $("#myPopover").popover('hide');
    });
});
</script>
```

.popover('toggle')

This method toggles an element's popover.

Example

Try this code »

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

.popover('destroy')

This method hides and destroys an element's popover.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".destroy-popover").click(function(){
        $("#myPopover").popover('destroy');
    });
});
</script>
```

Events

Bootstrap's popover class includes few events for hooking into popover functionality.

Event	Description
show.bs.popover	This event fires immediately when the show instance method is called.
shown.bs.popover	This event is fired when the popover has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired.
hide.bs.popover	This event is fired immediately when the hide instance method has been called.
hidden.bs.popover	This event is fired when the popover has finished being hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired.
inserted.bs.popover	This event is fired after the `show.bs.popover` event when the popover template has been added to the DOM.

The following example displays an alert message to the user when fade out transition of the popover has been fully completed.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('[data-toggle="popover"]').on('hidden.bs.popover', function(){
        alert("Popover has been completely closed.");
    });
});
</script>
```

Previous Page Next Page

Bootstrap Alerts

In this tutorial you will learn how to create alerts messages with Bootstrap.

Creating Alert Messages with Bootstrap

Alert boxes are used quite often to stand out the information that requires immediate attention of the end users such as warning, error or confirmation messages.

With Bootstrap you can easily create elegant alert messages box for various purposes. You can also add an optional close button to dismiss any alert.

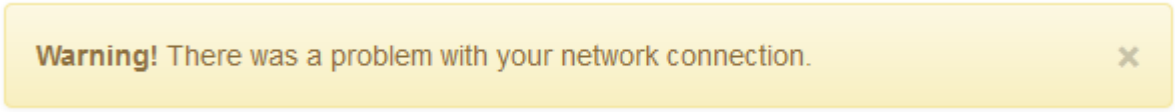
Warning Alerts

You can create a simple Bootstrap warning alert message box by adding the contextual class `.alert-warning` to the `.alert` base class, like this:

Example
Try this code »

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

— The output of the above example will look something like this:



Tip: If you want to enable the fading transition effect while closing the alert boxes, apply the classes `.fade` and `.in` to them along with the contextual class.

Example
Try this code »

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

— Similarly you can create other alert messages.

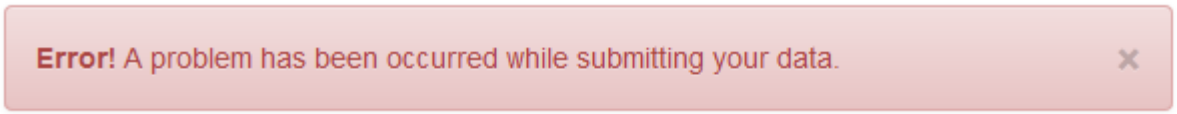
Error or Danger Alerts

Add the class `.alert-danger` to the `.alert` base class to create error or danger alerts.

Example
Try this code »

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

— The output of the above example will look something like this:



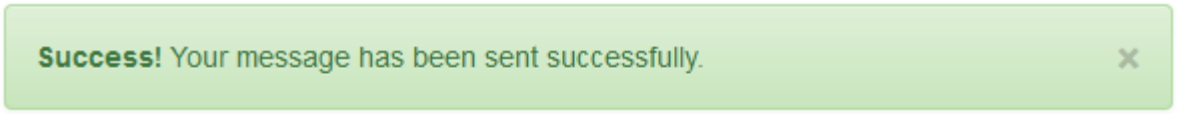
Success or Confirmation Alerts

Likewise, to create the success or confirmation alert message box add the contextual class `.alert-success` to the `.alert` base class.

Example
Try this code »

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

— The output of the above example will look something like this:



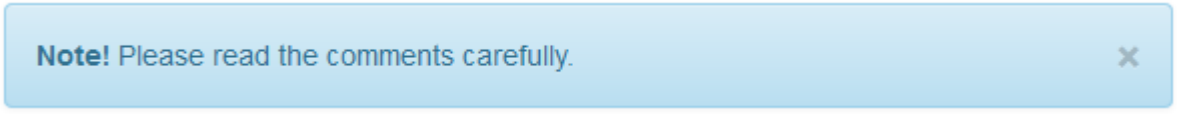
Information Alerts

For information alert messages add the class `.alert-info` to the `.alert` base class.

Example
Try this code »

```
<div class="alert alert-info fade in">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Note!</strong> Please read the comments carefully.
</div>
```

— The output of the above example will look something like this:



Dismiss Alerts via Data Attribute

Data attributes provides an easy way to add close functionality to the alert messages box. Just add the `data-dismiss="alert"` to the close button and it will automatically enable the dismissal of the containing alert message box.

Example

Try this code »

```
<a href="#" class="close" data-dismiss="alert">&times;</a><br>
<button type="button" class="close" data-dismiss="alert">&times;</button>
<span class="close" data-dismiss="alert">&times;</span>
```

Dismiss Alerts via JavaScript

You may also enable the dismissal of an alert via JavaScript.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".close").click(function(){
        $("#myAlert").alert();
    });
});
</script>
```

Methods

These are the standard bootstrap's alerts methods:

$().alert()

This method wraps all alerts with close functionality.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".close").click(function(){
        $(".alert").alert();
```

```
    });
});
</script>
```

$().alert('close')

This method closes an alert message box.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".close").click(function(){
        $("#myAlert").alert('close');
    });
});
</script>
```

Events

Bootstrap's alert class includes few events for hooking into alert functionality.

Event	Description
close.bs.alert	This event fires immediately when the close instance method is called.
closed.bs.alert	This event is fired when the alert message box has been closed. It will wait until the CSS transition process has been fully completed before getting fired.

The following example displays an alert message to the user when fade out transition of an alert message box has been fully completed.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myAlert").on('closed.bs.alert', function () {
        alert("Alert message box has been closed.");
    });
});
</script>
```

Previous Page Next Page

Bootstrap Stateful Buttons

In this tutorial you will learn how to create toggle and loading Bootstrap buttons.

Controlling Button States

In the previous section you've learnt about the Bootstrap button styling and the modifications as well as how to create button groups and toolbars. However, with Bootstrap you can do more with buttons like controlling the states of buttons, make checkbox and radio inputs behaves like toggle buttons, etc. Let's discuss about them in detail.

Creating Single Toggle Button

You can activate toggling (i.e. change the normal state of a button to a push state and vice versa) on a single button by simply adding the data attribute `data-toggle="button"`.

Example

Try this code »

```
<button type="button" class="btn btn-primary" data-toggle="button">Single Toggle Button</button>
```

— The output of the above example will look something like this:

Add Loading State on Buttons

You can change the normal state of a button to a loading state by simply adding the data attribute `data-loading-text="Loading..."` to a button.

Example

Try this code »

```
<button type="button" class="btn btn-primary" data-loading-text="Loading...">Loading state</button>
```

— The output of the above example will look something like this:

Note: Mozilla firefox persists the disabled state across page loads, to prevent this behavior, you may simply set `autocomplete="off"` on the form containing the buttons, or directly to the input or button element.

Creating Buttons Checkbox

You can add the attribute `data-toggle="buttons"` to a group of checkboxes for checkbox style toggling on button groups, like this:

Example

Try this code »

```
<div class="btn-group" data-toggle="buttons">
    <label class="btn btn-primary">
        <input type="checkbox" name="options"> Option 1
    </label>
    <label class="btn btn-primary">
        <input type="checkbox" name="options"> Option 2
    </label>
    <label class="btn btn-primary">
        <input type="checkbox" name="options"> Option 3
    </label>
</div>
```

— The output of the above example will look something like this:

Add the class `.active` on input's label element if you want options pre-checked by default.

Example

Try this code »

```
<div class="btn-group" data-toggle="buttons">
    <label class="btn btn-primary active">
        <input type="checkbox" name="options"> Option 1
    </label>
    <label class="btn btn-primary">
        <input type="checkbox" name="options"> Option 2
    </label>
    <label class="btn btn-primary active">
        <input type="checkbox" name="options"> Option 3
```

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

Creating Buttons Radio

Similarly you can add the attribute `data-toggle="buttons"` to a group of radio inputs for radio style toggling on button groups, like this:

Example

Try this code »

```
<div class="btn-group" data-toggle="buttons">
    <label class="btn btn-primary">
        <input type="radio" name="options"> Option 1
    </label>
    <label class="btn btn-primary">
        <input type="radio" name="options"> Option 2
    </label>
    <label class="btn btn-primary">
        <input type="radio" name="options"> Option 3
    </label>
</div>
```

— The output of the above example will look something like this:

Add the class `.active` on input's label element if you want an option pre-selected by default.

Example

Try this code »

```
<div class="btn-group" data-toggle="buttons">
    <label class="btn btn-primary active">
        <input type="radio" name="options"> Option 1
    </label>
    <label class="btn btn-primary">
        <input type="radio" name="options"> Option 2
    </label>
    <label class="btn btn-primary">
        <input type="radio" name="options"> Option 3
    </label>
</div>
```

Enable Buttons via JavaScript

You may also enable buttons via JavaScript:

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".nav-tabs a").click(function(){
        $(this).button('loading').delay(500).queue(function(){
            $(this).button('reset');
            $(this).dequeue();
        });
    });
});
</script>
```

Options

None

Methods

These are the standard bootstrap's buttons methods:

$().button('toggle')

This method toggles push state of the button. It changes the appearance of the button, and makes it look like that it has been activated. You can also enable auto toggling of a button by using the `"data-toggle"` attribute.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myButton").click(function(){
        $(this).button('toggle');
    });
});
</script>
```

$().button('loading')

This method sets the button state to loading. When loading, the button is disabled and the text is swapped with the value of `"data-loading-text"` attribute of button element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myButton").click(function(){
        $(this).button('loading');
    });
});
</script>
```

Note: The use of `data-loading-text` and `$().button('loading')` has been deprecated since Bootstrap v3.3.6 and will be removed from v4. You should better use `$().button(string)` method to ensure future compatibility.

$().button('reset')

This method resets button state by swapping text to original text.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myButton").click(function(){
        $(this).button('loading').delay(1000).queue(function(){
            $(this).button('reset');
            $(this).dequeue();
        });
    });
});
</script>
```

$().button(string)

This method resets button state by swapping text to any data defined text state.

Example

Try this code »

```
<script type="text/javascript">
```

```
$(document).ready(function(){
    $("#myButton").click(function(){
        $(this).button('loading').delay(1000).queue(function() {
            $(this).button('complete');
            $(this).dequeue();
        });
    });
});
</script>
```

Previous Page Next Page

Bootstrap Accordion

In this tutorial you will learn how to create accordion with Bootstrap.

Creating Accordion Widget with Bootstrap

Accordion widgets and menus are widely used on the websites to manage the large amount of content and navigation lists. With Bootstrap collapse plugin you can either create accordion or a simple collapsible panel without writing any JavaScript code.

The following example will show you how to build a simple accordion widget using the Bootstrap collapsible plugin and the panel component.

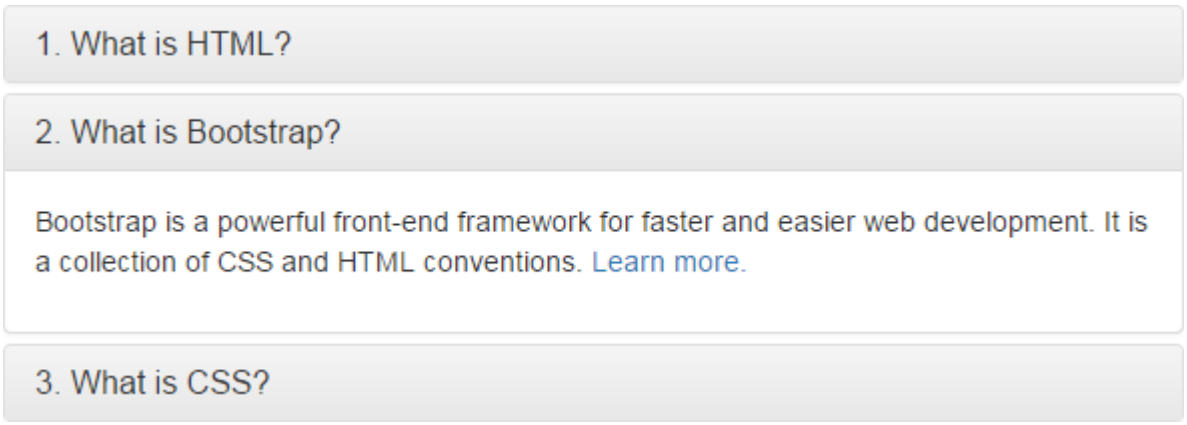
Example

Try this code »

```
<div id="accordion" class="panel-group">
    <div class="panel panel-default">
        <div class="panel-heading">
            <h4 class="panel-title">
                <a data-toggle="collapse" data-parent="#accordion" href="#collapseOne">1. What is HTML?</a>
            </h4>
        </div>
        <div id="collapseOne" class="panel-collapse collapse">
            <div class="panel-body">
                <p>HTML stands for HyperText Markup Language. HTML is the main markup language for describing the structure of Web pages. <a href="http://www.tutorialrepublic.com/html-tutorial/" target="_blank">Learn more.</a></p>
            </div>
        </div>
    </div>
    <div class="panel panel-default">
        <div class="panel-heading">
            <h4 class="panel-title">
```

```
                <a data-toggle="collapse" data-parent="#accordion" href="#collapseTwo">2. What is Bootstrap?</a>
            </h4>
        </div>
        <div id="collapseTwo" class="panel-collapse collapse in">
            <div class="panel-body">
                <p>Bootstrap is a powerful front-end framework for faster and easier web development. It is a collection of CSS and HTML conventions. <a href="http://www.tutorialrepublic.com/twitter-bootstrap-tutorial/" target="_blank">Learn more.</a></p>
            </div>
        </div>
    </div>
    <div class="panel panel-default">
        <div class="panel-heading">
            <h4 class="panel-title">
                <a data-toggle="collapse" data-parent="#accordion" href="#collapseThree">3. What is CSS?</a>
            </h4>
        </div>
        <div id="collapseThree" class="panel-collapse collapse">
            <div class="panel-body">
                <p>CSS stands for Cascading Style Sheet. CSS allows you to specify various style properties for a given HTML element such as colors, backgrounds, fonts etc. <a href="http://www.tutorialrepublic.com/css-tutorial/" target="_blank">Learn more.</a></p>
            </div>
        </div>
    </div>
</div>
```

— The output of the above example will look something like this:



Expanding and Collapsing Elements via Data Attributes

You can use the Bootstrap collapse feature for expanding and collapsing any specific element via data attributes without using the accordion markup.

Example
Try this code »

```
<!-- Trigger Button HTML -->
<input type="button" class="btn" data-toggle="collapse" data-target="#toggleDemo" value="Toggle Button">
<!-- Collapsible Element HTML -->
<div id="toggleDemo" class="collapse in"><p>This is a simple example of expanding and collapsing individual element via data attribute. Click on the <b>Toggle Button</b> button to see the effect.</p></div>
```

We've just created a collapsible control without writing any JavaScript code. Well, let's go through each part of this code one by one for a better understanding.

Explanation of Code

The Bootstrap collapse plugin basically requires the two elements to work properly — the controller element such as a button or hyperlink by clicking on which you want to collapse the other element, and the collapsible element itself.

- The `data-toggle="collapse"` attribute (*line no-2*) is added to the controller element along with a attribute `data-target` (for buttons) or href (for anchors) to automatically assign the control of a collapsible element.
- The `data-target` or `href` attribute accepts a CSS selector to apply the collapse to a specific element. Be sure to add the class `"collapse"` to the collapsible element.
- You can optionally add the class `"in"` (*line no-4*) to the collapsible element in addition to the class `"collapse"` to make it open by default.

To make the collapsible controls to work in group like accordion menu, you can utilize the Bootstrap panel component as demonstrated in the previous example.

Expanding and Collapsing Elements via JavaScript

You may also expand and collapse an individual element manually via JavaScript — just call the `collapse()` Bootstrap method with the `"id"` or `"class"` selector of the collapsible element in your JavaScript code.

Example
Try this code »

```
<!-- Trigger Button HTML -->
<input type="button" class="btn" value="Toggle Button">
```

```
<!-- Collapsible Element HTML -->
<div id="toggleDemo" class="collapse in"><p>This is a simple example of expanding and collapsing individual element via JavaScript. Click on the <b>Simple Collapsible</b> button to see the effect.</p></div>
```

Options

There are certain options which may be passed to `collapse()` Bootstrap method to customize the functionality of a collapsible element.

Name	Type	Default Value	Description
parent	selector	false	All other collapsible elements under the specified parent will be closed while this collapsible item is shown on invocation.
toggle	boolean	true	Toggles the collapsible element on invocation.

You can also set these options using the data attributes on accordion — just append the option name to `data-`, like `data-parent="#myAccordion"`, `data-toggle="false"` etc. as demonstrated in the basic implementation.

Methods

These are the standard bootstrap's collapse methods:

.collapse(options)

This method activates your content as a collapsible element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".btn").click(function(){
        $("#myCollapsible").collapse({
            toggle: false
        });
    });
});
</script>
```

.collapse('toggle')

This method toggles (show or hide) a collapsible element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".toggle-btn").click(function(){
        $("#myCollapsible").collapse('toggle');
    });
});
</script>
```

.collapse('show')

This method shows a collapsible element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".show-btn").click(function(){
        $("#myCollapsible").collapse('show');
    });
});
</script>
```

.collapse('hide')

This method hides a collapsible element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".hide-btn").click(function(){
        $("#myCollapsible").collapse('hide');
    });
});
</script>
```

Events

Bootstrap's collapse class includes few events for hooking into collapse functionality.

Event	Description
show.bs.collapse	This event fires immediately when the show instance method is called.
shown.bs.collapse	This event is fired when a collapse element has been made visible to the user. It will wait until the CSS transition process has been fully completed before getting fired.
hide.bs.collapse	This event is fired immediately when the hide method has been called.
hidden.bs.collapse	This event is fired when a collapse element has been hidden from the user. It will wait until the CSS transition process has been fully completed before getting fired.

The following example displays an alert message to the user when sliding transition of a collapsible element has been fully completed.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myCollapsible").on('hidden.bs.collapse', function(){
        alert("Collapsible element has been completely closed.");
    });
});
</script>
```

Previous Page Next Page

Bootstrap Carousel

In this tutorial you will learn how to create carousels with Bootstrap.

Creating Carousels with Bootstrap

The carousel also known as slideshow or image slider is some of the best way of showcasing the huge amount of contents within a small space on the web pages. It is a dynamic presentation of contents where text and images are made visible or accessible to the user by cycling through several items. The following example will show you how to build a simple carousel like image rotator using the Bootstrap carousel plugin.

Example

Try this code »

```
<div id="myCarousel" class="carousel slide" data-ride="carousel">
    <!-- 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 carousel items -->
    <div class="carousel-inner">
        <div class="item active">
            <img src="images/slide1.png" alt="First Slide">
        </div>
        <div class="item">
            <img src="images/slide2.png" alt="Second Slide">
        </div>
        <div class="item">
            <img src="images/slide3.png" alt="Third Slide">
        </div>
    </div>
    <!-- Carousel controls -->
    <a class="carousel-control left" href="#myCarousel" data-slide="prev">
        <span class="glyphicon glyphicon-chevron-left"></span>
    </a>
    <a class="carousel-control right" href="#myCarousel" data-slide="next">
        <span class="glyphicon glyphicon-chevron-right"></span>
    </a>
</div>
```

— The output of the above example will look something like this:

You can also add captions such as heading or description to the individual slides of the carousel, please check out the next example.

Activate Carousels via Data Attributes

With Bootstrap you can create carousels very easily via data attributes without writing a single line of JavaScript code. Let's go through the following example:

Example

Try this code »

```
<div id="myCarousel" class="carousel slide" data-interval="3000" data-ride="carousel">
    <!-- 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>
    <!-- Carousel items -->
    <div class="carousel-inner">
        <div class="item active">
            <img src="images/slide1.png" alt="First Slide">
            <div class="carousel-caption">
                <h3>First slide label</h3>
                <p>Lorem ipsum dolor sit amet...</p>
            </div>
        </div>
        <div class="item">
            <img src="images/slide2.png" alt="Second Slide">
            <div class="carousel-caption">
                <h3>Second slide label</h3>
                <p>Aliquam sit amet gravida nibh, facilisis...</p>
            </div>
        </div>
        <div class="item">
            <img src="images/slide3.png" alt="Third Slide">
            <div class="carousel-caption">
                <h3>Third slide label</h3>
                <p>Praesent commodo cursus magna vel...</p>
            </div>
        </div>
    </div>
    <!-- Carousel nav -->
```

```
    <a class="carousel-control left" href="#myCarousel" data-slide="prev">
        <span class="glyphicon glyphicon-chevron-left"></span>
    </a>
    <a class="carousel-control right" href="#myCarousel" data-slide="next">
        <span class="glyphicon glyphicon-chevron-right"></span>
    </a>
</div>
```

You might be wondering what this code was all about. Ok, let's go through each part of this carousel code one by one for a better understanding.

Explanation of Code

The Bootstrap carousel has basically three components — carousel indicators (*small circles*), carousel controls (*previous and next arrows*) and the carousel items or slides.

- The outermost container of every carousel requires a unique `id` (in our case `id="myCarousel"`) so that it can be targeted by the carousel indicators (*line no-4,5,6*) and carousel controls (*line no-33,36*) to function properly.
- The `data-ride="carousel"` attribute of the `.carousel` element tells the Bootstrap to start animating the carousel immediately when the page load. Whereas the `data-interval` attribute specifies the time delay between two slides.
- The `.data-slide-to` attribute (*line no-4,5,6*) move the slide position to a particular item (index beginning with `0`) when clicking on the specific carousel indicator.
- The slides are specified within the `.carousel-inner` (*line no-9*) and the content of each slide is defined within the `.item` element that can be text and images.
- The `data-slide` attribute on carousel controls (*line no-33,36*) accepts the keywords "`prev`" or "`next`", which alters the slide position relative to its current position.

Rest of the thing is self explanatory, such as the `.carousel` element specifies the Bootstrap carousel, the `.carousel-indicators` element indicates how many slides are there in the carousel and which slide is currently active, the `.carousel-caption` element used within the `.item` element defines the caption for that slide etc.

Tip: It is required to add the class `.active` to one of the slides (i.e. on the `.item` element). Otherwise, the carousel will not be visible.

Note: The `.slide` class adds CSS slide transition animation to the carousel that makes the items slide when showing the new item. But it doesn't work in Internet Explorer 8 & 9 due to lack of support of the necessary CSS3 properties.

Activate Carousels via JavaScript

You may also activate a carousel manually using the JavaScript — just call the `carousel()` method with the `"id"` or `"class"` selector of the wrapper element in your JavaScript code.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
      $("#myCarousel").carousel();
});
</script>
```

Options

There are certain options which may be passed to `carousel()` Bootstrap method to customize the functionality of a carousel widget.

Name	Type	Default Value	Description
interval	number	5000	Specifies the amount of time to delay (in milliseconds) between one slide to another in automatic cycling. If false, carousel will not automatically cycle.
pause	string	"hover"	Pauses the cycling of the carousel on mouse enter and resumes the cycling of the carousel on mouse leave.
wrap	boolean	true	Specifies whether the carousel should cycle continuously or have hard stops.
keyboard	boolean	true	Specifies whether the carousel should react to keyboard events.

Check out the following example to see these options works:

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
      $("#myCarousel").carousel({
          interval : 3000,
          pause: false
```

```
        });
});
</script>
```

You can also set these options using the data attributes on carousel — just append the option name to `data-`, like `data-interval="3000"`, `data-pause="hover"` as demonstrated in the section of data attribute implementation.

Disable Auto Sliding in Bootstrap Carousel

By default Bootstrap carousel is started playing or sliding automatically when the page loads. However, you can turn off this auto sliding by simply setting the carousel "`interval`" option to "`false`" via JavaScript. You can also set this option using the data attribute like `data-interval="false"` on the `.carousel` element.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
      $("#myCarousel").carousel({
          interval : false
      });
});
</script>
```

Methods

These are the standard bootstrap's carousels methods:

.carousel(options)

This method initializes the carousel with optional options and starts cycling through items.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
      $("#myCarousel").carousel({
          interval : 3000
      });
});
```

```
</script>
```

.carousel('cycle')

This method start carousel for cycling through the items from left to right.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".start-slide").click(function(){
        $("#myCarousel").carousel('cycle');
    });
});
</script>
```

.carousel('pause')

This method stops the carousel from cycling through items.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".pause-slide").click(function(){
        $("#myCarousel").carousel('pause');
    });
});
</script>
```

.carousel(number)

This method cycles the carousel to a particular frame (start with `0`, similar to an array).

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".slide-three").click(function(){
        $("#myCarousel").carousel(3);
    });
});
</script>
```

.carousel('prev')

This method cycles the carousel to the previous item.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".prev-slide").click(function(){
        $("#myCarousel").carousel('prev');
    });
});
</script>
```

.carousel('next')

This method cycles the carousel to the next item.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $(".next-slide").click(function(){
        $("#myCarousel").carousel('next');
    });
});
</script>
```

Events

Bootstrap's carousel class includes few events for hooking into carousel functionality.

Event	Description
slide.bs.carousel	This event fires immediately when the slide instance method is called.
slid.bs.carousel	This event is fired when the carousel has completed its slide transition.

The following example displays an alert message to the user when sliding transition of a carousel item has been fully completed.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('#myCarousel').on('slid.bs.carousel', function () {
        alert("The sliding transition of previous carousel item has been fully completed.");
    });
});
</script>
```

Previous Page Next Page

Bootstrap Typeahead

In this tutorial you will learn how to create typeaheads with Bootstrap.

Creating Typeaheads with Bootstrap

The typeahead input fields are very popular in modern web forms. The main purpose of using typeahead is to improve the user experience by supplying hints or a list of possible choices based on the text they've entered while filling a form or searching something — like the Google instant search. It also saves time and reduces the number of potential errors, because the user has less likelihood of making a spelling mistake.

Note: Typeahead plugin has been dropped from the latest version of Bootstrap (v3.0+), in favor of using Twitter typeahead.

Twitter typeaheads is a fast and fully-featured autocomplete library inspired by twitter.com's autocomplete search functionality. To create Twitter typeaheads first download typeahead.js from GitHub and include in your project, after that you can turn any text-based `<input>` element (i.e. `input[type="text"]`) into a typeahead.

Creating Twitter Typeahead with Local Dataset

The following example will show you how to create Twitter typeahead with local dataset.

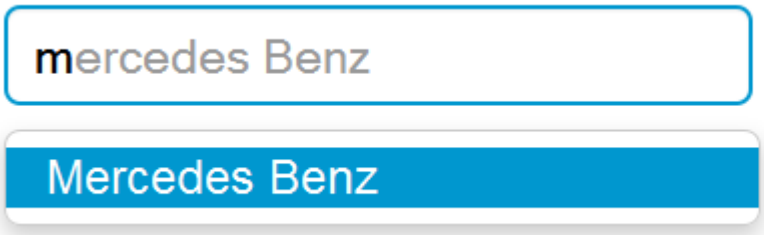
Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('input.typeahead').typeahead({
      name: 'accounts',
```

```
        local: ['Audi', 'BMW', 'Bugatti', 'Ferrari', 'Ford',
    'Lamborghini', 'Mercedes Benz', 'Porsche', 'Rolls-Royce', 'Volkswagen']
    });
});
</script>
```

— The output of the above example will look something like this:



Tip: Set `autocomplete="off"` for the input box if you want to prevent default browser menus from appearing over the Bootstrap type-ahead dropdown.

Creating Twitter Typeahead External Dataset

You can also specify external dataset through a URL pointing to a JSON file containing an array of datums. The individual units that compose datasets are called datums.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('input.typeahead').typeahead({
        name: 'countries',
        prefetch: 'data/countries.json',
        limit: 10
    });
});
</script>
```

Previous Page Next Page

Bootstrap ScrollSpy

In this tutorial you will learn how to create scrollspy with Bootstrap.

Creating ScrollSpy with Bootstrap

The Bootstrap scrollspy is a navigation mechanism that automatically highlights the nav links based on the scroll position to indicate the visitor where they are currently on the page. The scrollspy will make your web page more elegant and accessible, if you are using the bookmark links for directing the visitors to the different sections of a page that has a huge amount of content. Here's a typical example of Bootstrap scrollspy.

Example

Try this code »

```
<body data-spy="scroll" data-target="#myScrollspy">
<div class="container">
    <div class="row">
        <div class="col-sm-3" id="myScrollspy">
            <ul class="nav nav-tabs nav-stacked" data-offset-top="120" data-spy="affix">
                <li class="active"><a href="#section1">Section One</a></li>
                <li><a href="#section2">Section Two</a></li>
                <li><a href="#section3">Section Three</a></li>
            </ul>
        </div>
        <div class="col-sm-9">
            <div id="section1">
                <h2>Section One</h2>
                <p>This is section one content…</p>
            </div>
            <hr>
            <div id="section2">
                <h2>Section Two</h2>
                <p>This is section two content…</p>
            </div>
            <hr>
            <div id="section3">
                <h2>Section Three</h2>
                <p>This is section three content…</p>
            </div>
        </div>
    </div>
</div>
</body>
```

Note: The Bootstrap scrollspy plugin requires the use of a Bootstrap nav component (e.g. navbar, nav tabs or pills) for proper highlighting of active links.

Creating ScrollSpy via Data Attributes

You can easily add scrollspy behavior to your topbar navigation via data attributes without writing a single line of JavaScript code. Let's check out the following example:

Example

Try this code »

```
<body data-spy="scroll" data-target="#myNavbar" data-offset="70">
    <nav id="myNavbar" class="navbar navbar-inverse navbar-fixed-top" role="navigation">
        <div class="container">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbarCollapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">Scrollspy</a>
            </div>
            <!-- Collection of nav links, forms, and other content for toggling -->
            <div class="collapse navbar-collapse" id="navbarCollapse">
                <ul class="nav navbar-nav">
                    <li class="active"><a href="#section1">Section 1</a></li>
                    <li><a href="#section2">Section 2</a></li>
                    <li><a href="#section3">Section 3</a></li>
                    <li class="dropdown"><a href="#" class="dropdown-toggle" data-toggle="dropdown">Section 4<b class="caret"></b></a>
                        <ul class="dropdown-menu">
                            <li><a href="#section4dot1">Section 4.1</a></li>
                            <li><a href="#section4dot2">Section 4.2</a></li>
                            <li><a href="#section4dot3">Section 4.3</a></li>
                        </ul>
                    </li>
                    <li><a href="#section5">Section 5</a></li>
                </ul>
            </div>
        </div>
    </nav>
    <div class="container">
        <div id="section1">
```

```
            <h2>Section 1</h2>
            <p>This is section 1 content…</p>
        </div>
        <hr>
        <div id="section2">
            <h2>Section 2</h2>
            <p>This is section 2 content…</p>
        </div>
        <hr>
        <div id="section3">
            <h2>Section 3</h2>
            <p>This is section 3 content…</p>
        </div>
        <hr>
        <h2>Section 4</h2>
        <p>This is section 4 content</p>
        <div id="section4dot1">
            <h3>Section 4.1</h3>
            <p>This is section 4.1 content…</p>
        </div>
        <div id="section4dot2">
            <h3>Section 4.2</h3>
            <p>This is section 4.2 content…</p>
        </div>
        <div id="section4dot3">
            <h3>Section 4.3</h3>
            <p>This is section 4.3 content…</p>
        </div>
        <hr>
        <div id="section5">
            <h2>Section 5</h2>
            <p>This is section 5 content…</p>
        </div>
    </div>
</body>
```

You might be thinking what this code was all about. Ok, let's go through each part of this scrollspy code one by one for a better understanding.

Explanation of Code

The Bootstrap scrollspy has basically two components — the target nav (e.g. navbar, nav tabs or pills) and the scrollable area to spy on, which is often the `<body>` section.

- The `data-spy="scroll"` attribute (*line no-01*) is applied to the scrollable element that is being spied on, which is the `<body>` element.

- The `data-target` attribute is added on the scrollable element with the ID or class of the parent element of the Bootstrap `.nav` component so that nav links can be targeted by the scrollspy for highlighting purpose.
- The optional `data-offset` attribute specifies the number of pixels to offset from top when calculating the position of scroll. Adjust the offset value if the targeted links are highlighting too early or too late. The default value is 10 pixels.

Rest of the thing is self explanatory, such as the `.navbar` element specifies a Bootstrap navbar, the element `<div id="section1"></div>` (*line no-33*) create a bookmark with the `id` attribute, whereas the element `<a href="#section1">Section 1</a>` (*line no-17*) add a link to this bookmark, from within the same page, and so on.

Creating ScrollSpy via JavaScript

You may also add scrollspy manually using the JavaScript — just call the `scrollspy()` Bootstrap method with the `"id"` or `"class"` selector of the navbar in your JavaScript code.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("body").scrollspy({
        target: "#myNavbar",
        offset: 70
    })
});
</script>
```

Options

There are certain options which may be passed to `scrollspy()` Bootstrap method to customize the functionality of a scrollspy.

Name	Type	Default Value	Description
offset	number	10	Number of pixels to offset from top when calculating position of scroll.

You can also set this options for scrollspy using the data attributes — just append the option name to `data-`, like `data-offset="0"`.

Methods

These are the standard bootstrap's scrollspy methods:

.scrollspy('refresh')

When using scrollspy in conjunction with adding or removing of elements from the DOM, you'll need to call the refresh method like this:

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $('[data-spy="scroll"]').each(function(){
        var $spy = $(this).scrollspy('refresh');
    });
});
</script>
```

Events

Bootstrap's scrollspy class includes few events for hooking into scrollspy functionality.

Event	Description
activate.bs.scrollspy	This event fires whenever a new item becomes activated by the scrollspy.

The following example displays an alert message to the user when a new item becomes highlighted by the scrollspy.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myNavbar").on("activate.bs.scrollspy", function(){
        var currentItem = $(".nav li.active > a").text();
        $("#info").empty().html("Currently you are viewing - " +
currentItem);
    })
});
</script>
```

Previous Page Next Page

Bootstrap Affix

Bootstrap affix plugin is used to add affix behavior to any element.

Creating Pinned Element with Bootstrap

In this tutorial you will learn how to apply pinning i.e. fixed positioning on a navbar or any other element automatically when they are scrolled beyond a certain distance as well as toggle its pinning on and off using the Bootstrap affix plugin. The pinning of an element is enabled through changing the value of its `position` CSS property from `static` to `fixed`.

To do this, the affix plugin toggles between three classes: `.affix`, `.affix-top`, and `.affix-bottom`. Each class represents a particular state.

- Initially, the plugin adds `.affix-top` or `.affix-bottom` class to indicate the element is in its top-most or bottom-most position.
- When the element scrolling past the offset limit provided by the `"data-offset-"` attribute the plugin replaces the `.affix-top` or `.affix-bottom` class with the `.affix` class (sets `position: fixed;`), which trigger the actual affixing.
- At this point the appropriate CSS `top` or `bottom` property is required to determine the position of affix element on the viewport.

Let's check out the following example to see it in real action.

Enable Affix via Data Attributes

You can easily add affix behavior to any element — just add `data-spy="affix"` to the element you want to spy on. Then use `"data-offset-"` attributes to define when to toggle the pinning of an element 'on' and 'off'.

Example

Try this code »

```
<ul class="nav nav-tabs nav-stacked" data-spy="affix" data-offset-top="195">
    <li class="active"><a href="#one">Section One</a></li>
    <li><a href="#two">Section Two</a></li>
    <li><a href="#three">Section Three</a></li>
</ul>
```

Note: The `"data-offset-"` attributes only specify how many pixels that you must scroll in order to toggle the pinning of an element, it did not set the position of pinned element. You must define the `top` or `bottom` CSS property for the pinned element specifically in your style sheet to set its position in the viewport.

Enable Affix via JavaScript

You may also enable the affix plugin manually using the JavaScript — just call the `affix()` method with the `"id"` or `"class"` selector of the required element in your JavaScript code.

Example

Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myNav").affix({
        offset: {
            top: $(".header").outerHeight(true)
        }
    });
});
</script>
```

Options

There are certain options which may be passed to `affix()` Bootstrap method to customize the functionality of the affix plugin.

Name	Type	Default Value	Description
offset	number | function | object	10	Specify the number of pixels to offset from screen when calculating position of scroll. If a single number is provided, the offset will be applied in both top and bottom directions. To set offset for a single direction, or multiple unique offsets — just provide an object like `offset: {top:50, bottom:100}` You can also use a function if you want to dynamically provide an offset in case of responsive designs.
target	selector | node | jQuery element	the `window` object	Specifies the target element of the affix.

You can also set these options for affix using the data attributes — just append the option name to `data-`, like `data-offset-top="195"`.

Methods

These are the standard bootstrap's affix methods:

.affix(options)

This method activates your content as affixed content. Accepts an optional options `object`, like `offset: {top: 10}` or `offset: {top:50, bottom:100}`.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myNav").affix({
        offset: {
            bottom: 195
        }
    });
});
</script>
```

Events

Bootstrap's affix class includes few events for hooking into modal functionality.

Event	Description
affix.bs.affix	This event fires immediately before the element has been affixed.
affixed.bs.affix	This event is fired after the element has been affixed.
affix-top.bs.affix	This event fires immediately before the element has been affixed to top.
affixed-top.bs.affix	This event is fired after the element has been affixed to top.
affix-bottom.bs.affix	This event fires immediately before the element has been affixed-bottom.
affixed-bottom.bs.affix	This event is fired after the element has been affixed to bottom.

The following example displays an alert message when navigation menu has been affixed.

Example
Try this code »

```
<script type="text/javascript">
$(document).ready(function(){
    $("#myNav").on('affixed.bs.affix', function(){
        alert("The navigation menu has been affixed. Now it doesn't scroll with the page.");
    });
});
</script>
```

Previous Page Next Page

Bootstrap 3 Examples

This section contains a whole bunch of examples demonstrating the various Bootstrap components and its features in real action.

Bootstrap Grid System

- Bootstrap one column grid layout for all devices
- Bootstrap two column grid layouts for all devices
- Bootstrap two column grid layouts for tablets and desktops
- Bootstrap two column grid layouts for tablets in landscape mode and desktops
- Bootstrap three column grid layouts for all devices
- Bootstrap three column grid for layouts tablets and desktops
- Bootstrap three column grid layouts for tablets in landscape mode and desktops
- Bootstrap four column convertible grid layout with multiple views - 4 columns on desktops, 2 columns on tablets, one column on mobiles
- Bootstrap nested grid columns layout
- Bootstrap multi column grid layout for all devices
- Bootstrap fixed layout
- Bootstrap fluid layout
- Bootstrap responsive layout

Bootstrap Typography

- Bootstrap headings
- Bootstrap headings with secondary text
- Bootstrap page header
- Bootstrap paragraph
- Bootstrap text formatting
- Bootstrap text alignment
- Bootstrap text transformation
- Bootstrap text emphasis classes
- Formatting blockquotes in Bootstrap
- Bootstrap blockquotes right aligned

Bootstrap Tables

- Creating tables with Bootstrap
- Bootstrap table with alternate background
- Bootstrap table with borders
- Enable hover state on table rows
- Bootstrap condensed tables
- Optional emphasis classes for table rows
- Bootstrap responsive tables

Bootstrap Lists

- Creating lists with Bootstrap
- Bootstrap unstyled ordered and unordered list
- Placing ordered and unordered list items inline
- Creating horizontal definition lists with Bootstrap
- Bootstrap list groups
- Bootstrap list group with linked items
- Bootstrap linked list group with custom content
- Bootstrap list groups with emphasis classes
- Bootstrap linked list groups with emphasis classes

Bootstrap Forms

- Creating vertical form with Bootstrap
- Creating horizontal form with Bootstrap
- Creating inline form with Bootstrap
- Bootstrap static form control
- Height sizing of Inputs and select boxes with Bootstrap
- Grid sizing of form controls with Bootstrap
- Creating prepended and appended inputs with Bootstrap
- Creating button dropdowns with Bootstrap
- Creating segmented dropdown button groups with Bootstrap
- Creating disabled inputs with Bootstrap
- Creating readonly inputs with Bootstrap
- Creating disabled fieldsets with Bootstrap
- Placing help text around form controls
- Bootstrap form validation states
- Bootstrap form validation states with feedback icons
- Supported form controls in Bootstrap

Bootstrap Images

- Making thumbnails, rounded corner and circular images with Bootstrap

- Making responsive images with Bootstrap
- Making responsive videos with Bootstrap
- Creating thumbnails gallery with Bootstrap
- Creating thumbnails gallery with content with Bootstrap
- Bootstrap media objects
- Bootstrap media list

Bootstrap Icons

- Using Bootstrap icons inside buttons
- Using Bootstrap icons inside dropdowns
- Using Bootstrap icons inside extended form controls
- Using Bootstrap icons inside nav components

Bootstrap Navs

- Creating basic tabs with Bootstrap
- Creating basic pills nav with Bootstrap
- Creating stacked pills nav with Bootstrap
- Bootstrap tabs with dropdown menus
- Bootstrap pills with dropdown menus
- Creating justified tabs and pills with Bootstrap
- Disable links inside Bootstrap navs

Bootstrap Navbar

- Creating navbar with Bootstrap
- Bootstrap navbar with dropdown and search form
- Creating Bootstrap navbar fixed to top
- Creating Bootstrap navbar fixed to bottom
- Creating static top navbar with Bootstrap
- Bootstrap navbar with search from
- Creating inverted variation of responsive navbar with Bootstrap

Bootstrap Breadcrumbs and Pagination

- Creating breadcrumbs with Bootstrap
- Creating pagination with Bootstrap
- Bootstrap pagination with disabled and active states
- Changing the sizes of Bootstrap pagination
- Creating pager with Bootstrap
- Side alignment of Bootstrap pager

Bootstrap Labels and Badges

- Creating inline labels with Bootstrap
- Bootstrap inline labels with emphasis classes
- Creating inline badges with Bootstrap

Bootstrap Progress Bars

- Creating progress bars with Bootstrap
- Creating Bootstrap progress bar with label
- Creating stripped progress bars with Bootstrap
- Creating animated progress bars with Bootstrap
- Creating stacked progress bars with Bootstrap
- Bootstrap progress bars with emphasis classes
- Bootstrap striped progress bars with emphasis classes

Bootstrap Utility Components and Classes

- Bootstrap jumbotron
- Bootstrap wells
- Bootstrap contextual background classes
- Bootstrap close icon
- Bootstrap caret icon for dropdowns
- Bootstrap center alignment of content block
- Bootstrap show hide content
- Bootstrap `.sr-only` helper class
- Bootstrap `.text-hide` helper class
- Bootstrap `.pull-left` helper class
- Bootstrap `.pull-right` helper class
- Bootstrap `.clearfix` helper class

Bootstrap Buttons

- Available buttons styles in Bootstrap
- Changing the sizes of Bootstrap buttons
- Creating block level buttons with Bootstrap
- Creating disabled Bootstrap buttons using the anchor element
- Creating disabled Bootstrap buttons using the input and button element
- Creating stateful buttons with Bootstrap
- Creating single toggle button with Bootstrap
- Creating button groups with Bootstrap
- Creating buttons checkbox with Bootstrap
- Bootstrap buttons checkbox with pre checked options
- Creating buttons radio with Bootstrap
- Bootstrap buttons radio with pre selected option
- Bootstrap button toolbar
- Height sizing of Bootstrap button groups

- Creating justified button groups with Bootstrap
- Enable Bootstrap buttons via JavaScript
- Bootstrap `$().button('toggle')` method
- Bootstrap `$().button('loading')` method
- Bootstrap `$().button('reset')` method
- Bootstrap `$().button(string)` method

Bootstrap Dropdowns

- Adding dropdowns to Bootstrap navbar
- Adding dropdowns to Bootstrap navs
- Adding dropdowns to Bootstrap buttons
- Adding dropdowns to split buttons
- Adding dropdowns to button groups
- Adding dropdowns to any element via data attributes
- Adding dropdowns to any element via JavaScript
- Bootstrap `$().dropdown('toggle')` method

Bootstrap Tooltips

- Creating tooltips with Bootstrap
- Setting the position of Bootstrap tooltips via data attributes
- Setting the position of Bootstrap tooltips via JavaScript
- Setting the title text of Bootstrap tooltips via JavaScript
- Inserting HTML content inside the Bootstrap tooltips via JavaScript
- Setting the show hide timing of Bootstrap tooltips via JavaScript
- Creating the custom template for Bootstrap tooltips via JavaScript
- Setting the container element for Bootstrap tooltips via JavaScript
- Bootstrap `$().tooltip(options)` method
- Bootstrap `.tooltip('show')` method
- Bootstrap `.tooltip('hide')` method
- Bootstrap `.tooltip('toggle')` method
- Bootstrap `.tooltip('destroy')` method

Bootstrap Popovers

- Creating popovers with Bootstrap
- Setting the position of Bootstrap popovers via data attributes
- Setting the position of Bootstrap popovers via JavaScript
- Setting the title text of Bootstrap popovers via JavaScript
- Triggering the Bootstrap popovers on mouse hover instead of click via JavaScript
- Inserting HTML content inside the Bootstrap popovers via JavaScript
- Setting the show hide timing of Bootstrap popovers via JavaScript
- Creating the custom template for Bootstrap popovers via JavaScript
- Setting the container element for Bootstrap popovers via JavaScript

- Bootstrap `$().popover(options)` method
- Bootstrap `.popover('show')` method
- Bootstrap `.popover('hide')` method
- Bootstrap `.popover('toggle')` method
- Bootstrap `.popover('destroy')` method

Bootstrap Alerts Messages

- Creating alert messages with Bootstrap
- Creating error or danger alert messages with Bootstrap
- Creating success or confirmation alert messages with Bootstrap
- Creating information alert messages with Bootstrap
- Closing Bootstrap alert messages via data attributes
- Closing Bootstrap alert messages via JavaScript
- Bootstrap `$().alert()` method
- Bootstrap `$().alert('close')` method
- Display a message when Bootstrap alert box has been completely closed

Bootstrap Tabs

- Creating a basic tab component with Bootstrap
- Creating Bootstrap dynamic tabs via data attributes
- Creating Bootstrap dynamic tabs via via JavaScript
- Activate individual Bootstrap tabs via JavaScript
- Setting the directions of Bootstrap tabs
- Bootstrap `$().tab` method
- Display the names of active tab and previous tab

Bootstrap Modals

- Creating modals with Bootstrap
- Launching Bootstrap modal box via data attributes
- Launching Bootstrap modal box via JavaScript
- Changing the sizes Bootstrap modals
- Changing the Bootstrap modal content based on the trigger button
- Prevent Bootstrap modal from disappearing on click of the dark area via JavaScript
- Prevent Bootstrap modal from hiding on press of the escape key via JavaScript
- Loading content in Bootstrap modals via remote URL via data attributes
- Loading content in Bootstrap modals via remote URL via JavaScript
- Bootstrap `.modal(options)` method
- Bootstrap `.modal('toggle')` method
- Bootstrap `.modal('show')` method
- Bootstrap `.modal('hide')` method
- Display a message when Bootstrap modal window has been completely closed

Bootstrap Accordion

- Creating accordion widget with Bootstrap
- Expanding and collapsing elements via data attributes
- Expanding and collapsing elements via JavaScript
- Bootstrap `.collapse(options)` method
- Bootstrap `.collapse('toggle')` method
- Bootstrap `.collapse('show')` method
- Bootstrap `.collapse('hide')` method
- Display an alert message when collapsible element has been completely closed

Bootstrap Carousel

- Creating carousel with Bootstrap
- Activate Bootstrap carousels via data attributes
- Activate Bootstrap carousels via JavaScript
- Removing auto-sliding from Bootstrap carousel
- Setting options to Bootstrap `carousel()` method
- Bootstrap `.carousel(options)` method
- Bootstrap `.carousel('cycle')` method
- Bootstrap `.carousel(number)` method
- Bootstrap `.carousel('prev')` method
- Bootstrap `.carousel('next')` method
- Display an alert message when sliding transition of a Bootstrap carousel item has been fully completed

Bootstrap Typeahead

- Creating Twitter typeahead with local dataset
- Creating Twitter typeahead with external dataset

Bootstrap ScrollSpy

- Creating scrollspy with Bootstrap
- Adding scrollspy behavior to the navbar via data attributes
- Adding scrollspy behavior to the navbar via JavaScript
- Bootstrap `.scrollspy('refresh')` method
- Display the name of menu items when it is highlighted by the scrollspy
- Enable smooth scrolling in Bootstrap scrollspy via JavaScript

Bootstrap Affix

- Adding Bootstrap affix behavior to the element via data attributes
- Adding Bootstrap affix behavior to the element via JavaScript

- Display an alert message when navigation menu has been affixed

Previous Page Next Page