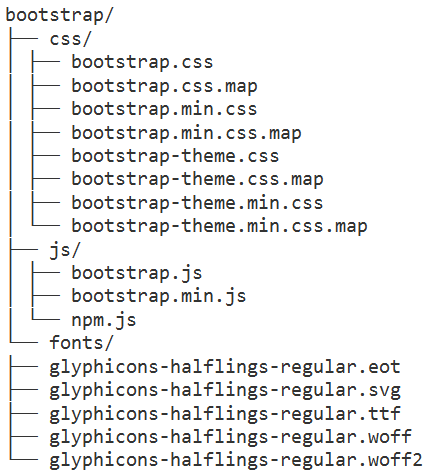
**Bootstraps:-**

1.What is bootstrap

**Bootstrap** is a free, open-source and is the most popular HTML, CSS, and JavaScript framework developed by twitter for creating responsive web applications.    
  
It includes HTML and CSS based design templates for common user interface components like Buttons, Dropdowns, Typography, Tabs, Forms, Tables, Navigations, Alerts, Modals, Accordion, Carousel etc. along with optional JavaScript extensions.   
  
Bootstrap framework is based on open standards - HTML, CSS and JavaScript. This means bootstrap can be used with any server side technology and any platform. You can use it with any web application built with any server side technology like ASP.NET, JAVA, PHP etc.   
  
**What are the advantages of using bootstrap**   
  
**Supports responsive design :**One of the greatest advantages of using bootstrap is that it helps us create responsive web applications faster and easier. So the obvious question that comes to our mind is, what is a responsive web application? A responsive web application automatically adapts to different screen sizes (i.e desktop computers, laptops, tablets. mobile phones etc). A responsive application provides optimal viewing and interaction experience i.e easy reading and navigation with a minimum of resizing, panning, and scrolling across a wide range of devices. So you don't have to worry about your application not being compatible with multiple devices.   
  
The images at the following links shows how a responsive and non-responsive application looks like on a mobile device.  
<http://www.webnersolutions.com/wp-content/uploads/2015/08/responsive-vs-non-responsive-web-design1.jpg>   
  
At the following link is a live responsive application example. Notice as we resize the browser, the content automatically adapts to the screen size.  
<http://bootstrapdocs.com/v3.0.3/docs/examples/jumbotron/>   
  
**Saves lot of development time :**One of the biggest advantages of using Bootstrap is that it saves lot of development time. Instead of writing code from the scratch, bootstrap offers ready made blocks of code that you can use and customize to suit your application requirements. There are also many websites out there that offer free and paid Bootstrap themes that saves even more development time.   
  
**Consistency :** Bootstrap was developed by Twitter to encourage consistency across thier internal tools by giving their developers a centralised development code. Since all the developers are working using a centralised code, the end result is consistent regardless of who’s working on the project and which web browser is being used.   
  
**Customizable :** If you are using only a few features of bootstrap, you can customize to download only those features using the following bootstrap customize page.  
<http://getbootstrap.com/customize/>  
  
**Support :** As Bootstrap is the most popular framework, it has a very large community base and excellent documentation. Bootstrap's excellent documentation, examples and demos helps a developer learn bootstrap quickly even if you are new to it. If you ever run into an issue you will usually get help quickly and easily from the vast online community and web forums.  
  
In our next video we will discuss,**downloading, setting up and understanding different bootstrap components**.

### 2. Setting up bootstrap

In this video we will discuss **how to download, set-up and create our first bootstrap web page**.   
  
  
  
To get started with Bootstrap the first step is to download Bootstrap from[http://getbootstrap.com](http://getbootstrap.com/). This website also has all the documentation you need to get started with bootstrap.  
  
As of this recording the version is **3.3.6**. With the download you get a single zip folder which contains all the required bootstrap components.  
  
Unzip the ZIP folder and you should see the following folder structure. Notice there are 3 sub-folders (css, fonts & js). Let us understand the use of each file, folder by folder.   
  
   
  
**Files in "css" folder**  
  
**bootstrap.css** -  This is the core css for BootStrap that defines all the style for various controls and components  
  
**bootstrap.css.map** - When debugging the minified code, the line numbers do not refer to the orignal files. The file that has the .map extension which is also called as source map file fixes this problem by allowing the web debuggers to refer to the original context from where the code was generated. This file is useful during development.   
  
**bootstrap.min.css** - This is the compressed version meaning all the whitespaces, line breaks and any other extra characters have been removed. As a result the size of the minified file is smaller than the non-minified file. Minified version is usually used on a production server for efficient download where as the non minified version is used in development environment as it is more readable and easy to debug if there are issues.  
  
**bootstrap.min.css.map** - Source map file for bootstrap.min.css  
  
**bootstrap-theme.css** - As the name suggests this is the theme for bootstrap. Adding the core bootstrap.css is enough for bootstrap to work. The theme file is optional and is usually used for a visually enhanced experience. For example if you want 3D effects, gradients, shadows etc.  
  
**bootstrap-theme.css.map** - Source map file for bootstrap-theme.css  
  
**bootstrap-theme.min.css** - Minified version of bootstrap-theme.css  
  
**bootstrap-theme.min.css.map** - Source map file for bootstrap-theme.min.css  
  
**Files in "fonts" folder**  
There are 5 different font files from Glyphicons. These 5 different files are just different format of the Glyphicons font, to support different browsers.  
  
**Files in "js" folder :** These JavaScript files are optional. These are required if you want to use bootstrap widgets like picture carousel, dropdown menus, collapsible accordian etc. One important thing to keep in mind is that boostrap JavaScript has a dependency on jQuery, so a reference to jQuery must also exist on the page where you want to use Bootstrap.  
  
**bootstrap.js** - This is the non-minified readable version that is usually used during development.  
  
**bootstrap.min.js** - Minified version of bootstrap.js optimised for faster download. This is the version that is usually used in a production environment.  
  
**npm.js** - npm is a file from Node.js and is used for npm installing bootstrap. If you are new to Node.js, don't worry, this is not going to come in the way to understand bootstrap.  
  
For this course I am going to use Visual Studio 2013 as the editor. You can use any editor of your choice.  
  
**Here are the steps to create your first web page with Bootstrap**  
  
1. Create a new empty ASP.NET web application project. Name it BootstrapDemo  
2. To use bootstrap in your website, copy the folder that contains 3 sub-folders (css, fonts, & js) in your website project folder.  
3. Add a new HTML file to the project. Name it index.html.  
3. There is a basic template available at the following link. Copy and paste the template code in index.html  
<http://getbootstrap.com/getting-started/#template>  
4. Finally modify the code in index.html as shown below, to make sure bootstrap is working as expected. 

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Bootstrap Example</title>

    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

    <h1 class="text-primary">Hello, Bootstrap!</h1>

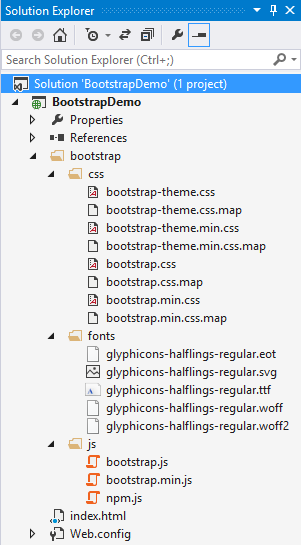
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">

    </script>

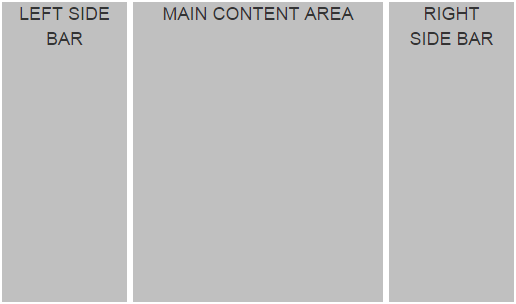
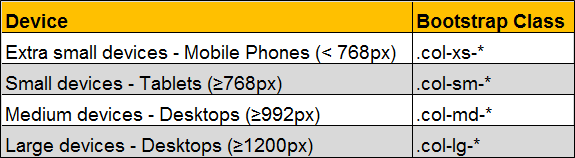
    <script src="bootstrap/js/bootstrap.min.js"></script>

</body>

</html>

**Please note :**The viewport meta tag ensure proper rendering and touch zooming on a mobile device.  
  
5. At this point, the project structure in solution explorer should be as shown below.   
  


### 3. Bootstrap Grid System

One of the very important concept that we need to understand in Bootstrap is it's **Grid System**. The Grid system is used for creating page layouts through a series of rows and columns. The Grid System consists of 12 columns. This grid system is so flexible that you can create any page layout that you want.   
  
bootstrap grid system example  
  
Let us now see how to create a 3 column layout using the Bootstrap Grid System. The 3 column layout should be as shown below.   
  
   
  
The page should be divided into 3 columns with 1:4:1 ratio. One important thing to keep in mind here is that, if we add the 3 numbers in the ratio (1:4:1), the sum should be equal to 12, because the bootstrap grid system consists of 12 columns. This means if the LEFT SIDE BAR column is 2 units, the MAIN CONTENT AREA column should be 8 units and the RIGHT SIDE BAR column should be 2 units. So when we add these 3 numbers (2 Units + 8 Units + 2 Units), we get 12 units.   
  
Bootstrap includes several grid classes for creating layouts for different devices like mobile phones, tablets, laptops, desktops, etc. The table below shows the available bootstrap grid classes that you can use to create grid column layouts ranging from extra small devices like mobile phones to large devices like large desktop screens.   
  
  
  
We will discuss these different classes in detail with examples in a later video. For now let's use **.col-md-\*** class to create a 3 column layout.    
  
There are 3 simple steps to create the 3 column layout that we want, using the bootstrap 12 column grid system  
1. First, create a container. We do this by creating a <div> element using the bootstrap .container class. All the rows will be within this container.  
2. Second, create a row within the container. We do this by creating a <div> element using the bootstrap .row class.  
3. Finally within the row we create the 3 columns that we need. We do this by creating <div> elements using any of the bootstrap grid classes (.col-xs-\*, .col-sm-\*, .col-md-\* & .col-lg-\*). In this example we are using col-xs-\* class.  
  
Please note that here I am also using a custom class (.customDiv) to apply some custom styles to each of the div elements.

<div class="container">

    <div class="row">

        <div class="col-md-2">

            <div class="customDiv">LEFT SIDE BAR</div>

        </div>

        <div class="col-md-8">

            <div class="customDiv">MAIN CONTENT AREA</div>

        </div>

        <div class="col-md-2">

            <div class="customDiv">RIGHT SIDE BAR</div>

        </div>

    </div>

</div>

Here is the definition of the custom css class (**.customDiv**)

.customDiv{

    margin:3px;

    min-height:300px;

    background-color:silver;

    text-align:center;

    font-size:large

}

I have created this class in a stylesheet that I named CustomStyles.css. I have placed this in the root folder of the project.  
  
Here is the complete HTML of index.html 

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <title>Bootstrap Example</title>

    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <link href="CustomStyles.css" rel="stylesheet" />

</head>

<body>

    <div class="container">

        <div class="row">

            <div class="col-md-2">

                <div class="customDiv">LEFT SIDE BAR</div>

            </div>

            <div class="col-md-8">

                <div class="customDiv">MAIN CONTENT AREA</div>

            </div>

            <div class="col-md-2">

                <div class="customDiv">RIGHT SIDE BAR</div>

            </div>

        </div>

    </div>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js">

    </script>

    <script src="bootstrap/js/bootstrap.min.js"></script>

</body>

</html>

Some important points to keep in mind when working with the bootstrap grid system to create page layouts  
1. Rows must be placed within a container for proper alignment and padding.  
2. Use rows to create horizontal groups of columns.  
3. Content should be placed within columns, and only columns may be immediate children of rows.  
4. If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line.  
5. Grid columns are created by specifying the number of twelve available columns you wish to span. For example, to create four equal columns we would use four .col-xs-3.  
  
To create 4 equal columns i.e with 1:1:1:1 ratio, we would use the following HTML

<div class="container">

    <div class="row">

        <div class="col-md-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 3</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

To create 4 columns with 1:2:2:1 ratio, we would use the following HTML

<div class="container">

    <div class="row">

        <div class="col-md-2">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-4">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-4">

            <div class="customDiv">Column 3</div>

        </div>

        <div class="col-md-2">

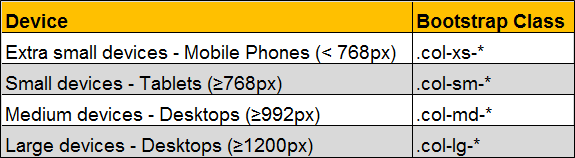
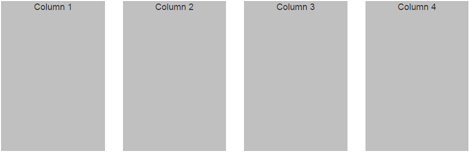
            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

### 4. Bootstrap 3 grid classes

**Here are the different grid classes that are available in Bootstrap 3**  
  
  
Grid classes for a given screen size apply to that screen size and larger unless another declaration overrides it. Let us understand what we mean by this statement with an example.   
  
For example, we want four equal columns on both medium and large devices.    
   
  
For medium devices the class that we use is col-md-\*. Since we want 4 equal columns, we would use four col-md-4 columns as shown below. So with the following HTML we get four equal columns on both medium and large devices. This is because col-md-\* class is applied to both medum and large deives as we have not used col-lg-\*.

<div class="container">

    <div class="row">

        <div class="col-md-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 3</div>

        </div>

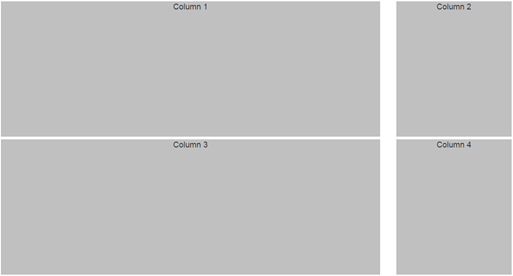
        <div class="col-md-3">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

If you want, you can change this on a large device by using col-lg-\* class along with col-md-\* class. Let us say on a large deive we want just 2 columns in every row with 3:1 ratio.  
   
  
To achieve this we would modify our HTML as shown below. So with these 2 classes in places we get 4 equal columns on a medium device and 2 columns with 3:1 ratio on a large device.

<div class="container">

    <div class="row">

        <div class="col-md-3 col-lg-9">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3 col-lg-3">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-3 col-lg-9">

            <div class="customDiv">Column 3</div>

        </div>

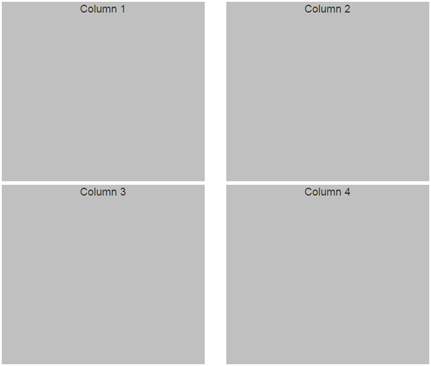
        <div class="col-md-3 col-lg-3">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

However, on a small device notice the columns are stacked on top of each other with each column spanning across the 12 column grid system. If you want you can change this default behaviour using col-sm-\* class. Let us say on a small device we want 2 equal columns in every row.   
   
  
To achieve this we would modify our HTML as shown below. So with these 3 classes in places we get 4 equal columns on a medium device, 2 columns with 3:1 ratio on a large device and 2 equal columns on a small device.

<div class="container">

    <div class="row">

        <div class="col-md-3 col-lg-9 col-sm-6">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3 col-lg-3 col-sm-6">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-3 col-lg-9 col-sm-6">

            <div class="customDiv">Column 3</div>

        </div>

        <div class="col-md-3 col-lg-3 col-sm-6">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

On an extra small device, the columns are stacked on top of each other with each column spanning across the 12 column grid system. If you want you can change this default behaviour very easily using col-xs-\* class.   
  
Now, if you remove all the grid classes except col-sm-6. In other words if we modify the code as shown below, we get 2 eqaul columns on small, medium and large devices. This is because grid classes for a given screen size apply to that screen size and larger unless another declaration overrides it.

<div class="container">

    <div class="row">

        <div class="col-sm-6">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-sm-6">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-sm-6">

            <div class="customDiv">Column 3</div>

        </div>

        <div class="col-sm-6">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

### By now I believe you have understood the use of these grid classes. With these classes, you have complete control over the layout of your website on different screen sizes.

### 5. Bootstrap grid column offset

When creating website layouts using the grid system, you may want to move grid columns to the right for alignment purpose. Let us understand this with an example.   
  
Let us say we want to create a layout as shown below.   
   
  
This can be very easily achieved with the following HTML

<div class="container">

    <div class="row">

        <div class="col-md-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 2</div>

        </div>

    </div>

    <div class="row">

        <div class="col-md-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 3</div>

        </div>

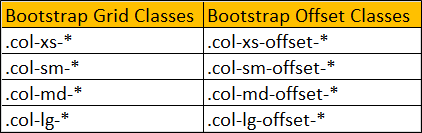
        <div class="col-md-3">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

Now, let us say we want the 2 columns in the first row to be centered. To center the columns, we need to push the first column to the right by 3 columns.   
  
   
  
This is exactly the purpose of bootstrap offset classes. Just like how we have different grid classes for different screen sizes, we have the corresponding offset classes as well.   
  
   
  
The following HTML will center the 2 columns in the first row. This is achieved by using col-md-offset-3 which is going to push the first column in the first row by 3 columns.

<div class="container">

    <div class="row">

        <div class="col-md-3 col-md-offset-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 2</div>

        </div>

    </div>

    <div class="row">

        <div class="col-md-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 3</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

Now, we want 6 columns gap between the first and second columns in the first row. In other words the layout should be as shown below.    
  
   
  
So we want the second column in the first row to be pushed 6 columns to the right. This can be very easily achieved by using col-md-offset-6 class on the second column in the first row.

<div class="container">

    <div class="row">

        <div class="col-md-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3 col-md-offset-6">

            <div class="customDiv">Column 2</div>

        </div>

    </div>

    <div class="row">

        <div class="col-md-3">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-3">

            <div class="customDiv">Column 3</div>

        </div>

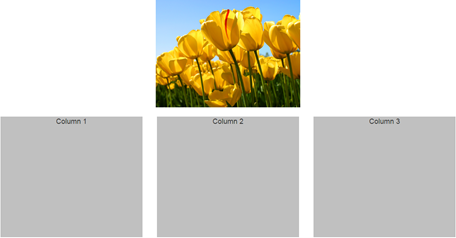
        <div class="col-md-3">

            <div class="customDiv">Column 4</div>

        </div>

    </div>

</div>

Let us now look at a simple example of where this grid column offset feature could be useful. Let us say we want to display 3 equal columns in the second row and only one image in the first row and the image should be centered.   
  
   
  
This can be very easily achieved with the following HTML.

<div class="container">

    <div class="row">

        <img src="Images/Tulips.jpg" class="col-md-4 col-md-offset-4" />

    </div>

    <br />

    <div class="row">

        <div class="col-md-4">

            <div class="customDiv">Column 1</div>

        </div>

        <div class="col-md-4">

            <div class="customDiv">Column 2</div>

        </div>

        <div class="col-md-4">

            <div class="customDiv">Column 3</div>

        </div>

    </div>

</div>

If you view the page in the browser at this point, you will see 2 problems  
1. The width of the image is slightly larger than the columns in the second row.  
2. As you resize the browser, and when the screen size is of a small device or extra small device, the image is not responsive.  
  
Let us see how to solve these problems one by one.  
  
1. To make the width of the image same as the width of the columns in the second row, remove margin style from CustomStyles.css and include margin-bottom style. With this change .customDiv class will be as shown below

.customDiv{

    margin-bottom:3px;

    min-height:300px;

    background-color:silver;

    text-align:center;

    font-size:large;

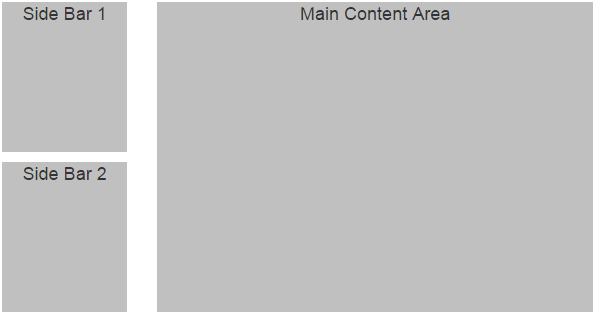
}

2. To make the image responsive, add img-responsive class on the img element

<img src="Images/Tulips.jpg" class="col-md-4 col-md-offset-4 img-responsive" />

### The following bootstrap classes are useful to style an img element .img-circle .img-thumbnail .img-rounded Please note : For some reason img-rounded class is not working as expected when used along with any of the grid or offset classes. Without these classes we get the rounded corners as expected. If you know the fix for this issue, please leave it as a comment so it could help us.

### 6. Bootstrap nested rows and columns

Bootstrap supports nested rows and columns. This means rows and columns can be placed inside an existing column. The formula for the the nested row is also the same, i.e the columns in the nested row should add up to 12 or fewer. However, please note that it is not required that you use all 12 available columns.  
  
Let's understand nested rows and columns with an example. Let us say, we want to create a layout as shown below.   
  
  
  
This can be very easily achieved by nesting rows and columns. Notice in the first column in the first row, we have 2 nested rows.

<div class="container">

    <div class="row">

        <div class="col-xs-3">

            <!--Nested row-->

            <div class="row">

                <div class="col-xs-12">

                    <div class="sidebarContent">Side Bar 1</div>

                </div>

            </div>

            <!--Nested rows-->

            <div class="row">

                <div class="col-xs-12">

                    <div class="sidebarContent">Side Bar 2</div>

                </div>

            </div>

        </div>

        <div class="col-xs-9">

            <div class="mainContent">Main Content Area</div>

        </div>

    </div>

</div>

Please not that we have applied some custom styles to the div elements to make the output pretty. The custom styles are in CustomStyles.css.

.sidebarContent{

    background-color:silver;

    text-align:center;

    font-size:large;

    min-height:150px;

    margin-bottom:10px;

}

.mainContent{

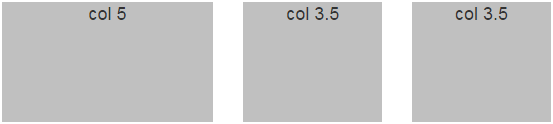
    background-color:silver;

    text-align:center;

    font-size:large;

    min-height:310px;

}

Let us look at another example, where nesting rows and columns is useful. Let us say we want 3 columns and the columns should be split as shown below.  
col 5 | col 3.5 | col 3.5   
  
   
  
The easiest way to achieve this is by nesting rows and columns

<div class="container">

    <div class="row">

        <div class="col-xs-5">

            <div class="customDiv">col 5</div>

        </div>

        <div class="col-xs-7">

            <div class="row">

                <div class="col-xs-6">

                    <div class="customDiv">col 3.5</div>

                </div>

                <div class="col-xs-6">

                    <div class="customDiv">col 3.5</div>

                </div>

            </div>

        </div>

    </div>

</div>

The following custom style is used with this example.

.customDiv {

    background-color: silver;

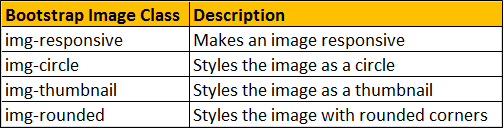
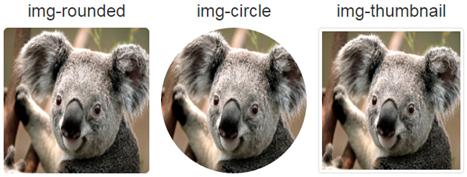
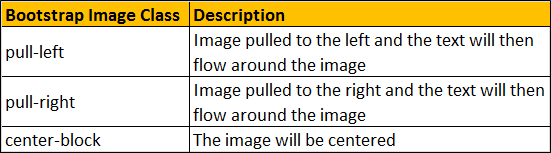
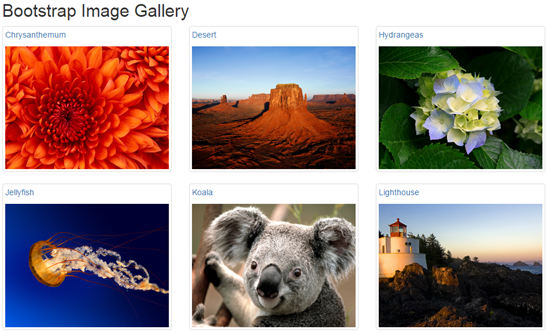
    text-align: center;

    font-size: large;

    min-height: 150px;

}

### 7. Bootstrap image gallery

In [Part 5](http://csharp-video-tutorials.blogspot.com/2016/05/bootstrap-grid-column-offset.html) of [Bootstrap tutorial for beginners](https://www.youtube.com/playlist?list=PL6n9fhu94yhXd4xnk-j5FGhHjUv1LsF0V), we have already discussed 4 bootsrap classes that are extremely useful to style images.   
  
  
  
  
**Bootstrap classes to position an image**   
  
  
**A simple bootstrap image gallery**   
  
  
The image gallery should  
1. Display 4 thumbnails on a large screen size  
2. Display 3 thumbnails on a medium screen size  
3. Display 2 thumbnails on a small screen size  
4. Display 1 thumbnail on an extra small screen size  
5. When you click on a thumbnail, the original image should be displayed  
  
Here is the complete HTML of the image gallery

<div class="container">

    <div class="row">

        <div class="col-lg-12">

            <h2>Bootstrap Image Gallery</h2>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Chrysanthemum.jpg" class="thumbnail">

                <p>Chrysanthemum</p>

                <img src="Images/Chrysanthemum.jpg" />

            </a>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Desert.jpg" class="thumbnail">

                <p>Desert</p>

                <img src="Images/Desert.jpg" />

            </a>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Hydrangeas.jpg" class="thumbnail">

                <p>Hydrangeas</p>

                <img src="Images/Hydrangeas.jpg" />

            </a>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Jellyfish.jpg" class="thumbnail">

                <p>Jellyfish</p>

                <img src="Images/Jellyfish.jpg" />

            </a>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Koala.jpg" class="thumbnail">

                <p>Koala</p>

                <img src="Images/Koala.jpg" />

            </a>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Lighthouse.jpg" class="thumbnail">

                <p>Lighthouse</p>

                <img src="Images/Lighthouse.jpg" />

            </a>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Penguins.jpg" class="thumbnail">

                <p>Penguins</p>

                <img src="Images/Penguins.jpg" />

            </a>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6">

            <a href="Images/Tulips.jpg" class="thumbnail">

                <p>Tulips</p>

                <img src="Images/Tulips.jpg" />

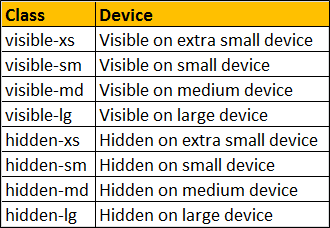
            </a>

        </div>

    </div>

</div>

### 8. Bootstrap 3 responsive utility classes

**What are bootstrap responsive utility classes**  
Bootstrap responsive utility classes are useful for showing and hiding content by device via media query.    
  
The following are some of the responsive utility classes available in bootstrap 3  
  
  
Let us now understand the use of these classes with an example. Let us say we have the following 4 columns on a web page.    
twitter bootstrap utility classes  
  
The following is the HTML I used to produce these 4 columns

<div class="container">

    <div class="row">

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">ALL Screens</div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">Medium, Large and Small</div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">Medium and Large </div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">Large</div>

        </div>

    </div>

</div>

Here is the CSS for the **customDiv** class

.customDiv{

    margin:10px;

    min-height:50px;

    background-color:silver;

    text-align:center;

    font-size:large;

    padding:10px;

}

**Here is what we want to do**  
1. The column that displays "ALL Screens" must be visible on all screen sizes  
2. The column that displays "Medium, Large and Small" must be visible only on medium, large and small screen sizes. It should be hidden on an extra small screen size.  
3. The column that displays "Medium and Large" must be visible only on medium and large screen sizes. It should be hidden on small and extra small screen sizes.  
4. The column that displays "Large" must be visible only on a large screen size. It should be hidden on all other screen sizes.  
  
**This can be very easily achieved by using the responsive utility classes**  
1. "ALL Screens" column must be visible on all screen sizes, so no change is required here.  
  
2. "Medium, Large and Small" column must be visible only on medium, large and small screen sizes. It should be hidden on an extra small screen size. So applying "hidden-xs" class on this column will hide this column on an extra small device but will be visible across all other devices.

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 hidden-xs">

    <div class="customDiv">Medium, Large and Small</div>

</div>

We can also achieve exactly the same thing by using visible-lg, visible-md, and visible-sm classes instead of hidden-xs class. Since with "hidden-xs" we only need to use one class, where as with visible classes we have to use 3 of them, so I have chosen to use hidden-xs.

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 visible-lg visible-md visible-sm">

    <div class="customDiv">Medium, Large and Small</div>

</div>

3. "Medium and Large" column must be visible only on medium and large screen sizes. It should be hidden on small and extra small screen sizes. To achieve this apply visible-lg and visible-md classes on this column.

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 visible-lg visible-md">

    <div class="customDiv">Medium and Large </div>

</div>

We can also achieve exactly the same thing by using hidden-sm and hidden-xs classes instead of visible-lg and visible-md classes.

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 hidden-sm hidden-xs">

    <div class="customDiv">Medium and Large </div>

</div>

4. "Large" column must be visible only on large screen sizes. It should be hidden on all other screen sizes. To achieve this apply visible-lg class on this column.

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 visible-lg">

    <div class="customDiv">Large</div>

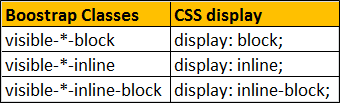
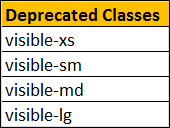
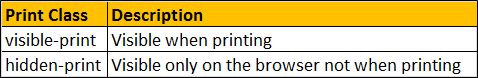
</div>

As you might have already guessed by now, you can also achieve exactly the same thing by using the hidden classes (hidden-xs hidden-sm and hidden-md), but with this approach you have to use 3 classes, where as with visible classes we have to use only one class (visible-lg)

<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12 hidden-xs hidden-sm hidden-md">

    <div class="customDiv">Large</div>

</div>

As of bootstrap version 3.2.0, the .visible-\*-\* classes for each screen size come in three variations, one for each CSS display property value shown below.   
  
  
The following article explains the difference between display: inline, display: inline-block and display: block  
<http://stackoverflow.com/questions/8969381/what-is-the-difference-between-display-inline-and-display-inline-block>  
  
The following classes also exist, but are deprecated as of v3.2.0.    
  
  
Similar to the regular responsive classes, you can use the following utility classes to show or hide certain elements for printing purpose   
  
  
**Example :** The column that has "hidden-print" class is visible only on the browser and not when printing. The column that has "visible-print" class is visible only during printing and not on the browser.

<div class="container">

    <div class="row">

        <div class="col-xs-12 hidden-print">

            <div class="customDiv">Visible on browser NOT when printing</div>

        </div>

        <div class="col-xs-12 visible-print">

            <div class="customDiv">Visible when printing</div>

        </div>

    </div>

</div>

### As of bootstrap version 3.2.0, the visible-print class come in three variations, one for each CSS display property value shown below. bootstrap 3 visible-print

### The visible-print class also exists but is deprecated as of bootstrap version 3.2.0.

### 9. Bootstrap typography

Let's start with HTML heading elements - <h1> through <h6>

<h1>Heading 1</h1>

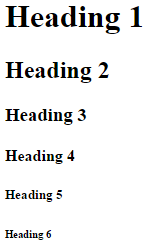
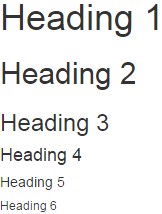
<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

The headings from h1 through h6 will be as shown below without bootstrap   
  
  
With bootstrap the headings are as shown below.   
  
  
**Bootstrap also has classes from h1 through h6.** These classes are useful when you want to match the font styling of a heading but still want your text to be displayed inline. Let us understand what we mean by this with an example.

<h4>Bootstrap</h4> is a free, open-source and is the most popular

HTML, CSS, and JavaScript framework developed by twitter for creating responsive

web applications. It includes HTML and CSS based design templates for common user

interface components like Buttons, Dropdowns, Typography, Tabs, Forms, Tables,

Navigations, Alerts, Modals, Accordion, Carousel etc. along with optional JavaScript

extensions. Bootstrap framework is based on open standards - HTML, CSS and

JavaScript. This means bootstrap can be used with any server side technology and a

ny platform. You can use it with any web application built with any server side

technology like ASP.NET, JAVA, PHP etc.

In the above html, we are using <h4> element with the word Bootstrap. Since <h4> is a block element it will render the word "Bootstrap" as if there is a line break before and after it as shown below.   
  
  
If you want the word "Bootstrap" to match the font style of <h4>, but still want the word to be displayed inline with other text then use h4 bootstrap class.

<span class="h4">Bootstrap</span> is a free, open-source and is the most popular

HTML, CSS, and JavaScript framework developed by twitter for creating responsive

web applications. It includes HTML and CSS based design templates for common

user interface components like Buttons, Dropdowns, Typography, Tabs, Forms,

Tables, Navigations, Alerts, Modals, Accordion, Carousel etc. along with

optional JavaScript extensions. Bootstrap framework is based on open standards

- HTML, CSS and JavaScript. This means bootstrap can be used with any server

side technology and any platform. You can use it with any web application built

with any server side technology like ASP.NET, JAVA, PHP etc.

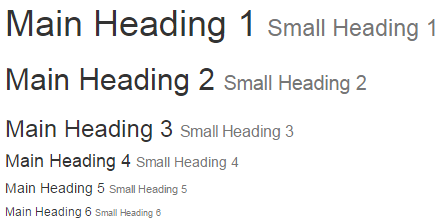
With h4 class, the word "Bootstrap" will be displayed inline with the other text as shown below.   
  
  
Another example where we could use .h1 through .h6 classes. Let us say we want to display all the headings from h1 to h6 in a single row as shown below.   
bootstrap header classes  
  
If we use the html elements <h1> through <h6> as shown below

<h1>H1</h1> <h2>H2</h2> <h3>H3</h3> <h4>H4</h4> <h5>H5</h5> <h6>H6</h6>

The headers will not be in the same line, as all the headers are block elements.   
html h1 to h6 tags  
  
Instead use .h1 through .h6 classes as shown below

<span class="h1">H1</span><span class="h2">H2</span><span class="h3">H3</span>

<span class="h4">H4</span><span class="h5">H5</span><span class="h6">H6</span>

If you want to create a lighter secondary text along with the main text in any heading, you can use either <small> tag or the .small class.   
  
  
**Creating secondary text in a heading using the <small> tag**

<h1>Main Heading 1 <small>Small Heading 1</small></h1>

<h2>Main Heading 2 <small>Small Heading 2</small></h2>

<h3>Main Heading 3 <small>Small Heading 3</small></h3>

<h4>Main Heading 4 <small>Small Heading 4</small></h4>

<h5>Main Heading 5 <small>Small Heading 5</small></h5>

<h6>Main Heading 6 <small>Small Heading 6</small></h6>

**Creating secondary text in a heading using the .small class**

<h1>Main Heading 1 <span class="small">Small Heading 1</span></h1>

<h2>Main Heading 2 <span class="small">Small Heading 2</span></h2>

<h3>Main Heading 3 <span class="small">Small Heading 3</span></h3>

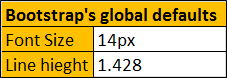
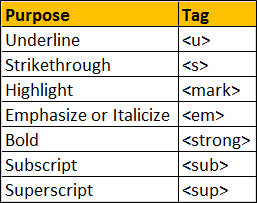
<h4>Main Heading 4 <span class="small">Small Heading 4</span></h4>

<h5>Main Heading 5 <span class="small">Small Heading 5</span></h5>

<h6>Main Heading 6 <span class="small">Small Heading 6</span></h6>

### In our next video we will discuss working with paragraph elements.

### 10. Bootstrap paragraphs

The following are the global defaults applied to the <body> and all paragraph <p> elements. In addition, paragraphs receive a bottom margin of half their computed line-height (10px by default).   
  
  
**Bootstrap inline text elements**   
  
  
**Example :**

<p>

    <u>This text will be underlined</u>

</p>

<p>

    <s>This text will be striken through</s>

</p>

<p>

    <mark>This text will be highlighted</mark>

</p>

<p>

    <em>This text will be italicized</em>

</p>

<p>

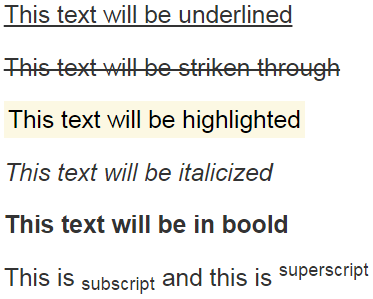
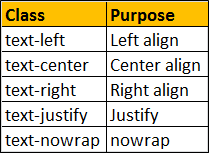
    <strong>This text will be in boold</strong>

</p>

<p>

    This is <sub>subscript</sub> and this is <sup>superscript</sup>

</p>

**Output :**   
  
  
**Bootstrap text alignment classes**   
  
  
**Example :**

<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">

    The text in this paragraph will be justified.

    Bootstrap is a free, open-source and is the most popular HTML,

    CSS, and JavaScript framework.

</p>

<p class="text-nowrap">

    This text will not wrap and you will get a horizontal scrollbar

    if there is no space on the screen to view.

</p>

### 11. Bootstrap blockquotes and lists

**Blockquotes are useful for quoting blocks of content from another source within your web page**  
1. <blockquote> element can be used with any HTML that you want as the quote. For simple text quotes bootstrap recomends using a <p> element.  
2. For identifying the source of the quote, use the <footer> element. Wrap the name of the source work using <cite> element. On hover the title attribute value will be displayed as a tooltip.  
3. To right-align the blockquote content, use .blockquote-reverse class on the <blockquote> element   
  
**Blockquotes example :**

<blockquote class="blockquote-reverse">

    <p>By failing to prepare, you are preparing to fail</p>

    <footer>

        This famous quote is by

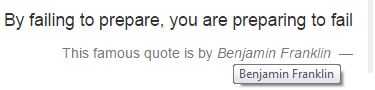
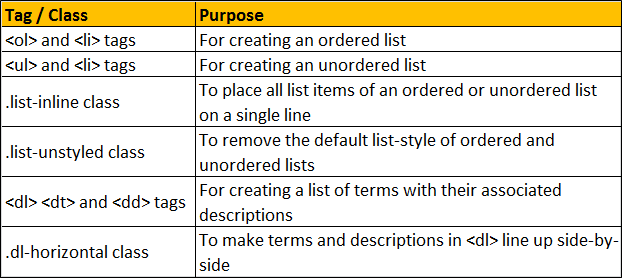
        <cite title="Benjamin Franklin">

            Benjamin Franklin

        </cite>

    </footer>

</blockquote>

**Output :**   
   
  
**Bootstrap List tags and classes**   
  
  
**Bootstrap Ordered List :** For an Ordered List use the <ol> element

<ol>

    <li>India</li>

    <li>USA</li>

    <li>UK</li>

    <li>Australia</li>

    <li>Canada</li>

</ol>

**Output :**   
   
  
**Bootstrap Unordered List :** For an Unordered List use the <ul> element

<ul>

    <li>India</li>

    <li>USA</li>

    <li>UK</li>

    <li>Australia</li>

    <li>Canada</li>

</ul>

**Output :**    
   
  
To place all list items of ordered or unordered list on a single line use the **.list-inline** class

<ul class="list-inline">

    <li>India</li>

    <li>USA</li>

    <li>UK</li>

    <li>Australia</li>

    <li>Canada</li>

</ul>

**Output :**   
bootstrap list items inline   
  
To remove the default list-style of ordered and unordered lists use the **.list-unstyled** class

<ul class="list-unstyled">

    <li>

        India

        <ul class="list-unstyled">

            <li>Andhra Pradesh</li>

            <li>Tamil Nadu</li>

            <li>Kerala</li>

        </ul>

    </li>

    <li>

        USA

        <ul>

            <li>Alabama</li>

            <li>Alaska</li>

            <li>Iowa</li>

        </ul>

    </li>

    <li>UK</li>

    <li>Australia</li>

    <li>Canada</li>

</ul>

**Output :**   
   
  
For creating a list of terms with their associated descriptions use <dl> <dt> and <dd> tags

<dl>

    <dt>ASP.NET</dt>

    <dd>Framework for developing web applications</dd>

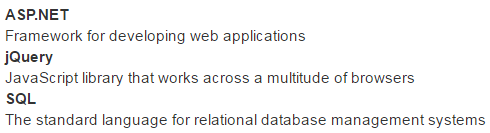
    <dt>jQuery</dt>

    <dd>JavaScript library that works across a multitude of browsers</dd>

    <dt>SQL</dt>

    <dd>The standard language for relational database management systems</dd>

</dl>

**Output :**   
   
  
To make terms and descriptions in <dl> line up side-by-side use **.dl-horizontal** class

<dl class="dl-horizontal">

    <dt>ASP.NET</dt>

    <dd>Framework for developing web applications</dd>

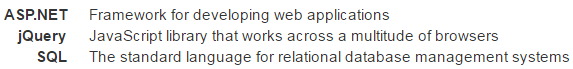
    <dt>jQuery</dt>

    <dd>JavaScript library that works across a multitude of browsers</dd>

    <dt>SQL</dt>

    <dd>The standard language for relational database management systems</dd>

</dl>

**Output :**   


### 12. Bootstrap list groups

**ootstrap list groups** are useful for displaying simple lists of elements, as well as complex ones with custom content.    
  
  
  
**Basic list group :** To create a basic list group, create an unordered list. Use .list-group class on the <ul> element and list-group-item class on the <li> element.

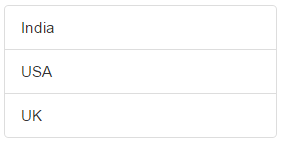
<ul class="list-group">

    <li class="list-group-item">India</li>

    <li class="list-group-item">USA</li>

    <li class="list-group-item">UK</li>

</ul>

**Output :**   
   
  
**List group item with a badge :** To create a badge, include a <span> element with .badge class inside the <li> element. 

<ul class="list-group">

    <li class="list-group-item">India</li>

    <li class="list-group-item">USA</li>

    <li class="list-group-item">UK</li>

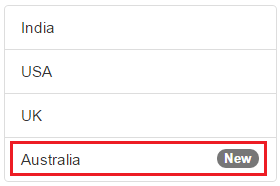
    <li class="list-group-item">

        Australia

        <span class="badge">New</span>

    </li>

</ul>

**Output :** Notice the new country **"Australia"** is displayed with the badge **"New"**   
   
  
**List group with hyperlinks :** To create list group with hyperlinks, use <div> instead of <ul> and <a> instead of <li>

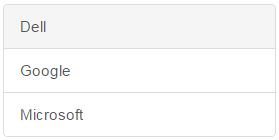
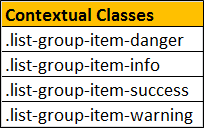
<div class="list-group">

    <a href="http://dell.com" class="list-group-item">Dell</a>

    <a href="http://google.com" class="list-group-item">Google</a>

    <a href="http://microsoft.com" class="list-group-item">Microsoft</a>

</div>

**Output :** On hover the cursor changes to a hand symbol and the background colour changes to grey. When you click on an item, you will be redirected to the respective URL.   
   
  
**Styling list items :** Use contextual classes to style the list items. The following are the contextual classes   
 

<ul class="list-group">

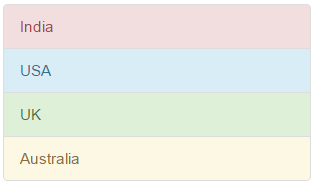
    <li class="list-group-item list-group-item-danger">India</li>

    <li class="list-group-item list-group-item-info">USA</li>

    <li class="list-group-item list-group-item-success">UK</li>

    <li class="list-group-item list-group-item-warning">Australia</li>

</ul>

   
  
**Highlight and disable list group items :** Use .active class to highlight a list-group-item and .disabled class to disable a list item.

<ul class="list-group">

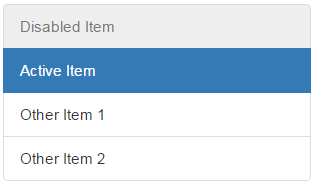
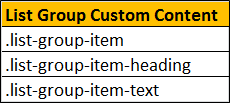
    <li class="list-group-item disabled">Disabled Item</li>

    <li class="list-group-item active">Active Item</li>

    <li class="list-group-item">Other Item 1</li>

    <li class="list-group-item">Other Item 2</li>

</ul>

**Output :**   
   
  
**List group custom content :** List groups can contain your own custom content. Nearly any HTML can be used. We will use the following 3 classes to create list group with custom   
content.   
 

<div class="list-group">

    <a href="#" class="list-group-item">

        <h4 class="list-group-item-heading">List Group Item 1 Heading</h4>

        <p class="list-group-item-text">List Group Item 1 Text</p>

    </a>

    <a href="#" class="list-group-item">

        <h4 class="list-group-item-heading">List Group Item 2 Heading</h4>

        <p class="list-group-item-text">List Group Item 2 Text</p>

    </a>

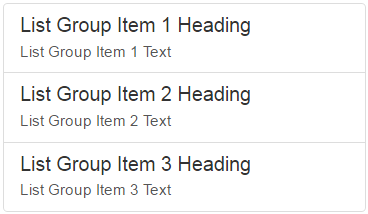
    <a href="#" class="list-group-item">

        <h4 class="list-group-item-heading">List Group Item 3 Heading</h4>

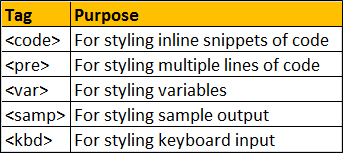
        <p class="list-group-item-text">List Group Item 3 Text</p>

    </a>

</div>



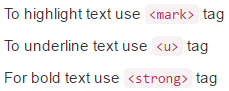
### 13. Bootstrap code blocks

**Bootstrap tags for styling code blocks**   
   
  
**Note :** Angle brackets must be escaped for proper rendering  
  
**For styling inline snippets of code use <code> tag**

<p>To highlight text use <code>&lt;mark&gt;</code> tag </p>

<p>To underline text use <code>&lt;u&gt;</code> tag </p>

<p>For bold text use <code>&lt;strong&gt;</code> tag </p>

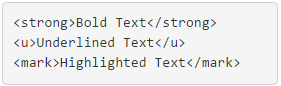
**Output :**   
   
  
**For styling multiple lines of code use <pre> tag**

<pre>&lt;strong&gt;Bold Text&lt;/strong&gt;

&lt;u&gt;Underlined Text&lt;/u&gt;

&lt;mark&gt;Highlighted Text&lt;/mark&gt;

</pre>

**Output :**   
   
  
**To set a max-height of 350px and provide a y-axis scrollbar use .pre-scrollableclass on <pre> tag**

<pre class="pre-scrollable">public class Program

{

    public static void Main()

    {

        Customer customer1 = new Customer()

        {

            ID = 101,

            Name = "Mark",

            Salary = 5000

        };

        Customer customer2 = new Customer()

        {

            ID = 102,

            Name = "Pam",

            Salary = 7000

        };

        Customer customer3 = new Customer()

        {

            ID = 104,

            Name = "Rob",

            Salary = 5500

        };

        Customer[] arrayCustomers = new Customer[2];

        arrayCustomers[0] = customer1;

        arrayCustomers[1] = customer2;

        List<customer> listCustomers = new List<customer>(2);

        listCustomers.Add(customer1);

        listCustomers.Add(customer2);

        listCustomers.Add(customer3);

        Customer cust = listCustomers[0];

        Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}",

                           cust.ID, cust.Name, cust.Salary);

        Console.WriteLine("--------------------------------");

        for (int i = 0; i < listCustomers.Count; i++)

        {

            Customer customer = listCustomers[i];

            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}",

                     customer.ID, customer.Name, customer.Salary);

        }

        Console.WriteLine("--------------------------------");

        foreach (Customer c in listCustomers)

        {

            Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}",

                                c.ID, c.Name, c.Salary);

        }

        Console.WriteLine("--------------------------------");

        listCustomers.Insert(1, customer3);

        Console.WriteLine("ID = {0}, Name = {1}, Salary = {2}",

               listCustomers[1].ID, listCustomers[1].Name,

               listCustomers[1].Salary);

        Console.WriteLine("--------------------------------");

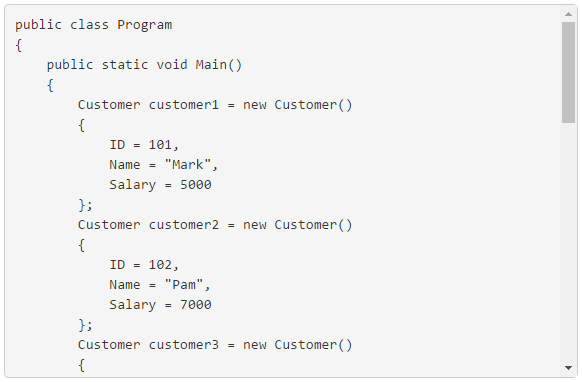
        Console.WriteLine("Index of Customer3 object in the List = "

                           + listCustomers.IndexOf(customer3));

        Console.WriteLine("--------------------------------");

    }

}</pre>

**Output :**    
  
  
**For styling variables use <var> tag**

var <var>fullName</var> = <var>firstName</var> + <var>lastName</var>;

**Output :**   
bootstrap styling variables   
  
**For styling sample output use <samp> tag**

<h4>Output</h4>

<samp>Hello world!</samp>

**Output :**   
bootstrap samp tag example   
  
**For styling keyboard input use <kbd> tag**

<h4>Visual Studio</h4>

<p>

    Keyboard shorcut for commenting code

    <kbd>CTRL</kbd> <kbd>K</kbd> and <kbd>CTRL</kbd> <kbd>C</kbd>

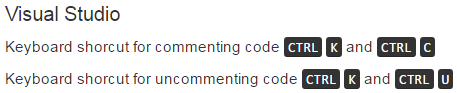
</p>

<p>

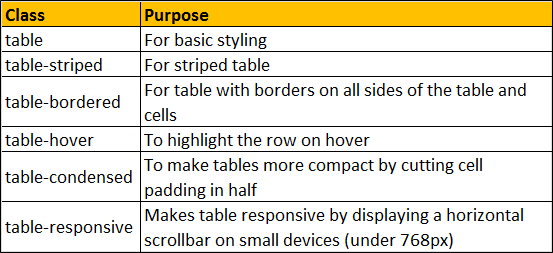
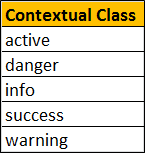
    Keyboard shorcut for uncommenting code

    <kbd>CTRL</kbd> <kbd>K</kbd> and <kbd>CTRL</kbd> <kbd>U</kbd>

</p>

**Output :**   


### 14. Bootstrap table classes

**Bootstrap classes for styling tables**   
   
  
**Contextual classes to color table rows or individual cells**   
   
  
Here is the HTML for the table without any boostrap table classes

<table>

    <thead>

        <tr>

            <th>Id</th>

            <th>Name</th>

            <th>Gender</th>

            <th>Email</th>

        </tr>

    </thead>

    <tbody>

        <tr>

            <td>1</td>

            <td>John</td>

            <td>Male</td>

            <td>John@emailDomain.com</td>

        </tr>

        <tr>

            <td>2</td>

            <td>Mary</td>

            <td>Femlae</td>

            <td>mary@emailDomain.com</td>

        </tr>

        <tr>

            <td>3</td>

            <td>Ben</td>

            <td>Male</td>

            <td>Ben@emailDomain.com</td>

        </tr>

        <tr>

            <td>4</td>

            <td>Sara</td>

            <td>Female</td>

            <td>sara@emailDomain.com</td>

        </tr>

        <tr>

            <td>5</td>

            <td>Sam</td>

            <td>Male</td>

            <td>sam@emailDomain.com</td>

        </tr>

    </tbody>

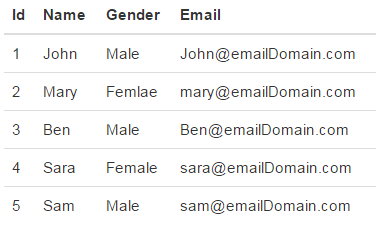
</table>

**Output :**   
   
  
**table class provides light padding and horizontal lines**

<table class="table">

    <!--rest of the HTML stays the same-->

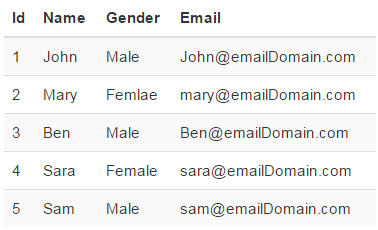
</table>

**Output :**   
   
  
**table-striped class provides zebra-striping for the table rows**

<table class="table table-striped">

    <!--rest of the HTML stays the same-->

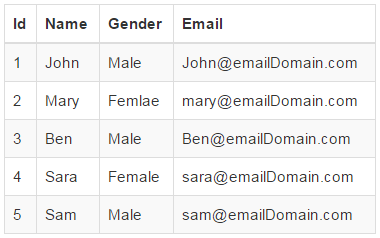
</table>

**Output :**   
   
  
**table-bordered class provide borders on all sides of the table and cells.**

<table class="table table-striped table-bordered">

    <!--rest of the HTML stays the same-->

</table>

**Output :**   
   
  
**table-hover class provides highlighting of rows on hover**

<table class="table table-bordered table-hover">

    <!--rest of the HTML stays the same-->

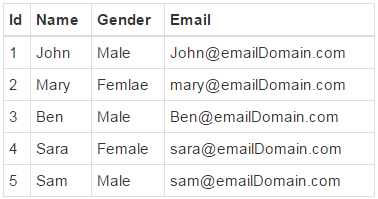
</table>

**table-condensed class makes table more compact by cutting cell padding in half**

<table class="table table-bordered table-hover table-condensed">

    <!--rest of the HTML stays the same-->

</table>

**Output :**   
   
  
**To make a table responsive**, place the table inside a <div> element, and apply table-responsive class on the <div> element. This will provide a horizontal scrollbar when the screen size is less than 768px (i.e on a small device). On a screen size larger than 768px you will not find any difference. Applying the table-responsive class directly on the table will not do anything useful.

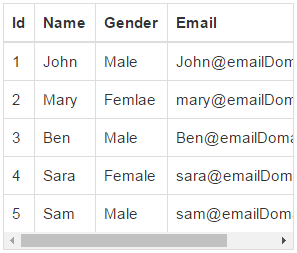
<div class="table-responsive">

    <table class="table table-bordered table-hover">

        <!--rest of the HTML stays the same-->

    </table>

</div>

**Output :**   
   
  
**Use the bootstrap contextual classes to colour the table rows**

<div class="table-responsive">

    <table class="table table-bordered table-hover">

        <thead>

            <tr>

                <th>Id</th>

                <th>Name</th>

                <th>Gender</th>

                <th>Email</th>

            </tr>

        </thead>

        <tbody>

            <tr class="active">

                <td>1</td>

                <td>John</td>

                <td>Male</td>

                <td>John@emailDomain.com</td>

            </tr>

            <tr class="danger">

                <td>2</td>

                <td>Mary</td>

                <td>Femlae</td>

                <td>mary@emailDomain.com</td>

            </tr>

            <tr class="info">

                <td>3</td>

                <td>Ben</td>

                <td>Male</td>

                <td>Ben@emailDomain.com</td>

            </tr>

            <tr class="success">

                <td>4</td>

                <td>Sara</td>

                <td>Female</td>

                <td>sara@emailDomain.com</td>

            </tr>

            <tr class="warning">

                <td>5</td>

                <td>Sam</td>

                <td>Male</td>

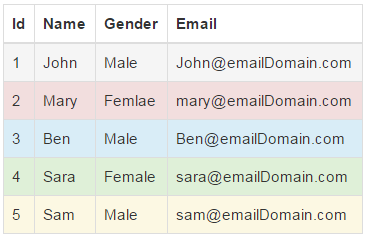
                <td>sam@emailDomain.com</td>

            </tr>

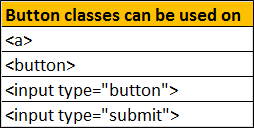
        </tbody>

    </table>

</div>

**Output :**   


### 15. Bootstrap button classes

   
  
Notice on the following elements we are not using any bootstrap classes.

<button>Button</button>

<input type="button" value="Input">

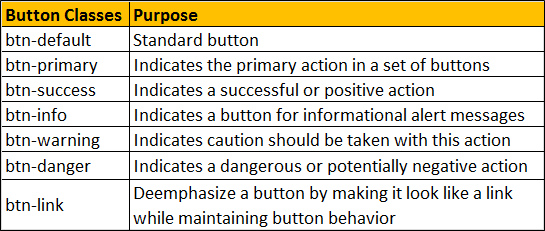
<input type="submit" value="Submit">

By default the above elements will be styled as shown below.   
Bootstrap button classes   
  
Adding **btn** class applies bootstrap styles to the button and input elements.

<button class="btn">Button</button>

<input class="btn" type="button" value="Input">

<input class="btn" type="submit" value="Submit">

**Output :**   
bootstrap button styles   
  
**Bootstrap button classes** : Along with the **btn** class, one of the following button classes can be used. The colour scheme you get depends on the class you use.   
   
  
**Example :** To get the buttons lined up one below the other instead of being side-by-side, I have placed them inside a <p> element.

<p><button class="btn btn-default">btn-default</button></p>

<p><button class="btn btn-primary">btn-primary</button></p>

<p><button class="btn btn-success">btn-success</button></p>

<p><button class="btn btn-info">btn-info</button></p>

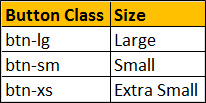
<p><button class="btn btn-warning">btn-warning</button></p>

<p><button class="btn btn-danger">btn-danger</button></p>

<p><button class="btn btn-link">btn-link</button></p>

**Output :**   
   
  
**To make an anchor element look like a button**, use **btn** class along with the other button classes (like btn-default, btn-primary etc.)

<a href="#" class="btn btn-primary">Hyperlink</a>

**Output :**   
bootstrap hyperlink button   
  
**Changing Button Size :** Use the following button classes to get large, small or extra small buttons   
 

<button class="btn btn-primary btn-lg">Large</button>

<button class="btn btn-primary btn-sm">Small</button>

<button class="btn btn-primary btn-xs">Extra Small</button>

**Output :**   
bootstrap 3 buttons size   
  
To create a button that span the full width of it's parent use **btn-block** class  
  
**Example :** The following example displays   
All the 4 buttons side by side in a single line on a large screen  
3 buttons side by side in the first line and the last button in the second line on a medium screen  
2 buttons side by side in the first line and the rest 2 buttons side by side in the second line on a small screen  
All the 4 buttons one below the other on an extra small screen  
  
What this example the button width is fixed

<div class="container">

    <div class="row">

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

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

            </div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

                <button class="btn btn-primary">Button 2</button>

            </div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

                <button class="btn btn-primary">Button 3</button>

            </div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

                <button class="btn btn-primary">Button 4</button>

            </div>

        </div>

    </div>

</div>

**Here is customDiv class**

.customDiv{

    padding:10px;

    width:100%;

}

We want the button width to span the full width of it's parent grid column. This can be very easily achieved by applying btn-blcok class on all the button elements. 

<div class="container">

    <div class="row">

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

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

            </div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

                <button class="btn btn-primary btn-block">Button 2</button>

            </div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

                <button class="btn btn-primary btn-block">Button 3</button>

            </div>

        </div>

        <div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">

            <div class="customDiv">

                <button class="btn btn-primary btn-block">Button 4</button>

            </div>

        </div>

    </div>

</div>

To disable a button set disabled attribute to disabled

<button class="btn btn-primary" disabled="disabled">Disabled Button</button>

**Output :** When you move the mouse over the disabled button, the cursor style changes to a stop sign, indicating that the button cannot be clicked.   
bootstrap disable button   
  
To disable an anchor element use disabled class

<a href="#" class="btn btn-primary disabled">Hyperlink</a>

bootstrap disable hyperlink

### 16. Bootstrap icons

If you want only icon on the button, without any text, then just remove the text

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

bootstrap button icon only   
  
Right now, the search icon colour is black which matches with the button font colour. Now if we apply btn and btn-primary classes on the button element, the search icon color is also automatically changed to match with the font colour.

<button class="btn btn-primary">

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

</button>

bootstrap glyphicons   
  
When the size of the button changes, so does the size of the icon. In the example below we are using btn-lg, btn-sm, and btn-xs for large, small and extra small buttons respectively.

<button class="btn btn-primary btn-lg">

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

</button>

<button class="btn btn-primary btn-sm">

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

</button>

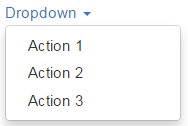
<button class="btn btn-primary btn-xs">

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

</button>

Notice as the size of the button changes, the size of the icon also changes automatically   
bootstrap glyphicons example

### 17. Bootstrap dropdown

Creating **bootstrap hyperlink dropdown**   
   
  
**Here is the HTML required**

<div class="dropdown">

    <a data-toggle="dropdown" class="dropdown-toggle">

        Dropdown

        <span class="caret"></span>

    </a>

    <ul class="dropdown-menu">

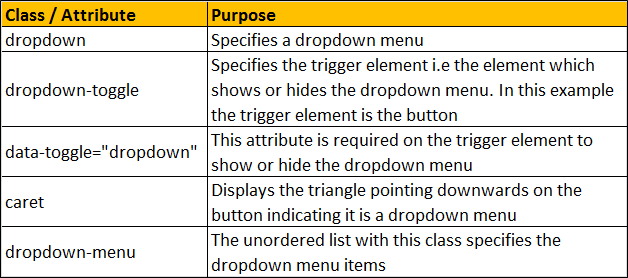
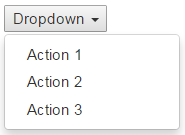
        <li><a href="#">Action 1</a></li>

        <li><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

    </ul>

</div>

  
  
**Creating bootstrap button dropdown**   
   
  
To create a dropdown with a button as the trigger element, replace the anchor element with a button element as shown below.

<div class="dropdown">

    <button data-toggle="dropdown" class="dropdown-toggle">

        Dropdown

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

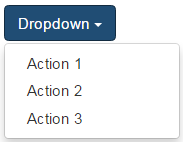
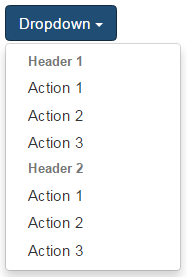
        <li><a href="#">Action 1</a></li>

        <li><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

    </ul>

</div>

Applying bootstrap button classes (btn & btn-primary) on the button element will style the dropdown button as shown below.   
   
  
**Dropdown menu header**   
   
  
To add a header to the dropdown menu, use dropdown-header class on an <li> element

<div class="dropdown">

    <button data-toggle="dropdown" class="btn btn-primary dropdown-toggle">

        Dropdown

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li class="dropdown-header"><strong>Header 1</strong></li>

        <li><a href="#">Action 1</a></li>

        <li><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

        <li class="dropdown-header"><strong>Header 2</strong></li>

        <li><a href="#">Action 1</a></li>

        <li><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

    </ul>

</div>

**Dropdown menu divider**   
   
  
To add a divider between the dropdown menu items, use divider class on an <li> element

<div class="dropdown">

    <button data-toggle="dropdown" class="btn btn-primary dropdown-toggle">

        Dropdown

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li class="dropdown-header"><strong>Header 1</strong></li>

        <li><a href="#">Action 1</a></li>

        <li><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

        <li class="divider"></li>

        <li class="dropdown-header"><strong>Header 2</strong></li>

        <li><a href="#">Action 1</a></li>

        <li><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

    </ul>

</div>

**Disabled dropdown menu item**   
   
  
To disable a menu item, use the disabled class

<div class="dropdown">

    <button data-toggle="dropdown" class="btn btn-primary dropdown-toggle">

        Dropdown

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li class="dropdown-header"><strong>Header 1</strong></li>

        <li><a href="#">Action 1</a></li>

        <li class="disabled"><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

        <li class="divider"></li>

        <li class="dropdown-header"><strong>Header 2</strong></li>

        <li><a href="#">Action 1</a></li>

        <li class="disabled"><a href="#">Action 2</a></li>

        <li><a href="#">Action 3</a></li>

    </ul>

</div>

If you want the dropdown menus to expand upwards, nest the dropdown markup in a <div> element with class dropup as shown below.   
   
  
Here is the HTML

<div class="dropup">

    <div class="dropdown">

        <button data-toggle="dropdown" class="btn btn-primary dropdown-toggle">

            Dropdown

            <span class="caret"></span>

        </button>

        <ul class="dropdown-menu">

            <li class="dropdown-header"><strong>Header 1</strong></li>

            <li><a href="#">Action 1</a></li>

            <li class="disabled"><a href="#">Action 2</a></li>

            <li><a href="#">Action 3</a></li>

            <li class="divider"></li>

            <li class="dropdown-header"><strong>Header 2</strong></li>

            <li><a href="#">Action 1</a></li>

            <li class="disabled"><a href="#">Action 2</a></li>

            <li><a href="#">Action 3</a></li>

        </ul>

    </div>

</div>

### 18. Bootstrap button group

To create a button group wrap the buttons in a <div> element with btn-group class

<div class="btn-group">

    <button class="btn btn-danger">Button 1</button>

    <button class="btn btn-default">Button 2</button>

    <button class="btn btn-info">Button 3</button>

    <button class="btn btn-primary">Button 4</button>

    <button class="btn btn-success">Button 5</button>

    <button class="btn btn-warning">Button 6</button>

</div>

Bootstrap button group   
  
**Button toolbar :** To create a button toolbar wrap button groups in a <div> element with btn-toolbar class

<div class="btn-toolbar">

    <div class="btn-group">

        <a href="#" class="btn btn-danger">Button 1</a>

        <a href="#" class="btn btn-default">Button 2</a>

    </div>

    <div class="btn-group">

        <a href="#" class="btn btn-info">Button 3</a>

        <a href="#" class="btn btn-primary">Button 4</a>

        <a href="#" class="btn btn-success">Button 5</a>

    </div>

    <div class="btn-group">

        <a href="#" class="btn btn-warning">Button 6</a>

        <a href="#" class="btn btn-primary">Button 7</a>

    </div>

</div>

bootstrap button toolbar  
  
**Button groups with different sizes :** To create button groups with different sizes (i.e large, small and extra small) use btn-group-lg, btn-group-sm and btn-group-xs classes respectively

<div class="btn-toolbar">

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

        <a href="#" class="btn btn-danger">Button 1</a>

        <a href="#" class="btn btn-default">Button 2</a>

    </div>

    <div class="btn-group btn-group-sm">

        <a href="#" class="btn btn-info">Button 3</a>

        <a href="#" class="btn btn-primary">Button 4</a>

        <a href="#" class="btn btn-success">Button 5</a>

    </div>

    <div class="btn-group btn-group-xs">

        <a href="#" class="btn btn-warning">Button 6</a>

        <a href="#" class="btn btn-primary">Button 7</a>

    </div>

</div>

bootstrap button group size  
  
**Vertical button group :** To create a vertical button group use btn-group-vertical class

<div class="btn-toolbar">

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

        <a href="#" class="btn btn-danger">Button 1</a>

        <a href="#" class="btn btn-default">Button 2</a>

    </div>

    <div class="btn-group-vertical btn-group-sm">

        <a href="#" class="btn btn-info">Button 3</a>

        <a href="#" class="btn btn-primary">Button 4</a>

        <a href="#" class="btn btn-success">Button 5</a>

    </div>

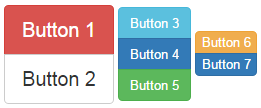
    <div class="btn-group-vertical btn-group-xs">

        <a href="#" class="btn btn-warning">Button 6</a>

        <a href="#" class="btn btn-primary">Button 7</a>

    </div>

</div>

   
  
**Nested button groups :** To create a button group with a dropdown menu, nest button groups

<div class="btn-group">

    <a href="#" class="btn btn-danger">Button 1</a>

    <a href="#" class="btn btn-default">Button 2</a>

    <div class="btn-group">

        <button data-toggle="dropdown" class="btn btn-primary dropdown-toggle">

            Dropdown

            <span class="caret"></span>

        </button>

        <ul class="dropdown-menu">

            <li><a href="#">Action 1</a></li>

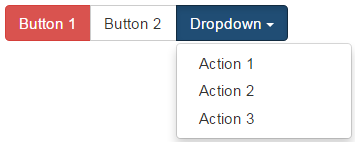
            <li><a href="#">Action 2</a></li>

            <li><a href="#">Action 3</a></li>

        </ul>

    </div>

</div>

   
  
**Justified hyperlink elements button group :** To create a justified hyperlink elements button group that span the entire width of it's parent use btn-group and btn-group-justifiedclasses.

<div class="btn-group btn-group-justified">

    <a href="#" class="btn btn-danger">Button 1</a>

    <a href="#" class="btn btn-default">Button 2</a>

    <a href="#" class="btn btn-info">Button 3</a>

    <a href="#" class="btn btn-primary">Button 4</a>

    <a href="#" class="btn btn-success">Button 5</a>

    <a href="#" class="btn btn-warning">Button 6</a>

</div>

bootstrap justified button group   
  
**Justified button elements** (i.e <button>, <input type="button"> and <inputtype="submit">) **button group** : To create a justified button group of button elements use btn-group and btn-group-justified classes. In addition you must also wrap each button in a <div> element with btn-group class.

<div class="btn-group btn-group-justified">

    <div class="btn-group">

        <button class="btn btn-danger">Button 1</button>

    </div>

    <div class="btn-group">

        <button class="btn btn-default">Button 2</button>

    </div>

    <div class="btn-group">

        <button class="btn btn-info">Button 3</button>

    </div>

    <div class="btn-group">

        <button class="btn btn-primary">Button 4</button>

    </div>

    <div class="btn-group">

        <button class="btn btn-success">Button 5</button>

    </div>

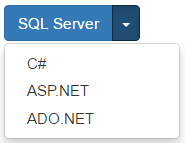
    <div class="btn-group">

        <button class="btn btn-warning">Button 6</button>

    </div>

</div>

### 19. Bootstrap split button dropdown

**Split buttons** have the primary action on the left and the rest of the actions are available as drop down menu items on the right   
   
  
**Here is the HTML**

<div class="btn-group">

    <button class="btn btn-primary">SQL Server</button>

    <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li><a href="#">C#</a></li>

        <li><a href="#">ASP.NET</a></li>

        <li><a href="#">ADO.NET</a></li>

    </ul>

</div>

To change the size of the split button dropdowns use btn-group-lg, btn-group-sm or btn-group-xs classes   
bootstrap split button dropdown size 

<div class="btn-group btn-group-lg">

    <button class="btn btn-primary">

        SQL Server

    </button>

    <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li><a href="#">C#</a></li>

        <li><a href="#">ASP.NET</a></li>

        <li><a href="#">ADO.NET</a></li>

    </ul>

</div>

<div class="btn-group btn-group-sm">

    <button class="btn btn-primary">

        SQL Server

    </button>

    <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li><a href="#">C#</a></li>

        <li><a href="#">ASP.NET</a></li>

        <li><a href="#">ADO.NET</a></li>

    </ul>

</div>

<div class="btn-group btn-group-xs">

    <button class="btn btn-primary">

        SQL Server

    </button>

    <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

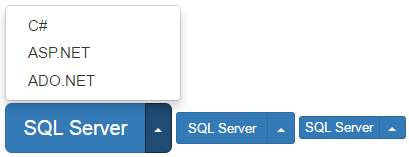
        <li><a href="#">C#</a></li>

        <li><a href="#">ASP.NET</a></li>

        <li><a href="#">ADO.NET</a></li>

    </ul>

</div>

To exapnd the menu items of a split button dropdown upwards use dropup class   
 

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

    <button class="btn btn-primary">

        SQL Server

    </button>

    <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li><a href="#">C#</a></li>

        <li><a href="#">ASP.NET</a></li>

        <li><a href="#">ADO.NET</a></li>

    </ul>

</div>

<div class="btn-group btn-group-sm dropup">

    <button class="btn btn-primary">

        SQL Server

    </button>

    <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li><a href="#">C#</a></li>

        <li><a href="#">ASP.NET</a></li>

        <li><a href="#">ADO.NET</a></li>

    </ul>

</div>

<div class="btn-group btn-group-xs dropup">

    <button class="btn btn-primary">

        SQL Server

    </button>

    <button class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

        <span class="caret"></span>

    </button>

    <ul class="dropdown-menu">

        <li><a href="#">C#</a></li>

        <li><a href="#">ASP.NET</a></li>

        <li><a href="#">ADO.NET</a></li>

    </ul>

</div>

### 20. Bootstrap forms

**Bootstrap provides the following 3 form layouts**  
1. Vertical form (Default)  
2. Horizontal form  
3. Inline form  
  
**The followin are the classes that are used to style forms**  
form-group - Use this class on the <div> element that wraps labels and form controls for optimum spacing  
form-control - Use this class on all textual elements (<input>, <textarea>, and <select>)  
  
**Example :** In the HTML below we have 2 text boxes and a submit button

<form>

    <div>

        <label for="inputUserName">Username</label>

        <input type="text" id="inputUserName" />

    </div>

    <div>

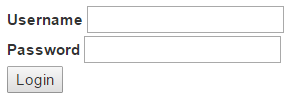
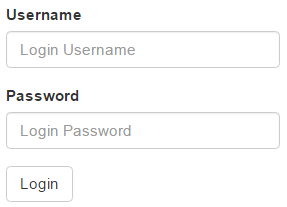
        <label for="inputPassword">Password</label>

        <input type="password" id="inputPassword" />

    </div>

    <button type="submit">Login</button>

</form>

At the moment, we have not applied any of the Bootstrap form classes. If we view this page, in the browser this is how the form looks. Not very pretty.   
  
  
**Creating a vertical form layout :** This is the default form layout. In the vertical form layout, the label and it's associated form control are stacked. Creating a vertical bootstrap form is very simple.  
1. Apply form-group class on the <div> element that wraps the label and the textbox  
2. Apply form-control class on the textbox   
  
 

<form>

    <div class="form-group">

        <label for="inputUserName">Username</label>

        <input class="form-control" placeholder="Login Username"

                type="text" id="inputUserName" />

    </div>

    <div class="form-group">

        <label for="inputPassword">Password</label>

        <input class="form-control" placeholder="Login Password"

                type="password" id="inputPassword" />

    </div>

    <button type="submit" class="btn btn-default">Login</button>

</form>

**Creating an inline form layout :** Inline form layout places the form controls side by side. To create an inline form layout, all you have to do is apply form-inline class on the <form> element.   
  
bootstrap inline form example 

<form class="form-inline">

    <div class="form-group">

        <label for="inputUserName">Username</label>

        <input class="form-control" placeholder="Login Username"

                type="text" id="inputUserName" />

    </div>

    <div class="form-group">

        <label for="inputPassword">Password</label>

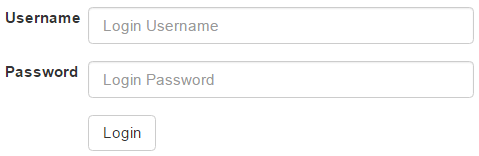
        <input class="form-control" placeholder="Login Password"

                type="password" id="inputPassword" />

    </div>

    <button type="submit" class="btn btn-default">Login</button>

</form>

**Please note :** You get the inline form layout only if the viewport is at least 768px wide. If the viewport width falls below 768px, the form layout reverts to vertical.  
  
**Creating horizontal form layout :** In the horizontal form layout, the label is on the left and it's associated form control is on the right in a single line. To create a horizontal form layout  
1. Use the form-horizontal class on the <form> element  
2. Use the control-label class on the <label> element  
3. Use Bootstrap's grid classes to align labels and form controls   
  
 

<form class="form-horizontal">

    <div class="form-group">

        <label for="inputUserName" class="control-label col-sm-2">Username</label>

        <div class="col-sm-10">

            <input class="form-control" placeholder="Login Username"

                    type="text" id="inputUserName" />

        </div>

    </div>

    <div class="form-group">

        <label for="inputPassword" class="control-label col-sm-2">Password</label>

        <div class="col-sm-10">

            <input class="form-control" placeholder="Login Password"

                    type="password" id="inputPassword" />

        </div>

    </div>

    <div class="form-group">

        <div class="col-sm-offset-2 col-sm-10">

            <button type="submit" class="btn btn-default">Login</button>

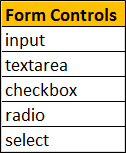
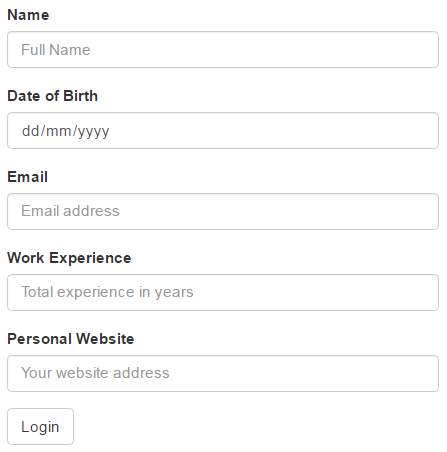
        </div>

    </div>

</form>

**Please note :**With this example, we will only have the horizontal form layout on a large, medium, and small screen sizes. On an extra small screen size, like a mobile phone, the form reverts to a vertical layout. If you want the horizontal layout on all screen sizes, including an extra small screen size, use col-xs-\* grid classes instead of col-sm-\* classes.

### 21.Bootstrap form controls

Bootstrap supports all the following standard form controls   
   
  
**Input :** In [Part 20](http://csharp-video-tutorials.blogspot.com/2016/06/bootstrap-forms.html) of [Bootstrap tutorial](https://www.youtube.com/playlist?list=PL6n9fhu94yhXd4xnk-j5FGhHjUv1LsF0V), we used <input type="text"> and <input type="password">. In addition to these 2 types all the other HTML5 types like date, number, email, url etc are also supported. Please note that the type attribute is required for bootstrap to fully style the input element.  
  
Here is a form with a few of the input types   
 

<form>

    <div class="form-group">

        <label for="inputName">Name</label>

        <input class="form-control" type="text" id="inputName"

                placeholder="Full Name" />

    </div>

    <div class="form-group">

        <label for="inputDOB">Date of Birth</label>

        <input class="form-control" type="date" id="inputDOB" />

    </div>

    <div class="form-group">

        <label for="inputEmail">Email</label>

        <input class="form-control" type="email" id="inputEmail"

                placeholder="Email address" />

    </div>

    <div class="form-group">

        <label for="inputExperience">Work Experience</label>

        <input class="form-control" type="number" id="inputExperience"

                placeholder="Total experience in years" />

    </div>

    <div class="form-group">

        <label for="inputUrl">Personal Website</label>

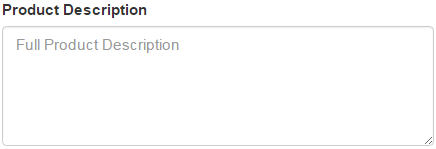
        <input class="form-control" type="url" id="inputUrl"

                placeholder="Your website address" />

    </div>

    <button type="submit" class="btn btn-default">Login</button>

</form>

**Textarea :** Use to capture multiple lines of text from the user.   
 

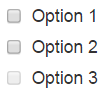
<div class="form-group">

    <label for="inputDescription">Product Description</label>

    <textarea class="form-control" id="inputDescription" rows="5"

                placeholder="Full Product Description"></textarea>

</div>

**Checkboxes :** Use when you want the user to select any number of options from the list of available options. The last option in the example below is disabled. To disable a checkbox, use disabled attribute on the checkbox. If you also want the cursor to be disabled on hover, then use disabled class on the parent <div> element of the checkbox.   
 

<div class="checkbox">

    <label>

        <input type="checkbox" value="option1">

        Option 1

    </label>

</div>

<div class="checkbox">

    <label>

        <input type="checkbox" value="option2">

        Option 2

    </label>

</div>

<div class="checkbox disabled">

    <label>

        <input type="checkbox" value="option3" disabled>

        Option 3

    </label>

</div>

If you want the checkboxes to appear side-by-side on the same line, use checkbox-inline class   
bootstrap checkboxes side by side 

<label class="checkbox-inline">

    <input type="checkbox" value="option1">Option 1

</label>

<label class="checkbox-inline">

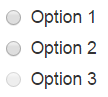
    <input type="checkbox" value="option2">Option 2

</label>

<label class="checkbox-inline">

    <input type="checkbox" value="option3">Option 3

</label>

**Radio Buttons :** Use when you want the user to select ONLY ONE option from the list of available options. The last option in the example below is disabled. To disable a radio button, use disabled attribute on the radio button. If you also want the cursor to be disabled on hover, then use disabled class on the parent <div> element of the radio button.   
 

<div class="radio">

    <label>

        <input type="radio" name="radioGroup" id="opt1" value="1">

        Option 1

    </label>

</div>

<div class="radio">

    <label>

        <input type="radio" name="radioGroup" id="opt2" value="2">

        Option 2

    </label>

</div>

<div class="radio disabled">

    <label>

        <input type="radio" name="radioGroup" id="opt3" value="3" disabled>

        Option 3

    </label>

</div>

If you want the radio buttons to appear side-by-side on the same line, use radio-inline class  
bootstrap radio buttons side by side 

<label class="radio-inline">

    <input type="radio" name="radioGroup" id="opt1" value="1">

    Option 1

</label>

<label class="radio-inline">

    <input type="radio" name="radioGroup" id="opt2" value="2">

    Option 2

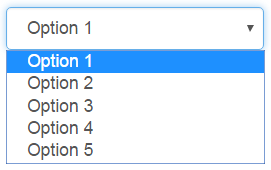
</label>

<label class="radio-inline">

    <input type="radio" name="radioGroup" id="opt3" value="3" disabled>

    Option 3

</label>

**Single Select List or Single Select DropDownList :** Use when you want the user to select one option from the list of available options.   
 

<select class="form-control">

    <option>Option 1</option>

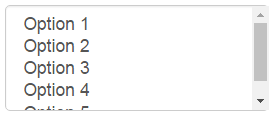
    <option>Option 2</option>

    <option>Option 3</option>

    <option>Option 4</option>

    <option>Option 5</option>

</select>

**Multi-Select List :** Use when you want the user to select any number of options from the list of available options. To select multiple options, hold down the CTRL key. For a multi-select list, use the multiple attribute on the <select> element.   
 

<select multiple class="form-control">

    <option>Option 1</option>

    <option>Option 2</option>

    <option>Option 3</option>

    <option>Option 4</option>

    <option>Option 5</option>

</select>

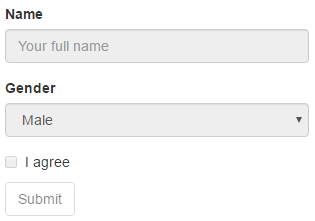
### 22. Bootstrap disabled and readonly form controls

**Classes / Attributes used to disable or make form controls readonly**

| **Class / Attribute** | **Purpose** |
| --- | --- |
| disabled | To disable individual controls or all the controls in a fieldset |
| readonly | To make form input controls readonly |
| form-control-static | To display plain text next to a label within a form |

To disable some of the form controls we may have to use both disabled attribute and disabled class. For example, In [Part 21](http://csharp-video-tutorials.blogspot.com/2016/06/bootstrap-form-controls.html) of [Bootstrap tutorial](https://www.youtube.com/playlist?list=PL6n9fhu94yhXd4xnk-j5FGhHjUv1LsF0V), to disable checkboxes and radio buttons we have used both disabled attribute and disabled class. Most of the form controls can be disabled just by adding disabled attribute. Disabled form controls does not allow their value to be changed.  
  
For example, to disable an input element it is enough if we just use the disabled attribute.

<input class="form-control" type="text" placeholder="Your name" disabled>

A fieldset with a disabled attribute, will disable all the from controls within it.  
 

<fieldset disabled>

    <div class="form-group">

        <label for="txtName">Name</label>

        <input type="text" id="txtName" class="form-control"

                placeholder="Your full name">

    </div>

    <div class="form-group">

        <label for="selectGender">Gender</label>

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

            <option>Male</option>

            <option>Female</option>

        </select>

    </div>

    <div class="checkbox">

        <label>

            <input type="checkbox"> I agree

        </label>

    </div>

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

</fieldset>

To make a form input control readonly, use readonly attribute. Readonly control does not allow the user to make any changes to it's content and appear just like a disabled control except the cursor style does not change to a stop sign on hover.

<input class="form-control" type="text" placeholder="Your name" readonly>

To display plain text next to a label in a form, use form-control-static class on a <p> element.   
 

<form class="form-horizontal">

    <div class="form-group">

        <label class="col-xs-2 control-label">Id</label>

        <div class="col-xs-10">

            <p class="form-control-static">101</p>

        </div>

    </div>

    <div class="form-group">

        <label for="txtFirstName" class="col-xs-2 control-label">First Name</label>

        <div class="col-xs-10">

            <input type="text" class="form-control" id="txtFirstName"

                    placeholder="Your first name">

        </div>

    </div>

    <div class="form-group">

        <label for="txtLastName" class="col-xs-2 control-label">Last Name</label>

        <div class="col-xs-10">

            <input type="text" class="form-control" id="txtLastName"

                    placeholder="Your last name">

        </div>

    </div>

</form>

23.Bootstrap form validation states

The following are the boostrap classes and glyphicons that can be used to style the form controls depending on the validation state i.e success, warning or error

| **Class** | **Glyphicon** | **Purpose** |
| --- | --- | --- |
| has-success | glyphicon glyphicon-ok | Success Validation State |
| has-warning | glyphicon glyphicon-warning-sign | Warning Validation State |
| has-error | glyphicon glyphicon-remove | Error Validation State |

The following example, styles the form controls depending on their validation state. The bootstrap validation classes (has-success, has-warning & has-error) are being used on the parent <div> element. We are also using help-block class to display the help text associated with the form control.   
 

<form class="form-horizontal">

    <div class="form-group has-success">

        <label class="col-xs-2 control-label" for="txtStrongPassword">Password</label>

        <div class="col-xs-10">

            <input type="password" id="txtStrongPassword" class="form-control"

                    placeholder="Your password">

            <span class="help-block">Strong Password</span>

        </div>

    </div>

    <div class="form-group has-warning">

        <label class="col-xs-2 control-label" for="txtWeakPassword">Password</label>

        <div class="col-xs-10">

            <input type="password" id="txtWeakPassword" class="form-control"

                    placeholder="Your password">

            <span class="help-block">Weak Password</span>

        </div>

    </div>

    <div class="form-group has-error">

        <label class="col-xs-2 control-label" for="txtAge">Age</label>

        <div class="col-xs-10">

            <input type="number" id="txtAge" class="form-control"

                    placeholder="Your Age">

            <span class="help-block">Invalid Age</span>

        </div>

    </div>

</form>

In the following example, we are also displaying the validation icons depending on the validation state of the control. To use icons we also have to use has-feedback and form-control-feedback classes. Please note that Icons only work with textual <input> elements   
 

<form class="form-horizontal">

    <div class="form-group has-success has-feedback">

        <label class="col-xs-2 control-label" for="txtStrongPassword">Password</label>

        <div class="col-xs-10">

            <input type="password" id="txtStrongPassword" class="form-control"

                    placeholder="Your password">

            <span class="glyphicon glyphicon-ok form-control-feedback"></span>

            <span class="help-block">Strong Password</span>

        </div>

    </div>

    <div class="form-group has-warning has-feedback">

        <label class="col-xs-2 control-label" for="txtWeakPassword">Password</label>

        <div class="col-xs-10">

            <input type="password" id="txtWeakPassword" class="form-control"

                    placeholder="Your password">

            <span class="glyphicon glyphicon-warning-sign form-control-feedback"></span>

            <span class="help-block">Weak Password</span>

        </div>

    </div>

    <div class="form-group has-error has-feedback">

        <label class="col-xs-2 control-label" for="txtAge">Age</label>

        <div class="col-xs-10">

            <input type="number" id="txtAge" class="form-control"

                    placeholder="Your Age">

            <span class="glyphicon glyphicon-remove form-control-feedback"></span>

            <span class="help-block">Invalid Age</span>

        </div>

    </div>

</form>

The following example, styles the input groups depending on their validation state. As you can see, styling input groups is similar to styling regular form controls.   
 

<form class="form-horizontal">

    <div class="form-group has-success has-feedback">

        <label class="col-xs-2 control-label" for="txtStrongPassword">Income</label>

        <div class="col-xs-10">

            <div class="input-group">

                <span class="input-group-addon">$</span>

                <input type="text" id="txtIncome" class="form-control"

                        placeholder="Your annual income after taxes">

            </div>

            <span class="glyphicon glyphicon-ok form-control-feedback"></span>

            <span class="help-block">Success</span>

        </div>

    </div>

    <div class="form-group has-warning has-feedback">

        <label class="col-xs-2 control-label" for="txtStrongPassword">Income</label>

        <div class="col-xs-10">

            <div class="input-group">

                <span class="input-group-addon">$</span>

                <input type="text" id="txtIncome" class="form-control"

                        placeholder="Your annual income after taxes">

            </div>

            <span class="glyphicon glyphicon-warning-sign form-control-feedback"></span>

            <span class="help-block">Warning</span>

        </div>

    </div>

    <div class="form-group has-error has-feedback">

        <label class="col-xs-2 control-label" for="txtStrongPassword">Income</label>

        <div class="col-xs-10">

            <div class="input-group">

                <span class="input-group-addon">$</span>

                <input type="text" id="txtIncome" class="form-control"

                        placeholder="Your annual income after taxes">

            </div>

            <span class="glyphicon glyphicon-remove form-control-feedback"></span>

            <span class="help-block">Error</span>

        </div>

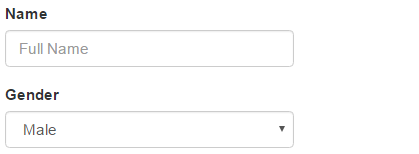
    </div>

</form>

24.Bootstrap form controls height and width

**Bootstrap classes to control the height and width of form controls**

| **Class** | **Purpose** |
| --- | --- |
| Bootstrap Grid Classes | To control the width of the form controls |
| input-lg or input-sm | To control the height of the form controls |
| form-group-lg or form-group-sm | To control the height of the form controls and associated labels |

**Controlling the width of the form controls :** In this example we are using predefined bootstrap grid classes to control the width of the form controls.    
 

<div class="container">

    <form>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputName">Name</label>

                <input class="form-control" type="text" id="inputName"

                        placeholder="Full Name" />

            </div>

        </div>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputDOB">Gender</label>

                <select class="form-control">

                    <option>Male</option>

                    <option>Female</option>

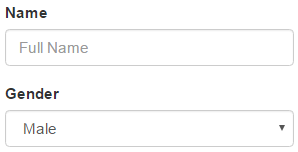
                </select>

            </div>

        </div>

    </form>

</div>

**Controlling the height of the form controls :** To control the height of the form controls you may use input-lg or input-sm classes. In the example below, we are not using any of these 2 classes, so we get the default height for the form controls.   
 

<div class="container">

    <form>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputName">Name</label>

                <input class="form-control" type="text" id="inputName"

                        placeholder="Full Name" />

            </div>

        </div>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputDOB">Gender</label>

                <select class="form-control">

                    <option>Male</option>

                    <option>Female</option>

                </select>

            </div>

        </div>

    </form>

</div>

To set a larger height for the form controls use input-lg class.   
 

<div class="container">

    <form>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputName">Name</label>

                <input class="form-control input-lg" type="text" id="inputName"

                        placeholder="Full Name" />

            </div>

        </div>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputDOB">Gender</label>

                <select class="form-control input-lg">

                    <option>Male</option>

                    <option>Female</option>

                </select>

            </div>

        </div>

    </form>

</div>

To set a smaller height for the form controls use input-sm class.   
 

<div class="container">

    <form>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputName">Name</label>

                <input class="form-control input-sm" type="text" id="inputName"

                        placeholder="Full Name" />

            </div>

        </div>

        <div class="row">

            <div class="form-group col-xs-3">

                <label for="inputDOB">Gender</label>

                <select class="form-control input-sm">

                    <option>Male</option>

                    <option>Female</option>

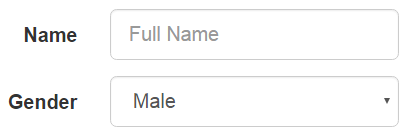
                </select>

            </div>

        </div>

    </form>

</div>

To control the height of the form controls and their associated labels on a horizontal form use form-group-lg or form-group-sm classes.  
  
To set a larger height for the form controls and their associated labels use form-group-lgclass   
 

<div class="container">

    <form class="form-horizontal">

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

            <label for="inputName" class="control-label col-xs-1">Name</label>

            <div class="col-xs-3">

                <input class="form-control input-sm" type="text" id="inputName"

                        placeholder="Full Name" />

            </div>

        </div>

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

            <label for="inputDOB" class="control-label col-xs-1">Gender</label>

            <div class="col-xs-3">

                <select class="form-control input-sm">

                    <option>Male</option>

                    <option>Female</option>

                </select>

            </div>

        </div>

    </form>

</div>

To set a smaller height for the form controls and their associated labels use form-group-smclass   
 

<div class="container">

    <form class="form-horizontal">

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

            <label for="inputName" class="control-label col-xs-1">Name</label>

            <div class="col-xs-3">

                <input class="form-control input-sm" type="text" id="inputName"

                        placeholder="Full Name" />

            </div>

        </div>

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

            <label for="inputDOB" class="control-label col-xs-1">Gender</label>

            <div class="col-xs-3">

                <select class="form-control input-sm">

                    <option>Male</option>

                    <option>Female</option>

                </select>

            </div>

        </div>

    </form>

</div>

25. Bootstrap input groups

**Bootstrap Input Groups**  
1. Extend <input> elements by adding text, icons or buttons before, after, or on both sides of the <input> elements  
2. Are limited to textual <input> elements only  
3. Avoid using <select> and <textarea> elements as they are not fully supported by some browsers   
  
**To add text before any text-based <input> element**  
1. Wrap the <input> element to which you want to add text, inside a <div> element with class=input-group  
2. Wrap the text within a <span> element with class=input-group-addon and place it before the <input> element   
bootstrap input-group-addon example 

<div class="input-group">

    <span class="input-group-addon">$</span>

    <input type="text" class="form-control" placeholder="Amount">

</div>

To add text after the <input> element, place the <span> element after the <input> element  
bootstrap 3 input group example 

<div class="input-group">

    <input type="text" class="form-control" placeholder="Amount">

    <span class="input-group-addon">.00</span>

</div>

To add text on bot the sides of the <input> element, place the <span> element on both the sides of the <input> element   
bootstrap span add on 

<div class="input-group">

    <span class="input-group-addon">$</span>

    <input type="text" class="form-control" placeholder="Amount">

    <span class="input-group-addon">.00</span>

</div>

To add an icon to the <input> element, create a <span> element with class="The Glyphicon Name" and place it inside the <span> element that has the class input-group-addon   
bootstrap glyphicons input group 

<div class="input-group">

    <span class="input-group-addon">

        <span class="glyphicon glyphicon-envelope"></span>

    </span>

    <input type="text" class="form-control" placeholder="Amount">

</div>

A checkbox or a radio button can also be used as an add-on, instead of text or icons. To append a checkbox to an <input> element, place the checkbox inside the <span> element that has the class input-group-addon   
bootstrap input append checkbox 

<div class="input-group">

    <span class="input-group-addon">

        <input type="checkbox" />

    </span>

    <input type="text" class="form-control" placeholder="Amount">

</div>

As you might have guessed by now, to append a radio button, simply place the radio button inside the <span> element that has the class input-group-addon   
bootstrap input-group radio button 

<div class="input-group">

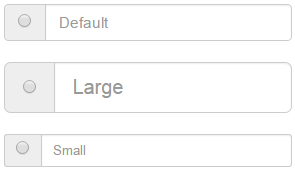
    <span class="input-group-addon">

        <input type="radio" />

    </span>

    <input type="text" class="form-control" placeholder="Amount">

</div>

For sizing the elements in an input group, use input-group-lg or input-group-sm classes on the input-group. This will size all the elements accordingly in that input-group. There is no need to apply the size classes on each element in the input-group.   
 

<div class="input-group">

    <span class="input-group-addon">

        <input type="radio" />

    </span>

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

</div>

<br />

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

    <span class="input-group-addon">

        <input type="radio" />

    </span>

    <input type="text" class="form-control" placeholder="Large">

</div>

<br />

<div class="input-group input-group-sm">

    <span class="input-group-addon">

        <input type="radio" />

    </span>

    <input type="text" class="form-control" placeholder="Small">

</div>

### 26.Bootstrap button in input group

To use a button in an input group use input-group-btn class instead of input-group-addonclass.   
bootstrap input group button 

<div class="input-group">

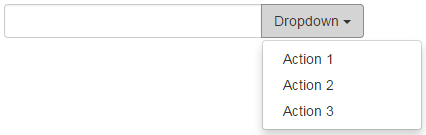
    <input type="text" class="form-control"/>

    <span class="input-group-btn">

        <button class="btn btn-default" type="button">Search</button>

    </span>

</div>

**Using a button dropdown in an input group**   
 

<div class="input-group">

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

    <div class="input-group-btn">

        <button type="button" data-toggle="dropdown"

                class="btn btn-default dropdown-toggle">

            Dropdown

            <span class="caret"></span>

        </button>

        <ul class="dropdown-menu">

            <li><a href="#">Action 1</a></li>

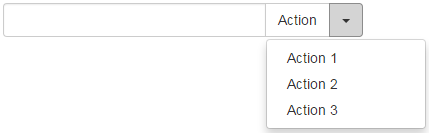
            <li><a href="#">Action 2</a></li>

            <li><a href="#">Action 3</a></li>

        </ul>

    </div>

</div>

Segmented buttons can also be added to the input element very easily as shown below. Notice we have a standard button along with a dropdown button appended on the right hand side of the input element.  
 

<div class="input-group">

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

    <div class="input-group-btn">

        <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 1</a></li>

            <li><a href="#">Action 2</a></li>

            <li><a href="#">Action 3</a></li>

        </ul>

    </div>

</div>

Multiple buttons can also be added to the input element.   
bootstrap input group multiple buttons 

<div class="input-group">

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

    <div class="input-group-btn">

        <button type="button" class="btn btn-default">Cut</button>

        <button type="button" class="btn btn-default">Copy</button>

        <button type="button" class="btn btn-default">Paste</button>

    </div>

</div>

Multiple buttons with icons added to the input element  
bootstrap input group button icon 

<div class="input-group">

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

    <div class="input-group-btn">

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

            <span class="glyphicon glyphicon-align-left"></span>

        </button>

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

            <span class="glyphicon glyphicon-align-center"></span>

        </button>

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

            <span class="glyphicon glyphicon-align-right"></span>

        </button>

    </div>

</div>

27.Bootstrap nav component

**Bootstrap nav component** makes it very easy to create navigation components i.e navigation tabs and pills. The following table shows the classes that are useful for creating navigation tabs and pills. 

| **Class** | **Purpose** |
| --- | --- |
| nav nav-tabs | Navigation tabs |
| nav nav-pills | Navigation pills |
| nav-stacked | Vertically stacked navigation pills |
| nav-justified | Justified pills or tabs. On screen size < 768px, the navigation links are stacked |
| disabled | Disables navigation tab or pill |
| **Dropdown menu can also be used with navigation tabs and pills** | |

**Please note :** The base markup for both navigation tabs and pills is in nav class.

**Creating navigation tabs :** Use nav-tabs class along with the base nav class.

bootstrap nav tabs example

<ul class="nav nav-tabs">

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

    <li><a href="#">Contact</a></li>

    <li><a href="#">About</a></li>

</ul>

Creating navigation tabs with icons :

bootstrap nav tabs icon

<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-earphone"></span> Contact

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-info-sign"></span> About

        </a>

    </li>

</ul>

**Creating navigation pills :** Use nav-pills class along with the base nav class.

bootstrap nav pills example

<ul class="nav nav-pills">

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

    <li><a href="#">Contact</a></li>

    <li><a href="#">About</a></li>

</ul>

**Creating navigation pills with icons:** Use a <span> element with class="Glyphicon Name"

bootstrap nav-pills icons

<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-earphone"></span> Contact

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-info-sign"></span> About

        </a>

    </li>

</ul>

**Creating stacked navigation pills :** To vertically stack navigation pills use nav-stackedclass



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

    <li class="active">

        <a href="#">

            <span class="glyphicon glyphicon-home"></span> Home

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-earphone"></span> Contact

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-info-sign"></span> About

        </a>

    </li>

</ul>

**Creating justified navigation pills :** Use nav-justified class

bootstrap justified nav-pills

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

    <li class="active">

        <a href="#">

            <span class="glyphicon glyphicon-home"></span> Home

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-earphone"></span> Contact

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-info-sign"></span> About

        </a>

    </li>

</ul>

**Creating justified navigation tabs :** Use nav-justified class

bootstrap justified nav tabs example

<ul class="nav nav-tabs nav-justified">

    <li class="active">

        <a href="#">

            <span class="glyphicon glyphicon-home"></span> Home

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-earphone"></span> Contact

        </a>

    </li>

    <li>

        <a href="#">

            <span class="glyphicon glyphicon-info-sign"></span> About

        </a>

    </li>

</ul>

**Disabling navigation links :** To make the links appear disabled use disabled class. This class only changes the appearance of the link, but not it's functionality. To disable link navigation, use custom JavaScript.

bootstrap disable nav pills

<ul class="nav nav-pills">

    <li class="active">

        <a href="#">

            <span class="glyphicon glyphicon-home"></span> Home

        </a>

    </li>

    <li class="disabled">

        <a href="#">

            <span class="glyphicon glyphicon-earphone"></span> Contact

        </a>

    </li>

    <li>

        <a href="#">

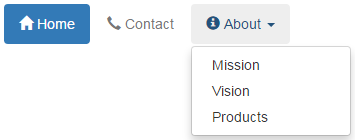
            <span class="glyphicon glyphicon-info-sign"></span> About

        </a>

    </li>

</ul>

**Navigation pills with dropdown menu :**



<ul class="nav nav-pills">

    <li class="active">

        <a href="#">

            <span class="glyphicon glyphicon-home"></span> Home

        </a>

    </li>

    <li class="disabled">

        <a href="#">

            <span class="glyphicon glyphicon-earphone"></span> Contact

        </a>

    </li>

    <li class="dropdown">

        <a href="#" class="dropdown-toggle" data-toggle="dropdown">

            <span class="glyphicon glyphicon-info-sign"></span> About

            <span class="caret"></span>

        </a>

        <ul class="dropdown-menu">

            <li><a href="#">Mission</a></li>

            <li><a href="#">Vision</a></li>

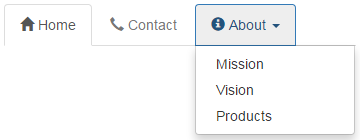
            <li><a href="#">Products</a></li>

        </ul>

    </li>

</ul>

**Navigation tabs with dropdown menu :** Same code as above, except nav-pills class is replaced with nav-tabs class



<ul class="nav nav-tabs">

    <li class="active">

        <a href="#">

            <span class="glyphicon glyphicon-home"></span> Home

        </a>

    </li>

    <li class="disabled">

        <a href="#">

            <span class="glyphicon glyphicon-earphone"></span> Contact

        </a>

    </li>

    <li class="dropdown">

        <a href="#" class="dropdown-toggle" data-toggle="dropdown">

            <span class="glyphicon glyphicon-info-sign"></span> About

            <span class="caret"></span>

        </a>

        <ul class="dropdown-menu">

            <li><a href="#">Mission</a></li>

            <li><a href="#">Vision</a></li>

            <li><a href="#">Products</a></li>

        </ul>

    </li>

</ul>