

Chapter-1

Getting Started with Bootstrap

Introduction

In Web development, User Interface plays key role. In scenarios, most of the developers used to create separate user interface pages for mobile and tablet and PC's. Using bootstrap we are able to create single front end framework for all devices instead of creating separate pages.

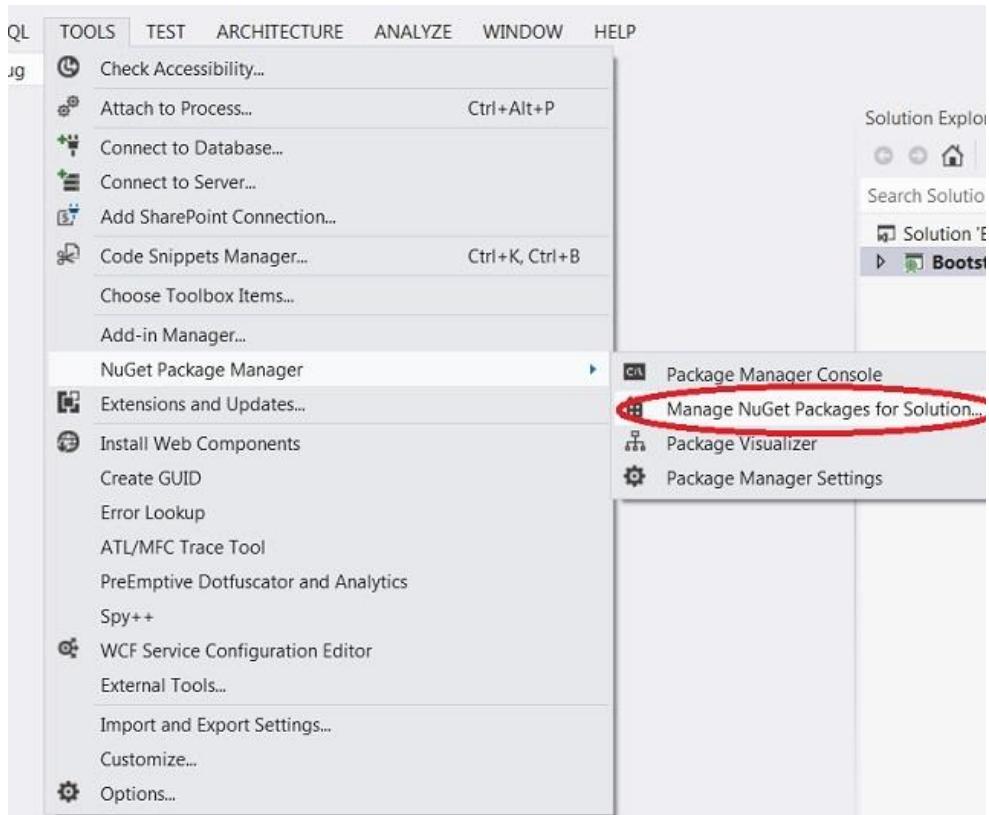
Getting Started

Bootstrap is powerful mobile first front end framework for faster and easy web development. It is very sleek, intuitive and any one can get started bootstrap with just knowledge of HTML and CSS.

Adding Bootstrap to a Project:

Adding bootstrap package to our project is pretty simple. Following are the steps involved in adding bootstrap package to our project.

- Go to NuGet Package manager and choose Manage NuGet packages for solution

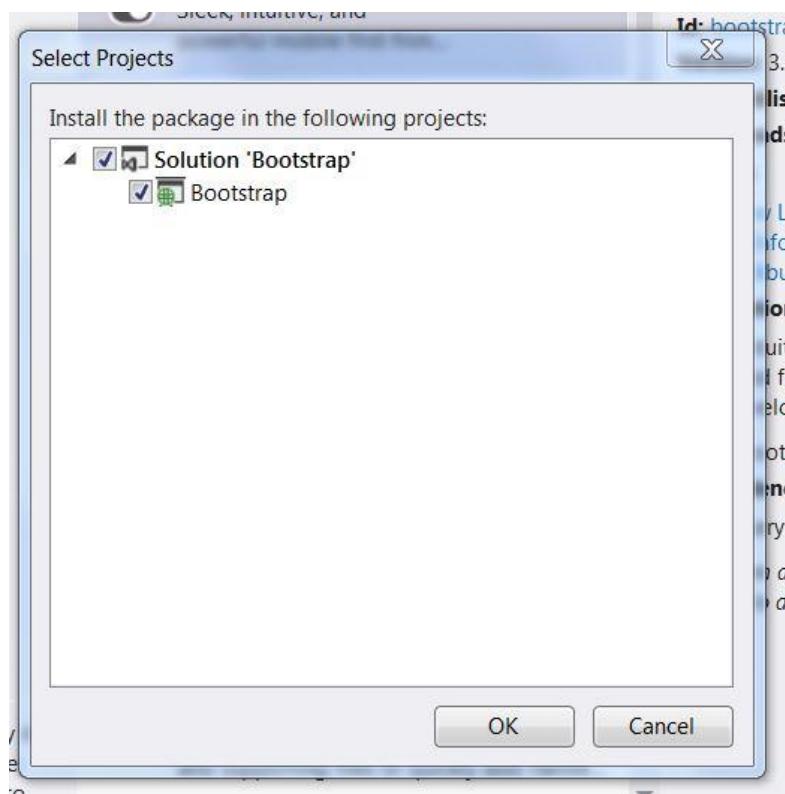


- Then manage NuGet packages pop-up window will appear.

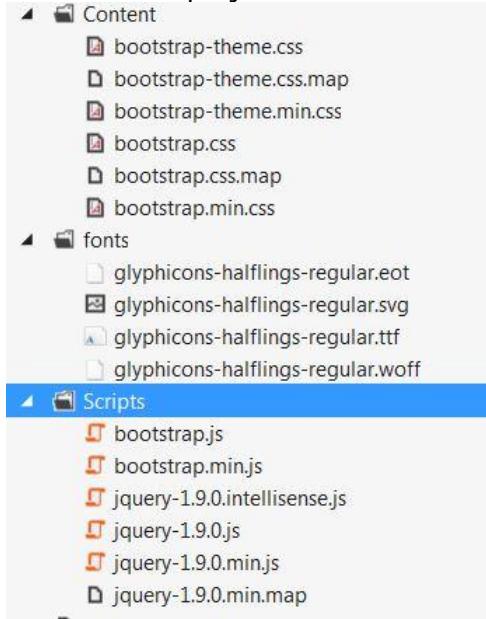
1. In that search for bootstrap at the top right corner search box then we will get list of several bootstrap packages.
2. Choose the very first Bootstrap package and click on install to add package to our project



- After clicking install button we will get a pop-up window for asking select projects to install. Here choose the bootstrap required projects and click on ok to complete the installation of this package.



- After completing this installation of bootstrap package, we will find the following files added to our project as show in below image



Using Bootstrap in a Project:

In a basic HTML page we will define bootstrap as follows

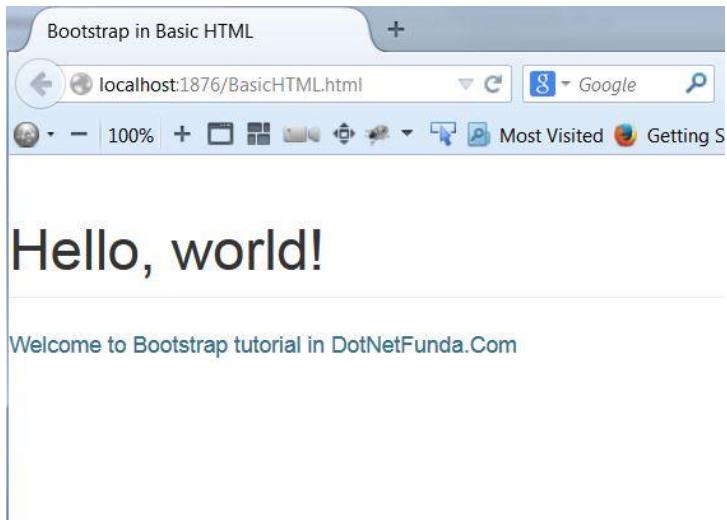
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Bootstrap in Basic HTML</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <!-- Bootstrap -->
    <link href="Content/bootstrap.min.css" rel="stylesheet" />
    <script src="Scripts/bootstrap.min.js"></script>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="Scripts/jquery-1.9.0.min.js"></script>
    <script src="https://code.jquery.com/jquery.js"></script>
</head>
<body>
    <h1 class="page-header">Hello, world!</h1>
    <p class="text-info">Welcome to Bootstrap tutorial in DotNetFunda.Com</p>
</body>
</html>
```

Here we will observe that the bootstrap.min.css, bootstrap.min.js, jquery.js files included in the above HTML page to make basic HTML page to Bootstrap Template.

Output of this page is as shown bellow



Conclusion

In this chapter, we have learnt how to start working with bootstrap, Hope this topic was useful.

We will see more about bootstrap in following chapters.

Thanks for reading.

Chapter-2

Mobile first Layout and Typography

Introduction

Bootstrap comes with predefined CSS, which allows us to create responsive designs faster and very easy way.

Using the code

To create mobile first layout make sure to define Meta tag with name is `Viewport` and content is `width=device-width` and for touch zooming include `initial-scale = 1.0` in content.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

If we want to disable zooming in mobile devices just add `user-scalable = no` to the viewport Meta tag and also include `maximum-scale = 1.0`

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
```

Define Media:

We will define display media by define categorized device widths using `@media ()` attribute. In bootstrap we are having categorized media widths for responsive mobile first design.

```
//for extra small devices i.e., phones  
@media (max-width: 768px) { ... }  
  
// for small devices i.e., tablets  
@media (min-width: 768px) and (max-width: 992px) { ... }  
  
// for medium devices i.e., desktops  
@media (min-width: 992px) and (max-width: 1200px) { ... }  
  
//for large devices i.e., large desktops  
@media (min-width: 1200px) { ... }
```

In Bootstrap we are having containers to set the content automatically to center of the page for respective device width. The container is defined with `.container`, `.container-fluid` class in Bootstrap CSS.

```
<div class="container">  
.....  
<p>Bootstrap container with fixed width</p>  
.....  
</div>
```

```
<div class=" container-fluid">
.....
<p>Bootstrap container with full width<p>
.....
</div>
```

Typography:

Usually we will set the background color by defining `background-color: #fff;` in body. Bootstrap sets basic global display and typography and links.

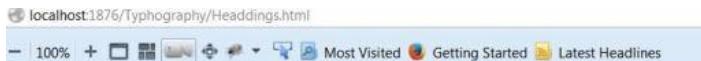
In bootstrap we are having some attributes to define typography for global display like `@font-family-base`, `@font-size-base` and `@line-height-base`. To display link color we are having `@link-color` attribute.

Usually we define heading tags as `h1`, `h2`, `h3`, `h4`, `h5`, `h6`. In bootstrap along with these tags we have also available `.h1`, `.h2`, `.h3`, `.h4`, `.h5`, `.h6` classes. Similarly we have `small` tag and `.small` class available to display small text along with the heading.

```
<div class="container">
    <h1>The heading one</h1>
    <h2>The heading two</h2>
    <h3>The heading three</h3>
    <p class="h4">The heading four</p>
    <p class="h5">The heading five</p>
    <p class="h6">The heading six</p>
<br />
    <p class="h1">The heading one <small>small text with heading</small></p>
    <p class="h2">The heading two <small>small text with heading</small></p>
    <p class="h3">The heading three <small>small text with heading</small></p>
    <h4>The heading four <span class="small">small text with heading</span></h4>
    <h5>The heading five <span class="small">small text with heading</span></h5>
    <h6>The heading six <span class="small">small text with heading</span></h6>
</div>
```

Here we have defined heading and small text tags are in both the ways as we already discussed.

The result page is as shown below



The heading one small text with heading

The heading two small text with heading

The heading three small text with heading

The heading four small text with heading

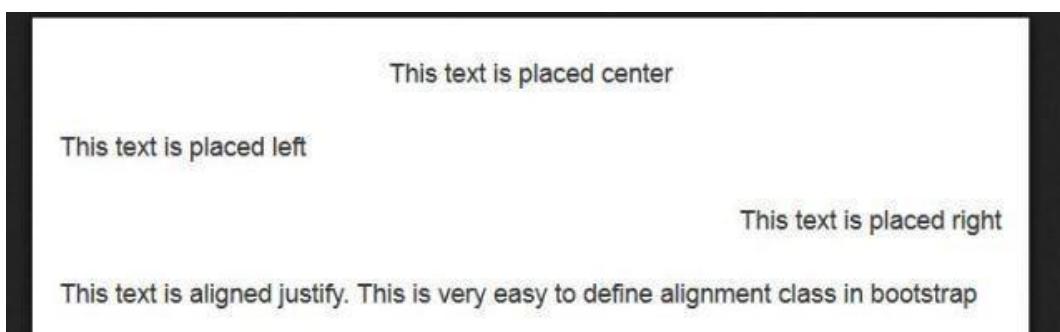
The heading five small text with heading

The heading six small text with heading

Similarly we have class for alignment. Here alignment class is alignment type comes along with prefix class name **text-** so this is very easy to define

```
<div class="container">
    <div class="text-center">This text is placed center </div>
    <div class="text-left">This text is placed left</div>
    <div class="text-right">This text is placed right</div>
    <div class="text-justify">This text is aligned justify. This is very easy to
define alignment class in bootstrap</div>
</div>
```

The result page is as shown below



List tags are much usable in front end web development. Here we have simple list class **.list-unstyle**, **.list-inline**, **list-group**, and **list-group-items** to define lists.

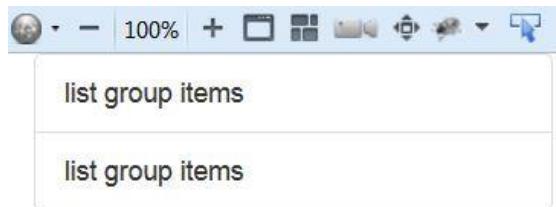
`.list-unstyle` will remove the current list style, it is majorly used at immediate child list.

`.list-inline` will used to display list items in one line.

`.list-group, list-group-items` used to display list items as a group

```
<div class="container">
    <ul class="list-group">
        <li class="list-group-item">list group items</li>
        <li class="list-group-item">list group items</li>
    </ul>
    <br />
    <ul class="list-inline">
        <li>In line items</li>
        <li>In line items</li>
        <li>In line items</li>
    </ul>
    <br />
    <ul>
        <li>list items</li>
        <li>list items</li>
        <li class="list-unstyled">chaild list</li>
        <li class="list-unstyled">chaild list</li>
        <li class="list-unstyled">chaild list</li>
        <li>list items</li>
        <li>list items</li>
    </ul>
</div>
```

The resultant page is as shown below:



In line items

- list items
 - list items
 - chaild list
 - chaild list
 - chaild list
 - list items
 - list items

There are few more useful classes in Bootstrap.

.lead class is for slightly larger and stand out text for a paragraph.

.blockquote-reverse class for align `blockquote` tag content at right side.

.initialism class is for slightly smaller text for `abbr` tag to define abbreviation.

```
<div class="container">
  <p>This is plain text in the bootstrap</p>
  <p class="lead">This is text with .lead class in bootstrap</p>
  <br />
  <blockquote>This is text from blockquote</blockquote>
  <blockquote class="blockquote-reverse">Text from blockquote with .blockquote-reverse
class</blockquote>
  <br />
  Normal abbr tag is:<abbr title="DotNetFunda"> DNF </abbr>&ampnbsp&ampnbsp&ampnbsp
<abbr title="DotNetFunda" class="initialism"> DNF </abbr>is abbr tag with initialism
class
</div>
```

The resultant page is as shown below:

This is plain text in the bootstrap

This is text with .lead class in bootstrap

This is text from blockquote

This is text from blockquote

Normal abbr tag is: DNF DNF is abbr tag with initialism class

Conclusion

In this chapter, we have learnt how to use typography classes in bootstrap. Hope this topic was helpful.

We will see bootstrap grid system in next chapter.

Thanks for reading.

Chapter-3

Grid System in Bootstrap

Introduction

Bootstrap Grid system plays key role in front end development. Bootstrap grid system allow us to create fluid responsive layout designs faster and easy way.

Using the code

In Bootstrap, we are having mobile first fluid grid system up to 12 columns to scales appropriately by increasing or decreasing the width of respective devices. The key break points for grid system based on the respective device width is depends on defined `@media` attribute minimum and maximum widths

Grid systems are used to create page layouts with rows and columns. Lets take a look on how to use grid system in bootstrap.

We are having predefined class `.row` to define rows. Rows must be placed in the container for proper padding and alignment.

For columns we are having various class for different devices based on device width to create responsive design view. In grid system devices classified as `xs`, `sm`, `md`, `lg`.

- `xs` is extra small screen devices with maximum of `768px` width and below (i.e., phones)
- `sm` is small screen deices with minimum of `768px` width and maximum of `992px` width (i.e., tablets)
- `md` is medium screen devices with minimum of `992px` width and maximum of `1200px` width (i.e., desktops)
- `lg` is large screen devices with minimum of `1200px` width and above (i.e., large desktops)

Here we have predefined column classes are `.col-xs/sm/nd/lg-*` (* is width of column in terms of columns).

We already know that grid system scales appropriately up to 12 columns.

Take look on this below code to get clear knowledge on these grid system columns

```
<div class="container">
<br />
<h2 class="page-header">
    Three columns in a row for all devices
</h2>
<div class="row">
    <!-- the below code is for 3 columns in a row for all devices--&gt;
    &lt;div class="col-lg-4 col-md-4 col-sm-4 col-xs-4"&gt;
        &lt;p class="text-danger"&gt;
            One of three columns in a row example. DotNetFunda.Com
        &lt;/p&gt;
    &lt;/div&gt;</pre>
```

```

<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <p class="text-info">
        Two of three columns in a row example. DotNetFunda.Com
    </p>
</div>
<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <p class="text-success">
        Three of three columns in a row example. DotNetFunda.Com
    </p>
</div>
<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <p class="text-warning">
        Four of three columns in a row example. DotNetFunda.Com
    </p>
</div>
<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <p class="text-primary">
        Five of three columns in a row example. DotNetFunda.Com
    </p>
</div>
<div class="col-lg-4 col-md-4 col-sm-4 col-xs-4">
    <p class="text-danger">
        Six of three columns in a row example. DotNetFunda.Com
    </p>
</div>
</div>
<br />
<h2 class="page-header">
    Reduce the no of columns in a row based on the device
</h2>
<div class="row">
    <!-- the below code is for in a row 6 columns for large screen, 4 columns for medium screen, 3 columns for small screen and 2 columns for extra small screen -->
    <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
        <p class="text-danger">
            1st column example. DotNetFunda.Com
        </p>
    </div>
    <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
        <p class="text-info">
            2nd column example. DotNetFunda.Com
        </p>
    </div>
    <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
        <p class="text-success">
            3rd column example. DotNetFunda.Com
        </p>
    </div>
    <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
        <p class="text-warning">
            4th column example. DotNetFunda.Com
        </p>
    </div>
    <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
        <p class="text-primary">
            5th column example. DotNetFunda.Com
        </p>
    </div>
</div>

```

```
<div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
    <p class="text-danger">
        6th column example. DotNetFunda.Com
    </p>
</div>
</div>
</div>
```

The output of the respective screens are as shown below:

In large display screens:

In this only large screen classes' .col-Lg-4 and .col-Lg-2 will applied for those respective tags.



Three columns in a row for all devices

One of three columns in a row example.
DotNetFunda.Com

Two of three columns in a row example.
DotNetFunda.Com

Three of three columns in a row example.
DotNetFunda.Com

Four of three columns in a row example.
DotNetFunda.Com

Five of three columns in a row example.
DotNetFunda.Com

Six of three columns in a row example.
DotNetFunda.Com

Reduce the no of coumns in a row based on the device

1st column example.
DotNetFunda.Com

2nd column example.
DotNetFunda.Com

3rd column example.
DotNetFunda.Com

4th column example.
DotNetFunda.Com

5th column example.
DotNetFunda.Com

6th column example.
DotNetFunda.Com

In medium display screens:

In this only medium screen classes' `.col-md-4` and `.col-md-3` will applied for those respective tags.



Three columns in a row for all devices

One of three columns in a row example.
[DotNetFunda.Com](#)

Two of three columns in a row example.
[DotNetFunda.Com](#)

Three of three columns in a row example.
[DotNetFunda.Com](#)

Four of three columns in a row example.
[DotNetFunda.Com](#)

Five of three columns in a row example.
[DotNetFunda.Com](#)

Six of three columns in a row example.
[DotNetFunda.Com](#)

Reduce the no of coumns in a row based on the device

1st column example.
[DotNetFunda.Com](#)

2nd column example.
[DotNetFunda.Com](#)

3rd column example.
[DotNetFunda.Com](#)

4th column example.
[DotNetFunda.Com](#)

5th column example.
[DotNetFunda.Com](#)

6th column example.
[DotNetFunda.Com](#)

In small display screens:

In this only small screen classes' `.col-sm-4` and `.col-sm-3` will applied for those respective tags.



Three columns in a row for all devices

One of three columns in a row example. [DotNetFunda.Com](#)

Two of three columns in a row example. [DotNetFunda.Com](#)

Three of three columns in a row example. [DotNetFunda.Com](#)

Four of three columns in a row example. [DotNetFunda.Com](#)

Five of three columns in a row example. [DotNetFunda.Com](#)

Six of three columns in a row example. [DotNetFunda.Com](#)

Reduce the no of coumns in a row based on the device

1st column example.
[DotNetFunda.Com](#)

2nd column example.
[DotNetFunda.Com](#)

3rd column example.
[DotNetFunda.Com](#)

4th column example.
[DotNetFunda.Com](#)

5th column example.
[DotNetFunda.Com](#)

6th column example.
[DotNetFunda.Com](#)

In extra small display screens:

In this only extra small screen classes' `.col-xs-4` and `.col-xs-6` will applied for those respective tags.



Three columns in a row for all devices

One of three columns in a row example.
DotNetFunda.Com

Four of three columns in a row example.
DotNetFunda.Com

Two of three columns in a row example.
DotNetFunda.Com

Five of three columns in a row example.
DotNetFunda.Com

Three of three columns in a row example.
DotNetFunda.Com

Six of three columns in a row example. DotNetFunda.Com

Reduce the no of columns in a row based on the device

1st column example. DotNetFunda.Com

3rd column example. DotNetFunda.Com

5th column example. DotNetFunda.Com

2nd column example. DotNetFunda.Com

4th column example. DotNetFunda.Com

6th column example. DotNetFunda.Com

In the code we used class for all four predefined device widths. If we need design for only some of them remove the remaining classes like `class="col-md-* col-xs-*"`. Here we given class for extra small and medium devices.

We have `.clearfix` class in grid system to set the column heights and appear correctly with in bounded area in responsive designs, If their column heights doesn't match with each other.

We are using `.clearfix` class along with bootstrap utility helper class.

```

<div class="container">
    <h2>Columns without clearfix class</h2>
    <div class="row">
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-danger">1st column example. DotNetFunda.Com.</p>
DotNetFunda.Com.1st column example. DotNetFunda.Com</p>
        </div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-info">2nd column example. DotNetFunda.Com</p>
        </div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-success">3rd column example. DotNetFunda.Com</p>
        </div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-warning">4th column example. DotNetFunda.Com</p>
        </div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-primary">5th column example. DotNetFunda.Com</p>
        </div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-danger">6th column example. DotNetFunda.Com</p>
        </div>
    </div>
    <br />
    <h2>Columns with clearfix class</h2>
    <div class="row">
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-danger">1st column example. DotNetFunda.Com.</p>
DotNetFunda.Com.1st column example. DotNetFunda.Com</p>
        </div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-info">2nd column example. DotNetFunda.Com</p>
        </div>
        <!-- it fixes the row height in extra small screens -->
        <div class="clearfix visible-xs"></div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-success">3rd column example. DotNetFunda.Com</p>
        </div>
        <!-- it fixes the row height in small screens -->
        <div class="clearfix visible-sm"></div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-warning">4th column example. DotNetFunda.Com</p>
        </div>
        <!-- it fixes the row height in medium screens -->
        <div class="clearfix visible-md"></div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-primary">5th column example. DotNetFunda.Com</p>
        </div>
        <div class="col-lg-2 col-md-3 col-sm-4 col-xs-6">
            <p class="text-danger">6th column example. DotNetFunda.Com</p>
        </div>
    </div>
</div>

```

In large display screens:

For large screens we doesn't use `.clearfix` class. In the below image we can observe the both scenarios are same.



Columns without clearfix class

1st column example. DotNetFunda.Com. DotNetFunda.Com.1st column example. DotNetFunda.Com	2nd column example. DotNetFunda.Com	3rd column example. DotNetFunda.Com	4th column example. DotNetFunda.Com	5th column example. DotNetFunda.Com	6th column example. DotNetFunda.Com
--	--	--	--	--	--

Columns with clearfix class

1st column example. DotNetFunda.Com. DotNetFunda.Com.1st column example. DotNetFunda.Com	2nd column example. DotNetFunda.Com	3rd column example. DotNetFunda.Com	4th column example. DotNetFunda.Com	5th column example. DotNetFunda.Com	6th column example. DotNetFunda.Com
--	--	--	--	--	--

In medium display screens:

For medium screens we used `.clearfix` class after four columns because we showed four columns in a row for medium screens. Here in the first case 1st column takes two rows place, in the second scenario `.clearfix` class sets the heights as same for first four columns. In the below image we can observe that.



Columns without clearfix class

1st column example. DotNetFunda.Com. DotNetFunda.Com.1st column example. DotNetFunda.Com	2nd column example. DotNetFunda.Com	3rd column example. DotNetFunda.Com	4th column example. DotNetFunda.Com	5th column example. DotNetFunda.Com	6th column example. DotNetFunda.Com
---	--	--	--	--	--

Columns with clearfix class

1st column example. DotNetFunda.Com. DotNetFunda.Com.1st column example. DotNetFunda.Com	2nd column example. DotNetFunda.Com	3rd column example. DotNetFunda.Com	4th column example. DotNetFunda.Com	5th column example. DotNetFunda.Com	6th column example. DotNetFunda.Com
---	--	--	--	--	--

In small display screens:

For small screens we used `.clearfix` class after three columns because we showed three columns in a row for small screens. Here in the first case 1st column takes two rows place, in the second scenario `.clearfix` class sets the heights as same for first three columns. In the below image we can observe that.



Columns without clearfix class

1st column example.

[DotNetFunda.Com](#)

DotNetFunda.Com. 1st column
example. [DotNetFunda.Com](#)

6th column example.

[DotNetFunda.Com](#)

2nd column example.

[DotNetFunda.Com](#)

4th column example.
[DotNetFunda.Com](#)

3rd column example.

[DotNetFunda.Com](#)

5th column example.
[DotNetFunda.Com](#)

Columns with clearfix class

1st column example.

[DotNetFunda.Com](#)

DotNetFunda.Com. 1st column
example. [DotNetFunda.Com](#)

2nd column example.

[DotNetFunda.Com](#)

3rd column example.

[DotNetFunda.Com](#)

4th column example.

[DotNetFunda.Com](#)

5th column example.

[DotNetFunda.Com](#)

6th column example.

[DotNetFunda.Com](#)

In extra small display screens:

For extra small screens we used `.clearfix` class after two columns because we showed two columns in a row for extra small screens. Here in the first case 1st column takes two rows place, in the second scenario `.clearfix` class sets the heights as same for first two columns. In the below image we can observe that.



Columns without clearfix class

1st column example. DotNetFunda.Com.

DotNetFunda.Com.1st column example.

DotNetFunda.Com

5th column example. DotNetFunda.Com

2nd column example. DotNetFunda.Com

3rd column example. DotNetFunda.Com

4th column example. DotNetFunda.Com

6th column example. DotNetFunda.Com

Columns with clearfix class

1st column example. DotNetFunda.Com.

DotNetFunda.Com.1st column example.

DotNetFunda.Com

3rd column example. DotNetFunda.Com

5th column example. DotNetFunda.Com

2nd column example. DotNetFunda.Com

4th column example. DotNetFunda.Com

6th column example. DotNetFunda.Com

If we observe the above pictures, we got the clear difference between the columns with and without `.clearfix` classes

Bootstrap Grid system we have `.col-xs/sm/md/lg-offset-*`(* is 0 to 12) classes to set columns placement after certain no of columns which we mentioned in the class name.

```
<div class="container">
    <h2>Offset class columns</h2>
    <div class="row">
        <div class="col-md-offset-6">
            <p class="thumbnail">This is column with offset style</p>
        </div>
        <div class="col-md-6">
            <p class="thumbnail">This is column with six columns width</p>
        </div>
        <div class="col-md-2 col-md-offset-4">
            <p class="thumbnail">This is column with offset style with two columns
width</p>
        </div>
    </div>
</div>
```

In this code, the first column placed after six columns with auto width and in the third column we have defined column width as 2 columns(out of 12 columns) and it was placed after 4 columns (out of 12) width from the second column. We will observed that in the below image.

Offset class columns



Using bootstrap grid system we will able to create nested columns faster and easy way.

```
<h2>Nested columns</h2>
<div class="row">
    <div class="col-md-4">
        <p class="thumbnail">This text placed in column with four columns
width</p>
    </div>
    <div class="col-md-4">
        <div class="row">
            <div class="col-md-6">
                <p class="thumbnail">nested column</p>
            </div>
            <div class="col-md-6">
                <p class="thumbnail">nested column</p>
            </div>
            <div class="col-md-6">
                <p class="thumbnail">nested column</p>
            </div>
            <div class="col-md-6">
                <p class="thumbnail">nested column</p>
            </div>
        </div>
    </div>
    <div class="col-md-4">
        <p class="thumbnail">This text placed in column with four columns
width</p>
    </div>
</div>
```

We can easily create nested columns in bootstrap.In the above code, we will have nested columns in place of second column. Nested columns is normal grid system columns those are placed inside the grid system column.We will observe the same in the below image.

Nested columns



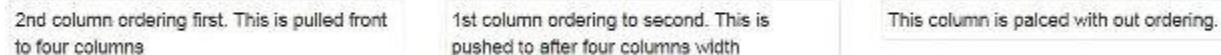
In Bootstrap grid system we have another useful `pull-*` and `push-*` classes along with `.col-xs/sm/md/lg-` prefix classes to order columns in a manner.

```
<h2>Ordered columns</h2>
    <div class="row">
        <div class="col-md-4 col-md-push-4">
            <p class="thumbnail">1st column ordering to second. This is pushed to
after four columns width</p>
        </div>
        <div class="col-md-4 col-md-pull-4">
            <p class="thumbnail">2nd column ordering first. This is pulled front to
four columns</p>
        </div>
        <div class="col-md-4">
            <p class="thumbnail">This column is placed with out ordering.</p>
        </div>
    </div>
```

In the above code, the first column the ordered class as `.col-md-push-4` is pushes the column to after 4 columns from its actual placement with the column width of 4 columns.

The second column the ordered class as `.col-md-pull-4` is pull the column forward to 4 columns from its actual placement. We will observe the same in the below image

Ordered columns



Conclusion

In this chapter, we have learnt working with grid system in bootstrap, Hope this topic was useful.

We will see more about bootstrap in following chapters.

Thanks for reading and do refer <http://www.dotnetfunda.com> to your friends and colleagues.

Chapter-4

Tables

Introduction

Twitter Bootstrap contains predefined CSS of all functions to make user easy to design. Now let's see creating Tables by using Bootstrap CSS.

Using the code

There are so many predefined classes in the `bootstrap.css` file to create different types of tables. Let's see one by one briefly.

We have the following code below as an example to create table

```
<table class="table">
  <thead>
    <tr>
      <th>S/no.</th>
      <th>Name</th>
      <th>Details</th>
      <th>Contact</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>DotNetFunda</td>
      <td>Fundamentals of .Net</td>
      <td>support@dotnetfunda.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>ItFunda</td>
      <td>Training of Microsoft Technologies</td>
      <td>support@itfunda.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>KidsFunda</td>
      <td>Learning Fundamentals for Kids</td>
      <td>support@kidsfunda.com</td>
    </tr>
  </tbody>
</table>
```

Now open the `bootstrap.css` file in your project and you will see a lot of CSS written in it. Dont get confused.

Just Press `Ctrl+F` and you will find search box. Search for '`table`' keyword.

Now there is a lot of CSS you can see under the `table` class. `tbody` and `thead` tags in the above code are being used from Bootstrap. You can see them in the `bootstrap.css`.

Output of the above code

S/no.	Name	Details	Contact
1	DotNetFunda	Fundamentals of .Net	support@dotnetfunda.com
2	ItFunda	Training of Microsoft Technologies	support@itfunda.com
3	KidsFunda	Learning Fundamentals for Kids	support@kidsfunda.com

If you want to make the table as a bordered one, just add '`table-bordered`' to the table class in your code as below.

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

Now you will see the table with borders in your browser as below,

S/no.	Name	Details	Contact
1	DotNetFunda	Fundamentals of .Net	support@dotnetfunda.com
2	ItFunda	Training of Microsoft Technologies	support@itfunda.com
3	KidsFunda	Learning Fundamentals for Kids	support@kidsfunda.com

And if you want to change your table with alternate backgrounds(alternate table rows), just add '`table-striped`' to the table class in your code as below.

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

Now you will see the table with alternate backgrounds in your browser. like below,

S/no.	Name	Details	Contact
1	DotNetFunda	Fundamentals of .Net	support@dotnetfunda.com
2	ItFunda	Training of Microsoft Technologies	support@itfunda.com
3	KidsFunda	Learning Fundamentals for Kids	support@kidsfunda.com

And if you want to add a hover state to the table rows, then add '`table-hover`' to the table class of your code as below,

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

Now run the code and you will see the hover effect for table rows like below.

S/no.	Name	Details	Contact
1	DotNetFunda	Fundamentals of .Net	support@dotnetfunda.com
2	ItFunda	Training of Microsoft Technologies	support@itfunda.com
3	KidsFunda	Learning Fundamentals for Kids	support@kidsfunda.com

Also we can change background colors of rows by using predefined color classes of Bootstrap as below code.

```
<table class="table table-hover">
  <thead>
    <tr>
      <th>S/no.</th>
      <th>Name</th>
      <th>Details</th>
      <th>Contact</th>
    </tr>
  </thead>
  <tbody>
    <tr class="warning">
      <td>1</td>
      <td>DotNetFunda</td>
      <td>Fundamentals of .Net</td>
      <td>support@dotnetfunda.com</td>
    </tr>
    <tr class="danger">
      <td>2</td>
      <td>ItFunda</td>
      <td>Training of Microsoft Technologies</td>
      <td>support@itfunda.com</td>
    </tr>
    <tr class="success">
      <td>3</td>
      <td>KidsFunda</td>
      <td>Learning Fundamentals for Kids</td>
      <td>support@kidsfunda.com</td>
    </tr>
  </tbody>
</table>
```

In the above code you can see that we added classes to the table rows which will change the background colors of the rows.

Now the output of the above code becomes as below,

S/no.	Name	Details	Contact
1	DotNetFunda	Fundamentals of .Net	support@dotnetfunda.com
2	IIFunda	Training of Microsoft Technologies	support@iifunda.com
3	KidsFunda	Learning Fundamentals for Kids	support@kidsfunda.com

Conclusion

In this chapter we have seen creating different types tables in HTML using Bootstrap. You can change the 'CSS' in your `bootstrap.css` as per your convenience. Hope you understand.

Thanks for reading.

Chapter-5

Lists

Introduction

So far we have seen that, how to create different types of tables using Bootstrap in [Chapter-4](#). Now let's see how to do listing styles using Bootstrap.

Using the code

For simple un-ordered list, try the below code.

```
<ul class="list-unstyled">
    <li>Home</li>
    <li>
        Training
        <ul>
            <li>Online Training</li>
            <li>Offline Training</li>
        </ul>
    </li>
    <li>Placement</li>
    <li>About Us</li>
    <li>Contact Us</li>
</ul>
```

You will see the following output in your browser,



```
Home
Training
    • Online Training
    • Offline Training
Placement
About Us
Contact Us
```

Now, change the class `.list-unstyled` as `.list-inline` like below,

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

Now you will see your items in line in your browser.

```
Home Training Placement About Us Contact Us
```

Creating Definition lists using Bootstrap

To Create Definition lists, we have to use '**dl**' tag in Html.

```
<dl class="dl-horizontal">
    <dt>DotNetFunda</dt>
    <dd>Dotnetfunda is the tutorials website to learn the fundamentals of .Net.</dd>
    <dt>ItFunda</dt>
    <dd>Itfunda is for .Net training and placement. We can buy e-books, materials of microsoft technologies in ItFunda. </dd>
    <dt>KidsFunda</dt>
    <dd>This is designed for learning fundamentals to kids.</dd>
</dl>
```

The above example shows a definition lists in your browser like below.

DotNetFunda	Dotnetfunda is the tutorials website to learn the fundamentals of .Net.
ItFunda	Itfunda is for .Net training and placement. We can buy e-books, materials of microsoft technologies in ItFunda.
KidsFunda	This is designed for learning fundamentals to kids.

List Grouping using Bootstrap

There are some predefined grouping classes in **bootstrap.css**. See the below example,

```
<ul class="list-group">
    <li class="list-group-item">Home</li>
    <li class="list-group-item">Favourites</li>
    <li class="list-group-item">My Cloud</li>
    <li class="list-group-item">Network</li>
    <li class="list-group-item">Messages</li>
    <li class="list-group-item">Settings</li>
</ul>
```

list-group & **list-group-item** are the classes in **bootstrap.css**. These predefined classes makes the list as below.



List Grouping with Predefined Glyphicons

`bootstrap.css` comes with so many Glyphicons. Lets make use of them.

```
<div class="list-group">
    <a href="#" class="list-group-item active">
        <span class="glyphicon glyphicon-home"></span> Home <span
class="badge">512</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-heart"></span> Favourites <span
class="badge">45</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-cloud"></span> My Cloud <span
class="badge">9</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-signal"></span> Network <span
class="badge">8</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-star"></span> Important <span
class="badge">32</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-envelope"></span> Messages <span
class="badge">16</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-globe"></span> Notifications <span
class="badge">58</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-time"></span> Reminders <span
class="badge">7</span>
    </a>
```

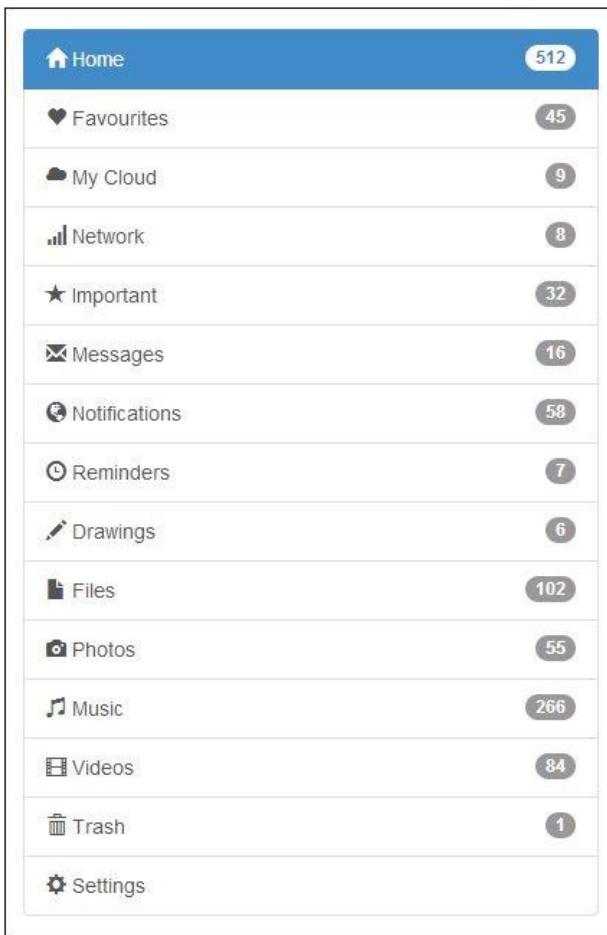
```

        <a href="#" class="list-group-item">
            <span class="glyphicon glyphicon-pencil"></span> Drawings <span
class="badge">6</span>
        </a>
        <a href="#" class="list-group-item">
            <span class="glyphicon glyphicon-file"></span> Files <span
class="badge">102</span>
        </a>
        <a href="#" class="list-group-item">
            <span class="glyphicon glyphicon-camera"></span> Photos <span
class="badge">55</span>
        </a>
        <a href="#" class="list-group-item">
            <span class="glyphicon glyphicon-music"></span> Music <span
class="badge">266</span>
        </a>
        <a href="#" class="list-group-item">
            <span class="glyphicon glyphicon-film"></span> Videos <span
class="badge">84</span>
        </a>
        <a href="#" class="list-group-item">
            <span class="glyphicon glyphicon-trash"></span> Trash <span
class="badge">1</span>
        </a>
        <a href="#" class="list-group-item">
            <span class="glyphicon glyphicon-cog"></span> Settings <span
class="badge"></span>
        </a>
    </div>

```

In the above code if you observe, there are so many class names such as home, heart, cloud, signal, star, envelope etc. These are the classes for their respective icons in Bootstrap.

Now, the output of the above list will be,



The Icons you see in the browser are the predefined icons in Bootstrap. Not only these icons available in Bootstrap. Open the `bootstrap.css` and press `ctrl+f` and type '`Glyphicon`' and see the all Glyphicons of Twitter Bootstrap.

'`badge`' class is used to design the background of numbers on the left side which is also predefined Bootstrap CSS.

Conclusion

In this chapter we have looked into creating beautiful lists in a simple way using Twitter Bootstrap. Hope you understand.

Thanks for reading.

Chapter-6

Forms-I

Introduction

So far we have seen how to create lists with predefined icons using Bootstrap in [Chapter-5](#).

Now let's see the different types of Form layouts in Bootstrap.

Using the code

Forms are the sectional parts of web pages. As we know that we can style the `form` using CSS. Twitter bootstrap comes with predefined CSS styles for different types of Form-Layouts such as Horizontal, Vertical etc.

Creating Horizontal Form using Bootstrap

Just give `form-horizontal` class to your Html `form` tag like below. Below code is the sample example for creating horizontal form.

```
<form class="form-horizontal">
    <div class="form-group">
        <label for="inputFirstName" class="control-label col-xs-2">FirstName</label>
        <div class="col-xs-10">
            <input type="text" class="form-control" id="inputFirstName"
placeholder="First Name">
        </div>
    </div>
    <div class="form-group">
        <label for="inputLastName" class="control-label col-xs-2">LastName</label>
        <div class="col-xs-10">
            <input type="text" class="form-control" id="inputLastName"
placeholder="Last Name">
        </div>
    </div>
    <div class="form-group">
        <label for="inputEmail" class="control-label col-xs-2">Email</label>
        <div class="col-xs-10">
            <input type="email" class="form-control" id="inputEmail"
placeholder="Email">
        </div>
    </div>
    <div class="form-group">
        <label for="inputPassword" class="control-label col-xs-2">Password</label>
        <div class="col-xs-10">
            <input type="password" class="form-control" id="inputPassword"
placeholder="Password">
        </div>
    </div>
    <div class="form-group">
        <div class="col-xs-offset-2 col-xs-10">
            <button type="submit" class="btn btn-primary">Sign Up</button>
        </div>
    </div>
</form>
```

Now Save and Run your project and you will see the following output in your browser

The form consists of four horizontal rows. Each row contains a label on the left and an input field on the right. A blue 'Sign Up' button is located at the bottom.

FirstName	<input placeholder="First Name" type="text"/>
LastName	<input placeholder="Last Name" type="text"/>
Email	<input placeholder="Email" type="text"/>
Password	<input placeholder="Password" type="text"/>

Sign Up

In the above figure if you observe, the elements are in horizontal manner.

Creating Vertical Form using Bootstrap

If you want to make the above alignment as vertical, change your Form style by changing the class of `form` tag.

```
<form>
    <div class="form-group">
        <label for="inputFirstName">FirstName</label>
        <input type="text" class="form-control" id="inputFirstName"
placeholder="First Name">
    </div>
    <div class="form-group">
        <label for="inputLastName">LastName</label>
        <input type="text" class="form-control" id="inputLastName" placeholder="Last
Name">
    </div>
    <div class="form-group">
        <label for="inputEmail">Email</label>
        <input type="email" class="form-control" id="inputEmail" placeholder="Email">
    </div>
    <div class="form-group">
        <label for="inputPassword">Password</label>
        <input type="password" class="form-control" id="inputPassword"
placeholder="Password">
    </div>
    <button type="submit" class="btn btn-primary">Sign Up</button>
</form>
```

In the above example code snippet, we have removed the `form-horizontal` class as by default it will be in Vertical alignment.

Check out the above code in your browser to see the output below,

The figure shows a sign-up form with the following structure:

- FirstName**: Input box labeled "First Name".
- LastName**: Input box labeled "Last Name".
- Email**: Input box labeled "Email".
- Password**: Input box labeled "Password".
- Sign Up**: A blue button at the bottom.

Observe the figure, all elements are aligned vertically.

Creating Inline Form using Bootstrap

Now we will see the Inline Form.

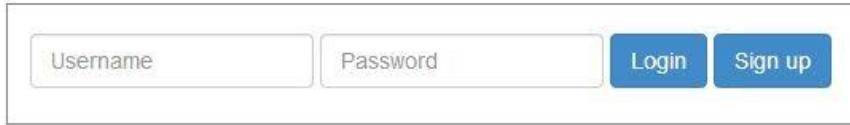
Give the `form-inline` class to your form tag.

```
<form class="form-inline">
    <div class="form-group">
        <label class="sr-only" for="inputUserName">Username</label>
        <input type="text" class="form-control" id="inputUserName"
placeholder="Username">
    </div>
    <div class="form-group">
        <label class="sr-only" for="inputPassword">Password</label>
        <input type="password" class="form-control" id="inputPassword"
placeholder="Password">
    </div>
    <button type="submit" class="btn btn-primary">Login</button>
    <button type="submit" class="btn btn-primary">Sign up</button>
</form>
```

'`sr-only`' is the predefined class in `bootstrap.css` which covers overflow, position properties. `sr-only` looks like below

```
.sr-only {
    position: absolute;
    width: 1px;
    height: 1px;
    padding: 0;
    margin: -1px;
    overflow: hidden;
    clip: rect(0, 0, 0, 0);
    border: 0;
}
```

Now refresh your browser and observe the output



The figure shows a horizontal form layout. It consists of two input fields: 'Username' and 'Password', followed by two buttons: 'Login' and 'Sign up'. All elements are aligned inline within a single row.

In the above figure, all the Html elements are aligned inline.

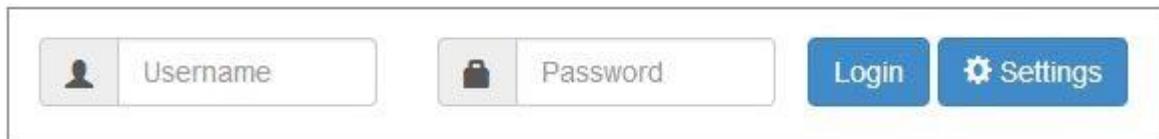
Creating Inline Form with Glyphicons using Bootstrap

We have already seen that there are some Glyphicons in Twitter Bootstrap. Now we are going to use them in here.

See the below example Code snippet of adding icons to the elements.

```
<form>
    <div class="row">
        <div class="col-xs-2">
            <div class="input-group">
                <span class="input-group-addon"><span class="glyphicon glyphicon-user"></span></span>
                    <input type="text" class="form-control" placeholder="Username">
                </div>
            </div>
            <div class="col-xs-2">
                <div class="input-group">
                    <span class="input-group-addon"><span class="glyphicon glyphicon-lock"></span></span>
                        <input type="password" class="form-control" placeholder="Password">
                    </div>
                </div>
                <button type="submit" class="btn btn-primary">Login</button>
                <button type="submit" class="btn btn-primary"><span class="glyphicon glyphicon-cog"></span> Settings</button>
            </div>
        </div>
    </form>
```

The output of the above code will be



The figure shows the same inline form as before, but with added Bootstrap 'input-group' classes. The 'Username' input field now has a user icon to its left. The 'Password' input field now has a lock icon to its left. The 'Settings' button now has a gear icon to its left. The overall look is more modern and integrated.

In the above figure, Glyphicons have made the form more beautiful and nice look and feel.

Changing Height of Inputs and Dropdowns

Following example code shows different sizes of Inputs and Select Boxes in terms of height.

```

<form>
    <div class="row">
        <div class="col-xs-3">
            <input type="text" class="form-control input-lg" placeholder="Input Large">
        </div>
        <div class="col-xs-3">
            <select class="form-control input-lg">
                <option>Large1</option>
                <option>Large2</option>
                <option>Large3</option>
            </select>
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-3">
            <input type="text" class="form-control" placeholder="Input Default">
        </div>
        <div class="col-xs-3">
            <select class="form-control">
                <option>Default1</option>
                <option>Default2</option>
                <option>Default3</option>
            </select>
        </div>
    </div>
    <br>
    <div class="row">
        <div class="col-xs-3">
            <input type="text" class="form-control input-sm" placeholder="Input Small">
        </div>
        <div class="col-xs-3">
            <select class="form-control input-sm">
                <option>Small1</option>
                <option>Small2</option>
                <option>Small3</option>
            </select>
        </div>
    </div>
</form>

```

Run the above code in your browser and you will see different sizes of Inputs like below

<input type="text" value="Input Large"/>	<input style="width: 100px; height: 30px; font-size: 10px; padding: 5px; margin: 5px;" type="text" value="Large1"/>
<input type="text" value="Input Default"/>	<input style="width: 100px; height: 30px; font-size: 10px; padding: 5px; margin: 5px;" type="text" value="Default1"/>
<input type="text" value="Input Small"/>	<input style="width: 100px; height: 30px; font-size: 10px; padding: 5px; margin: 5px;" type="text" value="Small1"/>

`input-lg`, `input-sm` are the predefined classes for larger and smaller inputs respectively.

Changing Width of Inputs and Dropdowns

Below example shows different sizes of Inputs and Select Boxes in terms of width.

```
<form>
  <div class="row">
    <div class="col-xs-1">
      <textarea class="form-control"></textarea>
    </div>
    <div class="col-xs-2">
      <textarea class="form-control"></textarea>
    </div>
    <div class="col-xs-3">
      <textarea class="form-control"></textarea>
    </div>
    <div class="col-xs-4">
      <textarea class="form-control"></textarea>
    </div>
  </div>
  <br>
  <div class="row">
    <div class="col-xs-1">
      <select class="form-control">
        <option>Option1</option>
        <option>Option2</option>
      </select>
    </div>
    <div class="col-xs-2">
      <select class="form-control">
        <option>Option1</option>
        <option>Option2</option>
      </select>
    </div>
    <div class="col-xs-3">
      <select class="form-control">
        <option>Option1</option>
        <option>Option2</option>
      </select>
    </div>
    <div class="col-xs-4">
      <select class="form-control">
        <option>Option1</option>
        <option>Option2</option>
      </select>
    </div>
  </div>
</form>
```

Now Save and Run the above example to see textareas and dropdowns with different width.



You can observe that width of elements is changed in your browser because of `col-xs-1`(column extra small), `col-xs-2`, `col-xs-3` and `col-xs-4` classes of Bootstrap.

Conclusion

In this chapter we have seen different types of Form-Layouts of using Bootstrap and also seen changing height and width of Html elements. Hope you understand.

We will see more Bootstrap features on forms in next chapter.

Thanks for reading.

Chapter-7

Forms-II

Introduction

Well, So far we have seen different types of forms in Bootstrap. Let's see some more of them in this chapter.

Using the code

Grouping elements such as dropdowns, checkboxes, radio buttons, buttons etc. makes the webpage nice look and feel. Let's discuss one by one,

Grouping Inputs with Dropdown menu using Bootstrap

Dropdown menu gives more beauty to a webpage. Now let's try to combine the `textbox` with `dropdown` menu using predefined styles of `bootstrap.css`.

Below is the example code for that scenario,

```
<form>
  <div class="row">
    <div class="col-xs-3">
      <div class="input-group">
        <div class="input-group-btn">
          <button type="button" class="btn btn-primary dropdown-toggle" data-
toggle="dropdown">Home <span class="caret"></span></button>
          <ul class="dropdown-menu">
            <li><a href="#">Photos</a></li>
            <li><a href="#">Videos</a></li>
            <li><a href="#">Music</a></li>
            <li class="divider"></li>
            <li><a href="#">Games</a></li>
            <li><a href="#">Drawings</a></li>
            <li><a href="#">Charts</a></li>
          </ul>
        </div>
        <input type="text" class="form-control">
      </div>
    </div>
    <div class="col-xs-3">
      <div class="input-group">
        <input type="text" class="form-control">
        <div class="input-group-btn">
          <button type="button" class="btn btn-primary dropdown-toggle" data-
toggle="dropdown">Mobile <span class="caret"></span></button>
          <ul class="dropdown-menu pull-right">
            <li><a href="#">Calls</a></li>
            <li><a href="#">Messages</a></li>
            <li><a href="#">Email</a></li>
            <li class="divider"></li>
            <li><a href="#">Games</a></li>
            <li><a href="#">Music</a></li>
            <li><a href="#">Gallery</a></li>
          </ul>
        </div>
      </div>
    </div>
  </div>
</form>
```

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

If you observe the above code, there are some class files named `input-group`, `input-group-btn` which groups the elements. `dropdown-toggle` class is for toggling the menu click. `caret` is for the dropdown icon.

Now run the above code in your browser and you will see the following output

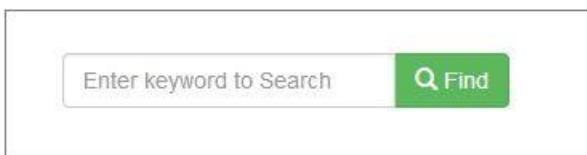


Grouping Input with Search Button

Using Bootstrap, we can easily create a beautiful Search box with Button grouped to it.

```
<form>
  <div class="row">
    <div class="col-xs-7">
      <div class="input-group">
        <input type="text" class="form-control" placeholder="Enter keyword to
Search">
        <span class="input-group-btn">
          <button type="button" class="btn btn-success"><span
class="glyphicon glyphicon-search"></span> Find</button>
        </span>
      </div>
    </div>
  </div>
</form>
```

In the Above example, we have used Grouping classes of Bootstrap. Output of the above code will be,



Thats it !!, Its pretty easy to create an awesome search box like above in Bootstrap.

Grouping RadioButton and CheckBox to the Inputs

Now lets see grouping **Checkbox** and a **Radiobutton** to the textbox.

```
<form>
  <div class="row">
    <div class="col-xs-5">
      <div class="input-group">
        <input type="text" class="form-control">
        <div class="input-group-addon">
          <input type="checkbox">
        </div>
      </div>
    </div>
    <div class="col-xs-5">
      <div class="input-group">
        <input type="text" class="form-control">
        <div class="input-group-addon">
          <input type="radio">
        </div>
      </div>
    </div>
  </div>
</form>
```

Try the above example and run it in your browser. You will see the following output.



Creating a disabled Input

In some cases there is a need of input that has to be disabled in certain conditions.

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

As we already know that setting **disabled** property of **input** to 'disabled' will makes our input to be disabled.

Output of the above code will be,



Creating Validation Forms using Bootstrap

Validation to the inputs is necessary to any web application to ensure good database without duplicates and also for high security.

Below example shows an overview of validations for different type of inputs with Bootstrap.

```
<form class="form-horizontal">
    <div class="form-group has-success">
        <label class="col-xs-2 control-label" for="inputSuccess">Username</label>
        <div class="col-xs-8">
            <div class="input-group">
                <input type="text" id="inputSuccess" class="form-control"
placeholder="Good Input">
                <div class="input-group-btn">
                    <button class="btn btn-primary">Check Availability</button>
                </div>
            </div>
            <span class="help-block">This name is available</span>
        </div>
    </div>
    <div class="form-group has-warning">
        <label class="col-xs-2 control-label" for="inputWarning">Password</label>
        <div class="col-xs-10">
            <input type="password" id="inputWarning" class="form-control"
placeholder="Warning Input">
            <span class="help-block">Choose Strong Password</span>
        </div>
        <label class="col-xs-2 control-label" for="inputWarning">Confirm
Password</label>
        <div class="col-xs-10">
            <input type="password" id="inputWarning" class="form-control"
placeholder="Warning Input">
            <span class="help-block">Choose Strong Password</span>
        </div>
    </div>
    <div class="form-group has-error">
        <label class="col-xs-2 control-label" for="inputError">Email</label>
        <div class="col-xs-10">
            <input type="email" id="inputError" class="form-control"
placeholder="Error Input">
            <span class="help-block">Please enter a valid email address</span>
        </div>
    </div>
    <button class="btn btn-danger">Back</button>
    <button class="btn btn-success">Register</button>
    <button class="btn btn-warning">Reset Fields</button>
</form>
```

warning, error, success, danger etc. are the colorful themes which we can find in **bootstrap.css**. These predefined styles beautifies the Html elements.

Now Run the above code in your browser and you will see the Validation forms as below

Username Check Availability
This name is available

Password
Choose Strong Password

Confirm Password
Choose Strong Password

Email
Please enter a valid email address

Back Register Reset Fields

Conclusion

In this chapter, we have seen different types of grouping elements with Inputs and Form Validation structure using Bootstrap. Hope you understand.

Thanks for reading.

Chapter-8

Images

Introduction

Placing Images in ordered manner, Grouping them with some descriptions and elements will make them looking good in our webpage. Now let's discuss them in this chapter.

Using Bootstrap code

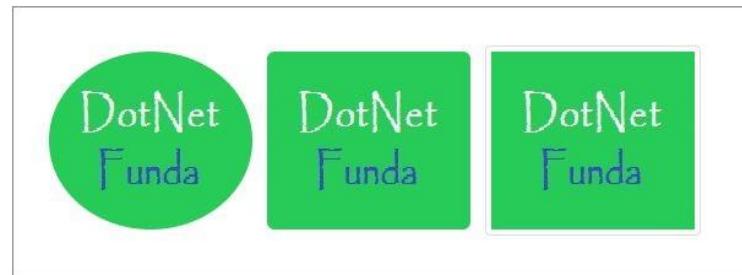
It is necessary to decorate the images to make them look nice in the web applications. Twitter Bootstrap helps in making such type of decorations easily.

Creating stylish Images

There are some predefined `img` classes in `bootstrap.css` such as `img-circle`(Sets circle frame to the Image), `img-rounded`(for curved borders) and `img-thumbnail`(for thumbnail images).

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

The output of the above example code snippet is,



Creating Image Thumbnails

Using `img-thumbnail` class, we can create image thumbnails in a tabular manner. Take a look on the below example code in which we have 8 images.

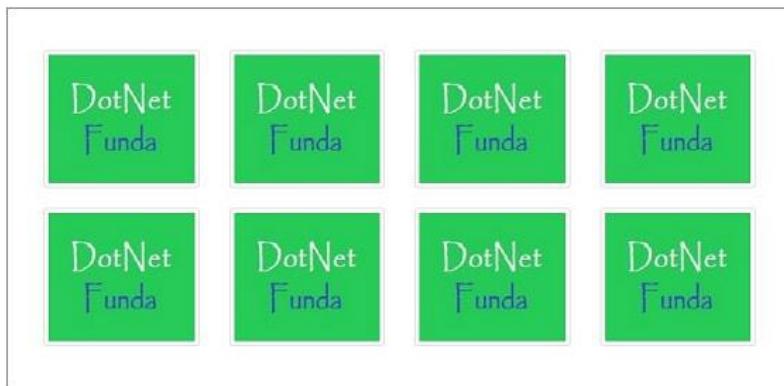
```
<div class="container">
    <div class="row">
        <div class="col-xs-3">
            <a href="#" class="thumbnail">
                
            </a>
        </div>
        <div class="col-xs-3">
            <a href="#" class="thumbnail">
                
            </a>
        </div>
```

```

<div class="col-xs-3">
    <a href="#" class="thumbnail">
        
    </a>
</div>
</div>
</div>

```

Run the above code in your browser and you will see the following arrange of thumbnails as below,



Creating Images with Description

Creating Images with some description is very helpful in some cases. Observe the below example containing `media` classes of Bootstrap.

```

<div class="container">
    <div class="col-xs-6">
        <div class="media">
            <a href="#" class="pull-left">
                
            </a>
            <div class="media-body">
                <h4 class="media-heading">DotNetFunda <small><i>Fundamentals of
.Net</i></small></h4>
                <p>DotNetFunda is the website for those who are looking for .Net Tips
and Tricks.</p>
            </div>
        </div>
        <div class="media">
            <a href="#" class="pull-left">
                
            </a>
            <div class="media-body">
                <h4 class="media-heading">IT Funda <small><i>IT Training &
Placement</i></small></h4>
                <p>IT Funda is the corporative website for those who need IT Training
and for those who are looking to buy IT Stuff.</p>
            </div>
        </div>
        <div class="media">
            <a href="#" class="pull-left">
                
            </a>
            <div class="media-body">
                <h4 class="media-heading">KidsFundा <small><i>Learn with Fun for
Kids</i></small></h4>
                <p>Kids Funda is designed for Kids to make them learn easily with
some fun.</p>
            </div>
        </div>
    </div>
</div>

```

Run and see the above code in your browser and you will get nice image lists like below,

 DotNetFundा <i>Fundamentals of .Net</i> DotNetFundा is the website for those who are looking for .Net Tips and Tricks.	 IT Funda <i>IT Training & Placement</i> IT Funda is the corporative website for those who need IT Training and for those who are looking to buy IT Stuff.	 KidsFundा <i>Learn with Fun for Kids</i> Kids Funda is designed for Kids to make them learn easily with some fun.
---	--	--

How nice it is looking Isn't it !!

Creating Images with some options

Now lets see images with some buttons and description as well. Observe the below example

```
<div class="container">
    <div class="row">
        <div class="col-xs-2">
            <div class="thumbnail">
                
                <div class="caption">
                    <h4>DotNetFunda</h4>
                    <p>Description.....</p>
                    <p><a href="#" class="btn btn-primary">Download</a> <a href="#" class="btn btn-warning"><span class="glyphicon glyphicon-share"></span>&ampnbsp</a></p>
                </div>
            </div>
        </div>
        <div class="col-xs-2">
            <div class="thumbnail">
                
                <div class="caption">
                    <h4>ITFunda</h4>
                    <p>Description.....</p>
                    <p><a href="#" class="btn btn-primary">Download</a> <a href="#" class="btn btn-warning"><span class="glyphicon glyphicon-share"></span>&ampnbsp</a></p>
                </div>
            </div>
        </div>
        <div class="col-xs-2">
            <div class="thumbnail">
                
                <div class="caption">
                    <h4>KidsFunda</h4>
                    <p>Description.....</p>
                    <p><a href="#" class="btn btn-primary">Download</a> <a href="#" class="btn btn-warning"><span class="glyphicon glyphicon-share"></span>&ampnbsp</a></p>
                </div>
            </div>
        </div>
    </div>
</div>
```

Output of the above code will gives Images with some options to operate,



Creating Nested Images with Description

Similar to above Descriptive Images, we can create Nested Descriptive Images by writing Nested code like below,

```
<ul class="media-list">
    <li class="media">
        <a href="#" class="pull-left">
            
        </a>
        <div class="media-body">
            <h4 class="media-heading">DotNetFunda</h4>
            <p>DotNetFunda is the website for those who are looking for .Net Tips and
Tricks.</p>
            <!-- Nested media object -->
            <div class="media">
                <a href="#" class="pull-left">
                    
                </a>
                <div class="media-body">
                    <h4 class="media-heading">IT Funda(Nested)</h4>
                    <p>IT Funda is the corporative website for those who need IT
Training and for those who are looking to buy IT Stuff.</p>
                    <!-- Chain-Nested media object -->
                    <div class="media">
                        <a href="#" class="pull-left">
                            
                        </a>
                        <div class="media-body">
                            <h4 class="media-heading">Kids Funda(Nested)</h4>
                            <p>Kids Funda is designed for Kids to make them learn
easily with some fun.</p>
                        </div>
                    </div>
                </div>
            </div>
            <!-- Nested media object -->
            <div class="media">
                <a href="#" class="pull-left">
                    
                </a>
                <div class="media-body">
                    <h4 class="media-heading">IT Funda(Nested)</h4>
                    <p>IT Funda is the corporative website for those who need IT
Training and for those who are looking to buy IT Stuff.</p>
                    <!-- Chain-Nested media object -->
                    <div class="media">
                        <a href="#" class="pull-left">
                            
                        </a>
                        <div class="media-body">
                            <h4 class="media-heading">Kids Funda(Nested)</h4>
                            <p>Kids Funda is designed for Kids to make them learn
```

```

        easily with some fun.</p>
                </div>
            </div>
        </div>
    </div>
</li>
<li class="media">
    <a href="#" class="pull-left">
        
    </a>
    <div class="media-body">
        <h4 class="media-heading">DotNetFunda</h4>
        <p>DotNetFunda is the website for those who are looking for .Net Tips and Tricks.</p>
    </div>
</li>
</ul>

```

The output of the above code gives Nested elements with description like below,

DotNet Funda
DotNetFunda is the website for those who are looking for .Net Tips and Tricks.

IT Funda
IT Funda is the corporative website for those who need IT Training and for those who are looking to buy IT Stuff.

IT Funda(Nested)
IT Funda is the corporative website for those who need IT Training and for those who are looking to buy IT Stuff.

Kids Funda(Nested)
Kids Funda is designed for Kids to make them learn easily with some fun.

DotNet Funda
DotNetFunda is the website for those who are looking for .Net Tips and Tricks.

Conclusion

In this chapter we have seen working with different types of Images using Twitter Bootstrap ([Get Here](#)). Hope you understand.

Thanks for reading.

Chapter-9

Tabs and Pills-I

Introduction

Menus are like a heart of the webpage which makes user easy to navigate particular sections. Bootstrap comes with **Navs** and **Navbars** to create menus.

Using Bootstrap code

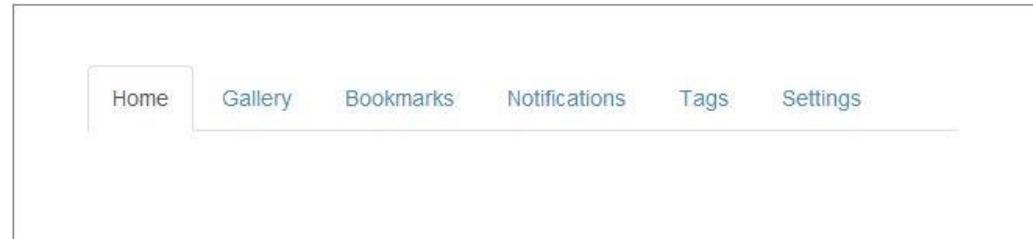
Including Tab's in the web application gives more flexibility to the user to access different pages of that website. Bootstrap have some predefined styles to create beautiful menus. Let's see one by one,

Creating Tabs & Pills

We have to use **nav-tabs** class to create tabs in Bootstrap. Have a look at the below example,

```
<ul class="nav nav-tabs">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Gallery</a></li>
    <li><a href="#">Bookmarks</a></li>
    <li><a href="#">Notifications</a></li>
    <li><a href="#">Tags</a></li>
    <li><a href="#">Settings</a></li>
</ul>
```

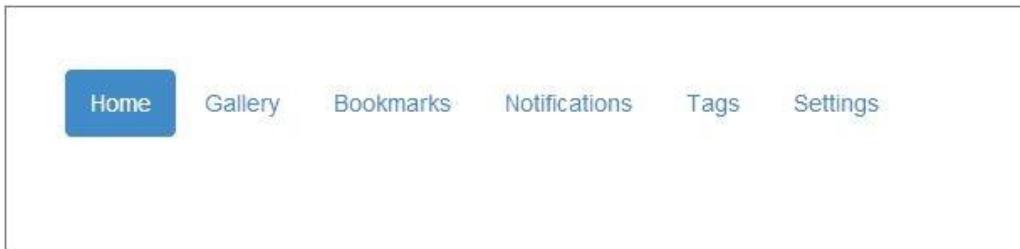
The output of the above code will be,



Now change the class of **ul** in the above code to **nav-pills** i.e.

```
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">Gallery</a></li>
    <li><a href="#">Bookmarks</a></li>
    <li><a href="#">Notifications</a></li>
    <li><a href="#">Tags</a></li>
    <li><a href="#">Settings</a></li>
</ul>
```

Look the change in your browser. It will be like,



Creating Tabs & Pills with Icons

As we know that, there are so many icons in Bootstrap. Lets try them in Tabs and Pills. Observe the below code,

```
<ul class="nav nav-tabs">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-home">/</span> Home</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-film">/</span> Gallery</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-bookmark">/</span>
Bookmarks</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-globe">/</span>
Notifications</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-tags">/</span> Tags</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-cog">/</span> Settings</a></li>
</ul>
```

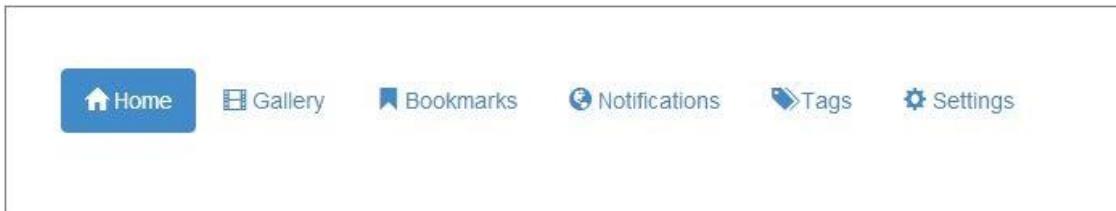
In the above code if you observe, we just added Glyphicons to the elements. Now see the output of this code. It seems like,



Now change the class of `ul` in the above code to `nav-pills` as below,

```
<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-film">/</span> Gallery</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-bookmark">/</span>
Bookmarks</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-globe">/</span>
Notifications</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-tags">/</span> Tags</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-cog">/</span> Settings</a></li>
</ul>
```

If you run the above code snip in your browser, you will get following output



Creating stacked Pills and Tabs

How about stacked tabs and pills?.. Bootstrap contains `nav-stacked` class for that. Add this class to the above code as below,

```
<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-film"></span> Gallery</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-bookmark"></span> Bookmarks</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-globe"></span> Notifications</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-tags"></span> Tags</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>
</ul>
```

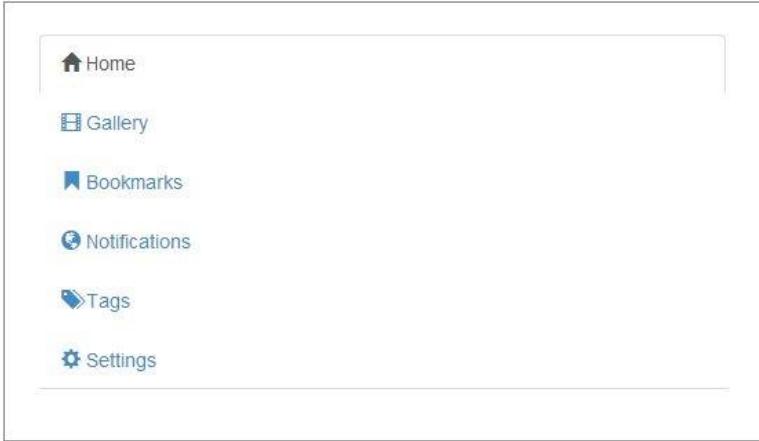
Output of the above snip will be,



For `nav-tabs` i.e.

```
<ul class="nav nav-tabs nav-stacked">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-film"></span> Gallery</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-bookmark"></span> Bookmarks</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-globe"></span> Notifications</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-tags"></span> Tags</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>
</ul>
```

Output will become,



Conclusion

In this chapter, we discussed on simple Tabs and Pills that are used to create menu bars in the Web applications. Hope you understand.

Please read the continuation of this topic in next chapter.

Thanks for reading.

Chapter-10

Tabs and Pills-II

Introduction

This is the continuation of previous chapter [Chapter-9](#). We have discussed simple Tabs and Pills in the previous chapter, now let's see dropdowns with them.

Using Bootstrap code

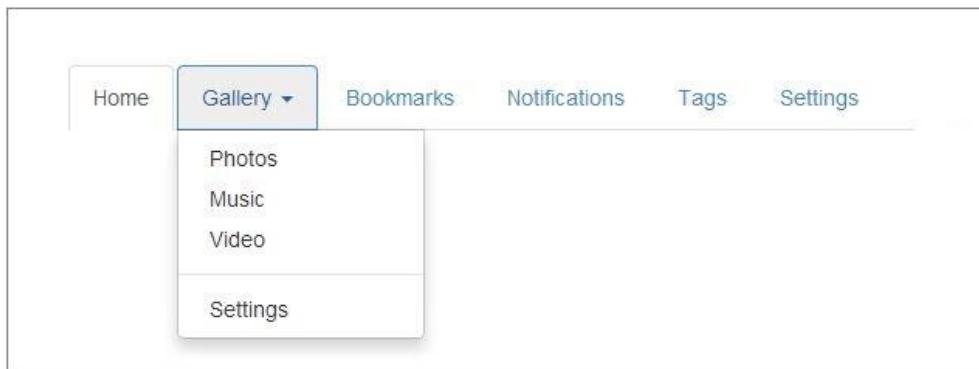
Tabs and Pills with Dropdown menus makes good attraction and easy surfing to the users. Let's see them briefly below

Creating Tabs and Pills with Dropdowns

Setting Dropdown for Tab or Pill elements is quite easy in Bootstrap. Let's see the below code snippet to make yourself easy to understand.

```
<ul class="nav nav-tabs">
    <li class="active"><a href="#">Home</a></li>
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-toggle">Gallery <b
class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#">Photos</a></li>
            <li><a href="#">Music</a></li>
            <li><a href="#">Video</a></li>
            <li class="divider"></li>
            <li><a href="#">Settings</a></li>
        </ul>
    </li>
    <li><a href="#">Bookmarks</a></li>
    <li><a href="#">Notifications</a></li>
    <li><a href="#">Tags</a></li>
    <li><a href="#">Settings</a></li>
</ul>
```

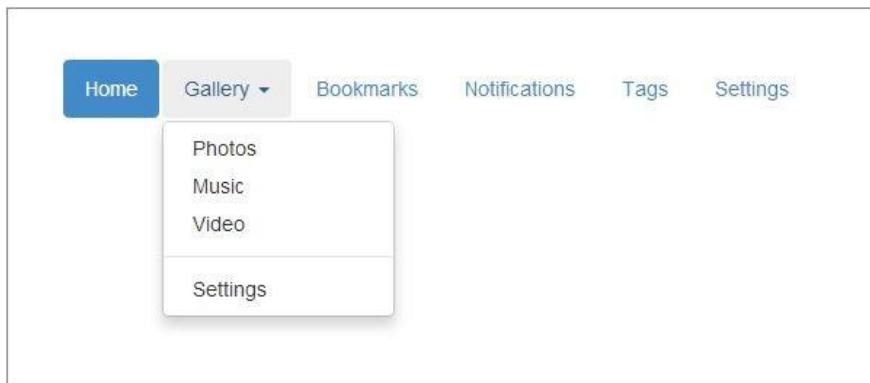
Run the above code in your browser and you will see the following output,



Now, change the class of `ul` in the above code to `nav-pills` like below.

```
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home</a></li>
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-toggle">Gallery <b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#">Photos</a></li>
            <li><a href="#">Music</a></li>
            <li><a href="#">Video</a></li>
            <li class="divider"></li>
            <li><a href="#">Settings</a></li>
        </ul>
    </li>
    <li><a href="#">Bookmarks</a></li>
    <li><a href="#">Notifications</a></li>
    <li><a href="#">Tags</a></li>
    <li><a href="#">Settings</a></li>
</ul>
```

Output of the above code will give the Pill nav with dropdown elements



Creating Tabs and Pills with Icons and Dropdowns

Let's set some Bootstrap Icons to the elements. Below example explains how to set Icons to each element.

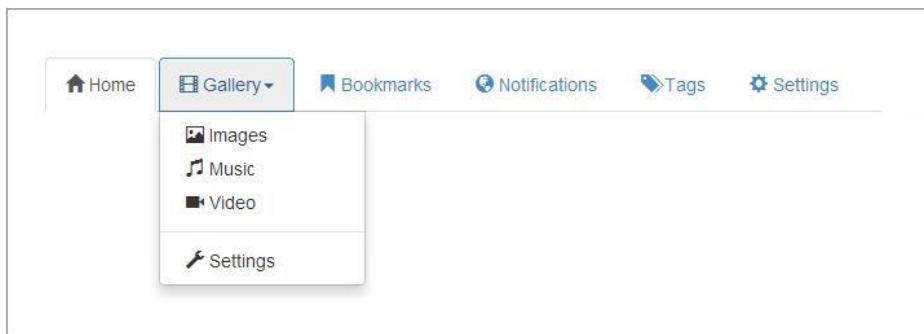
```
<ul class="nav nav-tabs">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-toggle"><span class="glyphicon glyphicon-film"></span> Gallery <b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#"><span class="glyphicon glyphicon-picture"></span> Images</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-music"></span> Music</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-facetime-video"></span> Video</a></li>
        </ul>
    </li>
</ul>
```

```

        <li class="divider"></li>
        <li><a href="#"><span class="glyphicon glyphicon-wrench"></span>
Settings</a></li>
        </ul>
    </li>
    <li><a href="#"><span class="glyphicon glyphicon-bookmark"></span>
Bookmarks</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-globe"></span>
Notifications</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-tags"></span> Tags</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>
</ul>

```

In the above code, we have used Glyphicons of Twitter Bootstrap to the Nav elements.
Output of the above snip will be,



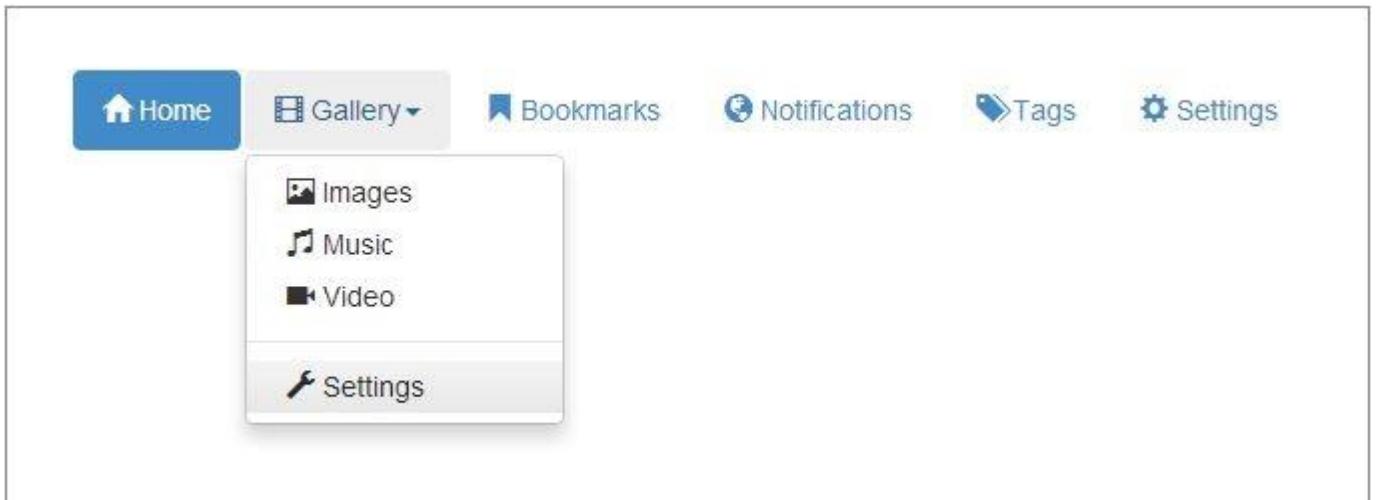
For the Pills type Nav (change the `ul` class),

```

<ul class="nav nav-pills">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
        <li class="dropdown">
            <a href="#" data-toggle="dropdown" class="dropdown-toggle"><span class="glyphicon glyphicon-film"></span> Gallery<b class="caret"></b></a>
            <ul class="dropdown-menu">
                <li><a href="#"><span class="glyphicon glyphicon-picture"></span> Images</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-music"></span> Music</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-facetime-video"></span> Video</a></li>
                <li class="divider"></li>
                <li><a href="#"><span class="glyphicon glyphicon-wrench"></span> Settings</a></li>
            </ul>
        </li>
        <li><a href="#"><span class="glyphicon glyphicon-bookmark"></span> Bookmarks</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-globe"></span> Notifications</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-tags"></span> Tags</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>
            </ul>
        </li>

```

Check out the above code in your browser and you will find the following result



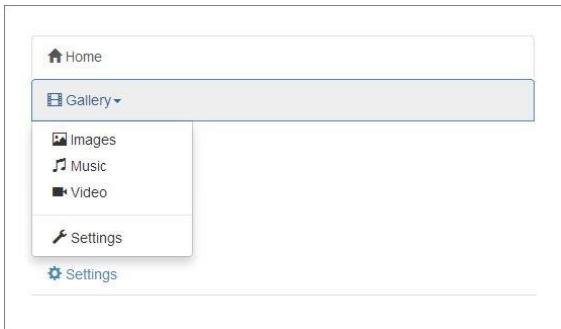
Creating Stacked Tabs and Pills with Icons and Dropdowns

Adding `nav-stacked` to the `ul` class will do the job.

For Stacked Tab, See the below example

```
<ul class="nav nav-tabs nav-stacked">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
        <li class="dropdown">
            <a href="#" data-toggle="dropdown" class="dropdown-toggle"><span class="glyphicon glyphicon-film"></span> Gallery<b class="caret"></b></a>
            <ul class="dropdown-menu">
                <li><a href="#"><span class="glyphicon glyphicon-picture"></span> Images</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-music"></span> Music</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-facetime-video"></span> Video</a></li>
                <li class="divider"></li>
                <li><a href="#"><span class="glyphicon glyphicon-wrench"></span> Settings</a></li>
            </ul>
        </li>
        <li><a href="#"><span class="glyphicon glyphicon-bookmark"></span> Bookmarks</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-globe"></span> Notifications</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-tags"></span> Tags</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>
            </ul>
        </li>
    </ul>
```

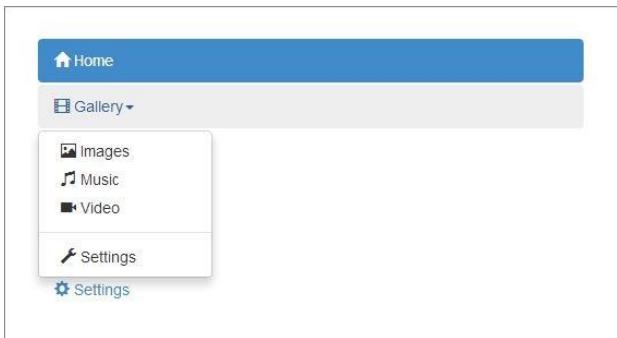
Output of the above code will be,



For Stacked Pills, Go through below code (changing `ul` class)

```
<ul class="nav nav-pills nav-stacked">
    <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
    <li class="dropdown">
        <a href="#" data-toggle="dropdown" class="dropdown-toggle"><span class="glyphicon glyphicon-film"></span> Gallery<b class="caret"></b></a>
        <ul class="dropdown-menu">
            <li><a href="#"><span class="glyphicon glyphicon-picture"></span> Images</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-music"></span> Music</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-facetime-video"></span> Video</a></li>
            <li class="divider"></li>
            <li><a href="#"><span class="glyphicon glyphicon-wrench"></span> Settings</a></li>
        </ul>
    </li>
    <li><a href="#"><span class="glyphicon glyphicon-bookmark"></span> Bookmarks</a></li>
    <li><a href="#"><span class="glyphicon glyphicon-globe"></span> Notifications</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-tags"></span> Tags</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-cog"></span> Settings</a></li>
    </ul>
```

Output of the above code will be,



Disabling links in the Nav components

Inorder to disable links of any **Nav** components, Bootstrap provides **disabled** class. Check the below example,

```
<ul class="nav nav-pills">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">User</a></li>
    <li><a href="#">Settings</a></li>
    <li class="disabled"><a href="#">Admin</a></li>
</ul>
```

In the above code, we have given **disabled** class to the Admin element. See the code in the browser



Conclusion

In this chapter, we have seen different Tabs and Pills of Nav elements with dropdowns as well as icons and also disabling elements, stacked Navs. Hope you understand.

Thanks for reading.

Chapter-11

Menus

Introduction

Menus are like a heart of the webpage which makes user easy to navigate particular sections. Bootstrap comes with **Navs** and **Navbars** to create menus.

It is very simple to create menus using **Navbars** in Bootstrap.

Using Bootstrap code

Bootstrap contains so many **navbar** classes which are used to create different types of Menus. You can find them in **bootstrap.css**.

Creating normal menu using Navbars

Observe the below Example,

```
<nav role="navigation" class="navbar navbar-default">
    <!-- Grouping Brand with Toggle for better mobile display -->
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="#" class="navbar-brand">Brand</a>
    </div>
    <!-- Next nav links in the Navbar -->
    <div id="navbarCollapse" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
            <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-certificate"></span> DotNetFunda</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-book"></span> ITFunda</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-gift"></span> KidsFunda</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-film"></span> FundooVideo</a></li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
            <li><a href="#">Login</a></li>
            <li><a href="#">SignUp</a></li>
        </ul>
    </div>
</nav>
```

In the above code, we have used **navbar** predefined classes such as **navbar-default**, **navbar-header**, **navbar-collapse** etc. of Bootstrap.

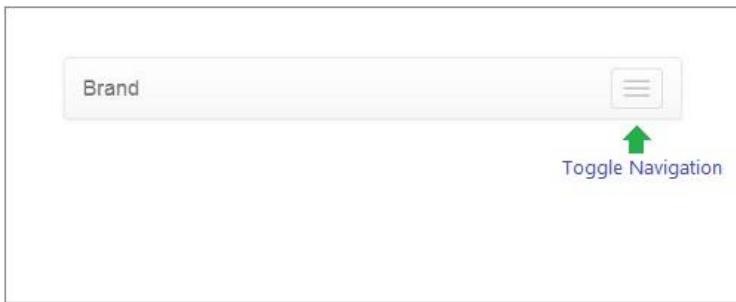
Go to `bootstrap.css` and take a look at those styles.

In the above code we also grouped Glyphicons to the Menu elements to make them attractive.

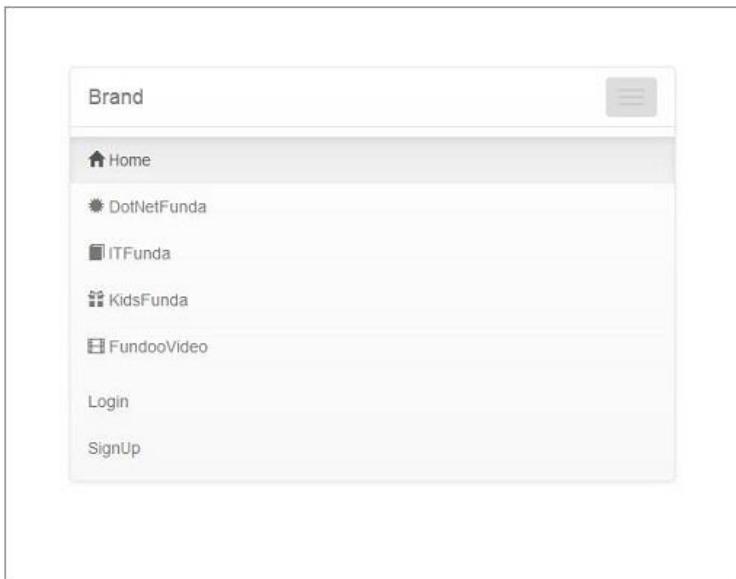
Run the above code in your browser and you will see the following output,



If you reduce the screen size (Means mobile layout), the above code will give following output,



Click the `Toggle Navigation` button to expand your Menu like below



Now change `navbar-default` to `navbar-inverse` and see the magic.

```
<nav role="navigation" class="navbar navbar-inverse">
    <!-- Grouping Brand with Toggle for better mobile display -->
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="#" class="navbar-brand">Brand</a>
    </div>
    <!-- Next nav links in the Navbar -->
    <div id="navbarCollapse" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
            <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-certificate"></span> DotNetFunda</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-book"></span> ITFunda</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-gift"></span> KidsFunda</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-film"></span> FundooVideo</a></li>
        </ul>
        <ul class="nav navbar-nav navbar-right">
            <li><a href="#">Login</a></li>
            <li><a href="#">SignUp</a></li>
        </ul>
    </div>
</nav>
```

Above change of `nav` class will result as below,



Creating Full Menu with Search box and Dropdowns

Now lets create a complete menu with Dropdowns as well as Search box. Take a look at the below code,

```
<nav role="navigation" class="navbar navbar-inverse">
    <!-- Grouping Brand with Toggle for better mobile display -->
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
```

```

        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
    </button>
    <a href="#" class="navbar-brand">Brand</a>
</div>
<!-- Next nav links in the Navbar --&gt;
&lt;div id="navbarCollapse" class="collapse navbar-collapse"&gt;
    &lt;ul class="nav navbar-nav"&gt;
        &lt;li class="active"&gt;&lt;a href="#"&gt;&lt;span class="glyphicon glyphicon-home"&gt;&lt;/span&gt; Home&lt;/a&gt;&lt;/li&gt;
        &lt;li class="dropdown"&gt;
            &lt;a data-toggle="dropdown" class="dropdown-toggle" href="#"&gt;&lt;span class="glyphicon glyphicon-certificate"&gt;&lt;/span&gt; DotNetFunda &lt;b class="caret"&gt;&lt;/b&gt;&lt;/a&gt;
            &lt;ul role="menu" class="dropdown-menu"&gt;
                &lt;li&gt;&lt;a href="#"&gt;Articles&lt;/a&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;Interviews&lt;/a&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;Forums&lt;/a&gt;&lt;/li&gt;
                &lt;li class="divider"&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;About Us&lt;/a&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;Contact Us&lt;/a&gt;&lt;/li&gt;
            &lt;/ul&gt;
        &lt;/li&gt;
        &lt;li class="dropdown"&gt;
            &lt;a data-toggle="dropdown" class="dropdown-toggle" href="#"&gt;&lt;span class="glyphicon glyphicon-book"&gt;&lt;/span&gt; ITFundamentals &lt;b class="caret"&gt;&lt;/b&gt;&lt;/a&gt;
            &lt;ul role="menu" class="dropdown-menu"&gt;
                &lt;li&gt;&lt;a href="#"&gt;Training&lt;/a&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;Placement&lt;/a&gt;&lt;/li&gt;
                &lt;li class="divider"&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;FAQ's&lt;/a&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;Write Us&lt;/a&gt;&lt;/li&gt;
            &lt;/ul&gt;
        &lt;/li&gt;
        &lt;li&gt;&lt;a href="#"&gt;&lt;span class="glyphicon glyphicon-gift"&gt;&lt;/span&gt; KidsFundamentals&lt;/a&gt;&lt;/li&gt;
        &lt;li&gt;&lt;a href="#"&gt;&lt;span class="glyphicon glyphicon-film"&gt;&lt;/span&gt; FundooVideo&lt;/a&gt;&lt;/li&gt;
    &lt;/ul&gt;
    &lt;form role="search" class="navbar-form navbar-left"&gt;
        &lt;div class="form-group"&gt;
            &lt;input type="text" placeholder="Search" class="form-control"&gt;
        &lt;/div&gt;
    &lt;/form&gt;
    &lt;ul class="nav navbar-nav navbar-right"&gt;
        &lt;li&gt;&lt;a href="#"&gt;Contact&lt;/a&gt;&lt;/li&gt;
        &lt;li class="dropdown"&gt;
            &lt;a data-toggle="dropdown" class="dropdown-toggle" href="#"&gt;&lt;span class="glyphicon glyphicon-user"&gt;&lt;/span&gt; Krishna &lt;b class="caret"&gt;&lt;/b&gt;&lt;/a&gt;
            &lt;ul role="menu" class="dropdown-menu"&gt;
                &lt;li&gt;&lt;a href="#"&gt;&lt;span class="glyphicon glyphicon-phone"&gt;&lt;/span&gt; Friends&lt;/a&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;&lt;span class="glyphicon glyphicon-envelope"&gt;&lt;/span&gt; Messages&lt;/a&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;&lt;span class="glyphicon glyphicon-globe"&gt;&lt;/span&gt; Notifications&lt;/a&gt;&lt;/li&gt;
                &lt;li class="divider"&gt;&lt;/li&gt;
                &lt;li&gt;&lt;a href="#"&gt;&lt;span class="glyphicon glyphicon-cog"&gt;&lt;/span&gt; &lt;/li&gt;
            &lt;/ul&gt;
        &lt;/li&gt;
    &lt;/ul&gt;
&lt;/div&gt;
</pre>

```

```

Settings</a></li>
        <li><a href="#"><span class="glyphicon glyphicon-off"></span>
Logout</a></li>
    </ul>
</li>
</ul>
</div>
</nav>

```

Dont get confused with above code, we are keeping more elements and hence it is little bigger.

The output of the above code will gives us a beautiful menu bar as shown below,

A screenshot of a website's header menu. The menu items include 'Brand', 'Home', 'DotNetFunda' (which has a dropdown menu), 'ITFundu', 'KidsFunda', 'FundooVideo', a search bar, 'Contact', and a user profile for 'Krishna'. The 'DotNetFunda' dropdown menu contains links for 'Articles', 'Interviews', 'Forums', 'About Us', and 'Contact Us'.

A screenshot of a website's header menu. The menu items include 'Brand', 'Home', 'DotNetFunda', 'ITFundu' (which has a dropdown menu), 'KidsFunda', 'FundooVideo', a search bar, 'Contact', and a user profile for 'Krishna'. The 'ITFundu' dropdown menu contains links for 'Training' and 'Placement'.

A screenshot of a website's header menu. The menu items include 'Brand', 'Home', 'DotNetFunda', 'ITFundu', 'KidsFunda', 'FundooVideo', a search bar, 'Contact', and a user profile for 'Krishna' (which has a dropdown menu). The 'Krishna' dropdown menu contains links for 'Friends', 'Messages', 'Notifications', 'Settings', and 'Logout'.

Making Menus fixed to the Layout

In order to make the Menu bar fixed, we have to add `navbar-fixed-top` (to fix it to the top) or `navbar-fixed-bottom` (to fix to the bottom).

See the below code of Fixing Menu bar to the Top,

```
<nav role="navigation" class="navbar navbar-inverse navbar-fixed-top">
    <!-- Grouping Brand with Toggle for better mobile display -->
    <div class="navbar-header">
        <button type="button" data-target="#navbarCollapse" data-toggle="collapse" class="navbar-toggle">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
        </button>
        <a href="#" class="navbar-brand">Brand</a>
    </div>
    <!-- Next nav links in the Navbar -->
    <div id="navbarCollapse" class="collapse navbar-collapse">
        <ul class="nav navbar-nav">
            <li class="active"><a href="#"><span class="glyphicon glyphicon-home"></span> Home</a></li>
            <li class="dropdown">
                <a data-toggle="dropdown" class="dropdown-toggle" href="#"><span class="glyphicon glyphicon-certificate"></span> DotNetFunda <b class="caret"></b></a>
                <ul role="menu" class="dropdown-menu">
                    <li><a href="#">Articles</a></li>
                    <li><a href="#">Interviews</a></li>
                    <li><a href="#">Forums</a></li>
                    <li class="divider"></li>
                    <li><a href="#">About Us</a></li>
                    <li><a href="#">Contact Us</a></li>
                </ul>
            </li>
            <li class="dropdown">
                <a data-toggle="dropdown" class="dropdown-toggle" href="#"><span class="glyphicon glyphicon-book"></span> ITFunda <b class="caret"></b></a>
                <ul role="menu" class="dropdown-menu">
                    <li><a href="#">Training</a></li>
                    <li><a href="#">Placement</a></li>
                    <li class="divider"></li>
                    <li><a href="#">FAQ's</a></li>
                    <li><a href="#">Write Us</a></li>
                </ul>
            </li>
            <li><a href="#"><span class="glyphicon glyphicon-gift"></span> KidsFunda</a></li>
            <li><a href="#"><span class="glyphicon glyphicon-film"></span> FundooVideo</a></li>
        </ul>
        <form role="search" class="navbar-form navbar-left">
            <div class="form-group">
                <input type="text" placeholder="Search" class="form-control">
            </div>
        </form>
    <ul class="nav navbar-nav navbar-right">
```

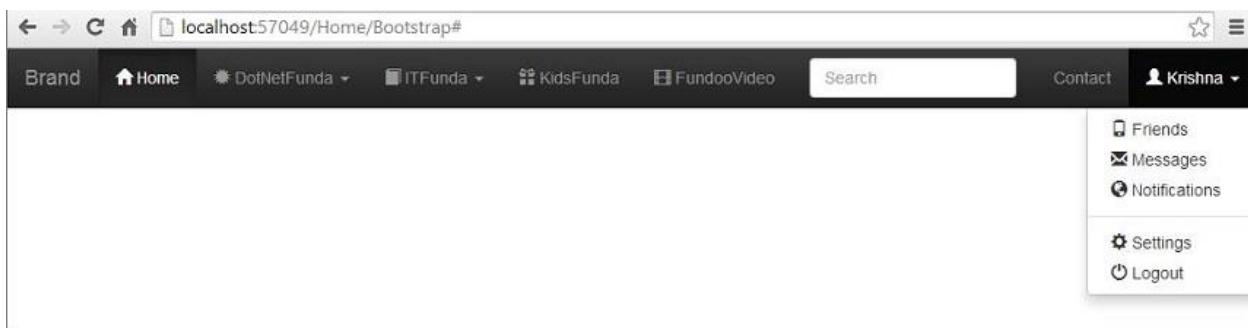
```

        <li><a href="#">Contact</a></li>
        <li class="dropdown">
            <a data-toggle="dropdown" class="dropdown-toggle" href="#"><span
class="glyphicon glyphicon-user"></span> Krishna <b class="caret"></b><span
class="glyphicon glyphicon-none"></span></a>
            <ul role="menu" class="dropdown-menu">
                <li><a href="#"><span class="glyphicon glyphicon-phone"></span>
Friends</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-
envelope"></span> Messages</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-globe"></span>
Notifications</a></li>
                <li class="divider"></li>
                <li><a href="#"><span class="glyphicon glyphicon-cog"></span>
Settings</a></li>
                <li><a href="#"><span class="glyphicon glyphicon-off"></span>
Logout</a></li>
            </ul>
        </li>
    </ul>
</div>
</nav>

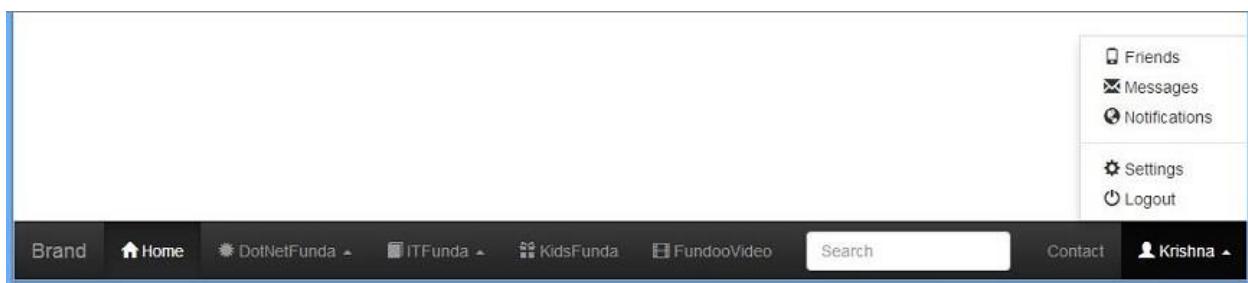
```

In the above code if you observe, we just added `navbar-fixed-top` to the `nav` element.

Let's run this code in your browser to get the following output



Observe the output that your Menu is fixed to top. Similarly, add `navbar-fixed-bottom` instead of `navbar-fixed-top` in the above code and run it. You will see the menu fixed to bottom like below,



Similarly we have `navbar-static-top` which gives static menu.

Conclusion

We have seen the creation of different types of Menus using Bootstrap in this Chapter. Hope you understand it.

Thanks for reading.

Chapter-12

Panels

Introduction

In some webpages, there is a need of placing content within panels to make them separate or look effective. Bootstrap comes with predefined `panel` classes which are used to satisfy our needs.

Using Bootstrap code

`Bootstrap.css` has a lot of CSS regarding panels which are predefined and easily usable. Let's see one by one,

Getting Started with Panels

First we are going to create a simple panel with a text in it. See the below code snippet,

```
<div class="panel panel-default">
    <div class="panel-body">This text is in Panel</div>
</div>
```

This will display a panel in your browser like below



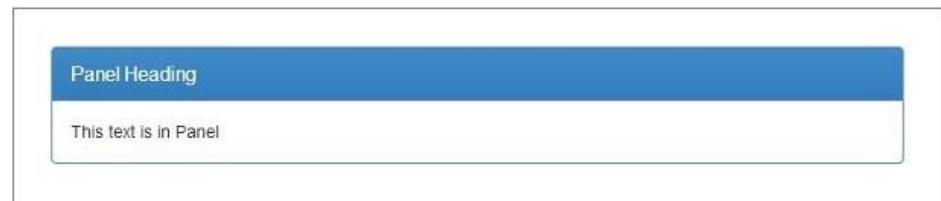
We can also use `panel-primary`, `panel-success`, `panel-warning`, `panel-info`, `panel-danger` classes for different colors of panels.

Also we can add Heading to the panel as following,

```
<div class="panel panel-primary">
    <div class="panel-heading"><p class="panel-title">Panel Heading</p></div>
    <div class="panel-body">This text is in Panel</div>
</div>
```

'`panel-heading`' is the class for Heading and `panel-body` is for description and everything that goes under panel.

If you run this code you will see the following output in your browser



We can also put the media elements such as Images in the panel. Take a look at below example for that

```
<div class="panel panel-success">
    <div class="panel-heading"><p class="panel-title">DotNetFunda</p></div>
    <div class="panel-body">
        <div class="media">
            <a href="#" class="pull-left">
                
            </a>
            <div class="media-body">
                <p>
                    DotNetFunda.Com is a popular online tutorials and guide for
latest Microsoft® technologies
                    aimed for beginners and intermediate level professionals. By
following our different sections regularly,
                    we hope you will soon become one of the 'Most Valuable'
professional and start shining in your career.
                </p>
            </div>
        </div>
    </div>
</div>
```

In the above code we are adding Image with some Description as we done in the previous chapter [Images in Bootstrap](#). Run the above code in your browser and see the result. It looks like below,



Panels with Footers

We can include footer in the panel by using `panel-footer` predefined class of Bootstrap.

Observe the below example of adding footer with some buttons to the Panel.

```
<div class="panel panel-primary">
    <div class="panel-heading"><p class="panel-title">DotNetFunda</p></div>
    <div class="panel-body">DotNetFunda is the website for those who are looking for
.Net Tips and Tricks.</div>
    <div class="panel-footer clearfix">
        <div class="pull-right">
            <a href="#" class="btn btn-warning">Back</a>
            <a href="http://dotnetfunda.com" class="btn btn-success">DotNetFunda</a>
            <a href="#" class="btn btn-info">Next</a>
        </div>
    </div>
</div>
```

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

Now, if you run this code you will see the following result in your browser

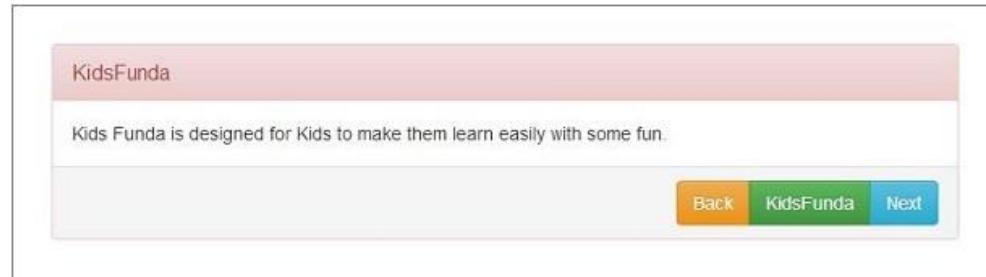


We can also group the buttons in the Footer of the Panel.

Let's try that with below code,

```
<div class="panel panel-danger">
    <div class="panel-heading"><p class="panel-title">KidsFunda</p></div>
    <div class="panel-body">Kids Funda is designed for Kids to make them learn easily
with some fun.</div>
    <div class="panel-footer clearfix">
        <div class="pull-right">
            <div class="btn-group">
                <a href="#" class="btn btn-warning">Back</a>
                <a href="http://kidsfunda.com" class="btn btn-success">KidsFunda</a>
                <a href="#" class="btn btn-info">Next</a>
            </div>
        </div>
    </div>
</div>
```

In the above code, we have used `btn-group` class to group the buttons as in the below output



And we can add any other elements in the Footer of the Panel. Example of adding Search box is below

```

<div class="panel panel-warning">
    <div class="panel-heading"><p class="panel-title">DotNetFunda</p></div>
    <div class="panel-body">DotNetFunda is the website for those who are looking for
    .Net Tips and Tricks.</div>
    <div class="panel-footer clearfix">
        <div class="pull-right">
            <div class="input-group">
                <input type="text" class="form-control" />
                <div class="input-group-btn">
                    <a href="#" class="btn btn-info"><span class="glyphicon glyphicon-search"></span> Search</a>
                </div>
            </div>
        </div>
    </div>
</div>

```

In the above code we are adding a Textbox grouped with Search button to the Footer. That looks like below in your browser



Tables inside the Panel

We have already seen creating [Tables in Bootstrap](#). Now lets put those tables in the panels.

```

<div class="panel panel-primary">
    <!-- Panel content goes here -->
    <div class="panel-heading">Panel Heading</div>
    <div class="panel-body">
        <h2>This is the Panel body.</h2>
    </div>
    <!-- Table in the Panel -->
    <div>
        <table class="table table-bordered">
            <thead>
                <tr>
                    <th>S/no.</th>
                    <th>Name</th>
                    <th>Details</th>
                    <th>Contact</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>1</td>
                    <td>DotNetFunda</td>
                    <td>Fundamentals of .Net</td>
                </tr>
            </tbody>
        </table>
    </div>
</div>

```

```

                <td>support@dotnetfunda.com</td>
            </tr>
            <tr>
                <td>2</td>
                <td>ItFunda</td>
                <td>Training of Microsoft Technologies</td>
                <td>support@itfunda.com</td>
            </tr>
            <tr>
                <td>3</td>
                <td>KidsFunda</td>
                <td>Learning Fundamentals for Kids</td>
                <td>support@kidsfunda.com</td>
            </tr>
        </tbody>
    </table>
</div>
</div>

```

In the above code we have placed a table inside the panel.

After running the above code, you will meet the following output

Panel Heading			
This is the Panel body.			
S/no.	Name	Details	Contact
1	DotNetFunda	Fundamentals of .Net	support@dotnetfunda.com
2	ItFunda	Training of Microsoft Technologies	support@itfunda.com
3	KidsFunda	Learning Fundamentals for Kids	support@kidsfunda.com

A Panel with Table in it looks very nice. Isn't it !!

Lists inside the Panels

We have already seen creating [Lists in Bootstrap](#). we can put Lists inside the Panel and make them more effective.

Go through the below example

```

<div class="panel panel-primary">
    
    <div class="panel-heading">Panel Heading</div>
    <div class="panel-body">
        <h4>This is the Panel body.</h4>
    </div>
    <!-- List group in the Panel --&gt;
    &lt;div class="list-group"&gt;
        &lt;a href="#" class="list-group-item"&gt;
            &lt;span class="glyphicon glyphicon-home"&gt;&lt;/span&gt; Home &lt;span
</pre>

```

```

class="badge">512</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-heart"></span> Favourites <span
class="badge">45</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-camera"></span> Photos <span
class="badge">55</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-music"></span> Music <span
class="badge">266</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-film"></span> Videos <span
class="badge">84</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-trash"></span> Trash <span
class="badge">1</span>
    </a>
    <a href="#" class="list-group-item">
        <span class="glyphicon glyphicon-cog"></span> Settings <span
class="badge"></span>
    </a>
</div>
</div>

```

Above code is the List of some Items that sit inside the Panel.

The output of the above code will be,

The screenshot shows a Bootstrap panel with a blue header bar containing the text "Panel Heading". The main body of the panel contains the text "This is the Panel body." followed by a list of items. Each item consists of an icon, a label, and a circular badge indicating a count. The items are:

- Home (icon: house) - badge: 512
- Favourites (icon: heart) - badge: 45
- Photos (icon: camera) - badge: 55
- Music (icon: musical note) - badge: 266
- Videos (icon: film) - badge: 84
- Trash (icon: trash bin) - badge: 1
- Settings (icon: gear) - badge: 0

Similarly, you can place so many elements like Dropdowns, Radio buttons, Checkboxes etc. in the Panel.

Conclusion

In this chapter we have seen Panels and placing elements in panels using Bootstrap predefined CSS. Hope you understand.

Thanks for reading.

Bread crumbs, Pagination and Pagers

Introduction

As we all know that **pagination** plays a vital role in the websites to divide the content into pages and make them easily accessible to the user.

Using Bootstrap code

We have predefined classes for pagination, pager and breadcrumb in `bootstrap.css`.

Bread Crumbs:

Have a look at the below code snippet of Bread Crumbs

```
<ul class="breadcrumb">
    <li><a href="#">Home</a></li>
    <li><a href="#">Articles</a></li>
    <li class="active">ASP.Net</li>
</ul>
```

We have used `breadcrumb` class to create them. Output of above code will be,



Pagination:

Pagination class in `bootstrap.css` helps us to create a pagination bar.

Basic Pagination:

Below is an example of doing simple pagination

```
<ul class="pagination">
    <li><a href="#">&lquo;</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">6</a></li>
    <li><a href="#">&raquo;</a></li>
</ul>
```

Run the code in your browser and you will see the pagination bar as below,



Active & Disabled Pagination:

If you want to disable the any `li` element, we have `disabled` class. Also we can make page active with `active` class. See the below code,

```
<ul class="pagination">
    <li class="disabled"><a href="#">&laquo;</a></li>
    <li class="active"><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">6</a></li>
    <li><a href="#">&raquo;</a></li>
</ul>
```

Now see the above code in your browser



We can also manage the size of the pagination bar with `pagination-sm` and `pagination-lg` classes of Bootstrap. They are,

Small

```
<ul class="pagination pagination-sm">
    <li class="disabled"><a href="#">&laquo;</a></li>
    <li class="active"><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">6</a></li>
    <li><a href="#">&raquo;</a></li>
</ul>
```

Output of the above code will give small pagination bar as follows



Large

```
<ul class="pagination pagination-lg">
    <li class="disabled"><a href="#">&laquo;</a></li>
    <li class="active"><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">6</a></li>
    <li><a href="#">&raquo;</a></li>
</ul>
```

Output of the above code will gives large pagination bar like below



Pagers:

Pagers in the sense next and previous buttons that used to navigate adjacent pages.

Bootstrap contains predefined **pager** classes. Lets see different types of pagers in Bootstrap as follows,

Pager-1

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

The above code creates two buttons like below



Pager-2

We can arrange the buttons to the respective edges.

```
<ul class="pager">
    <li class="previous"><a href="#"><span class="glyphicon glyphicon-chevron-left"></span> Previous</a></li>
    <li class="next"><a href="#">Next <span class="glyphicon glyphicon-chevron-right"></span></a></li>
</ul>
```

This code will give a proper arranged pager buttons in our browser



Pager-3

If you want to disable a pager button, simply use **disabled** class and do it like below.

```
<ul class="pager">
    <li class="previous disabled"><a href="#"><span class="glyphicon glyphicon-hand-left"></span> Previous</a></li>
    <li class="next"><a href="#">Next <span class="glyphicon glyphicon-hand-right"></span></a></li>
</ul>
```

Now observe the output with a disabled 'previous' button



Pager-4

Simple pagers without names comes as follows,

```
<ul class="pager">
    <li class="previous"><a href="#"><span class="glyphicon glyphicon-arrow-left"></span></a></li>
    <li class="next"><a href="#"><span class="glyphicon glyphicon-arrow-right"></span></a></li>
</ul>
```

Run the above code snip and you will see a simple pager in your screen



Conclusion

In this chapter, we have seen breadcrumbs, paginations and pagers in Bootstrap. Hope you understand them.

Thanks for reading.

Chapter-14

Labels

Introduction

Bootstrap consists of some `label` related classes which are used to style the labels in the webpage.

Using Bootstrap code

We have label classes in `bootstrap.css`. See the below example of creating a simple label.

```
<div>
    <span class="label label-default">I'm Label</span>
</div>
```

Now you will see the label in your browser as below,



Different sizes of Labels:

HTML had `h1`, `h2`, `h3`... tags for creating different sizes of labels.

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

Run the above code in your browser to see the following output



Types of Bootstrap Labels:

Bootstrap comes with different colors of label classes. Let's observe the below code,

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

If you observe the above code, we have `label-primary`, `label-warning`, `label-info` etc. predefined styles from Bootstrap.

The above code will give following output in your browser

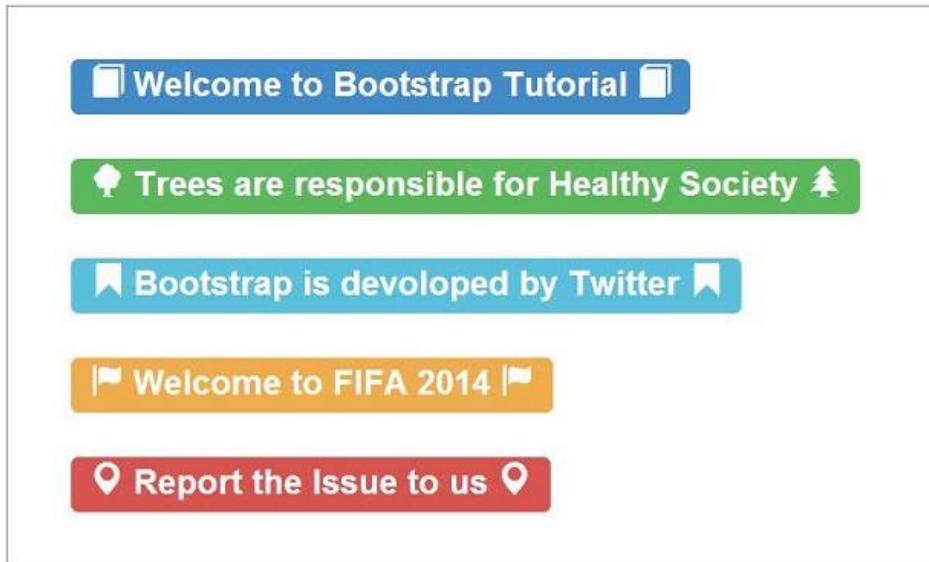


Creating Banners:

We can create Banners with Labels that might be used in some places like superior headings. See the below Example,

```
<div>
  <h1><span class="label label-primary"><span class="glyphicon glyphicon-book"></span>
Welcome to Bootstrap Tutorial <span class="glyphicon glyphicon-book"></span></span></h1>
<br />
<h1><span class="label label-success"><span class="glyphicon glyphicon-tree-deciduous"></span> Trees are responsible for Healthy Society <span class="glyphicon glyphicon-tree-conifer"></span></span></h1>
<br />
<h1><span class="label label-info"><span class="glyphicon glyphicon-bookmark"></span>
Bootstrap is developed by Twitter <span class="glyphicon glyphicon-bookmark"></span></span></h1>
<br />
<h1><span class="label label-warning"><span class="glyphicon glyphicon-flag"></span>
Welcome to FIFA 2014 <span class="glyphicon glyphicon-flag"></span></span></h1>
<br />
<h1><span class="label label-danger"><span class="glyphicon glyphicon-map-marker"></span> Report the Issue to us <span class="glyphicon glyphicon-map-marker"></span></span></h1>
</div>
```

Save and Run the above code to see some example banners in your browser



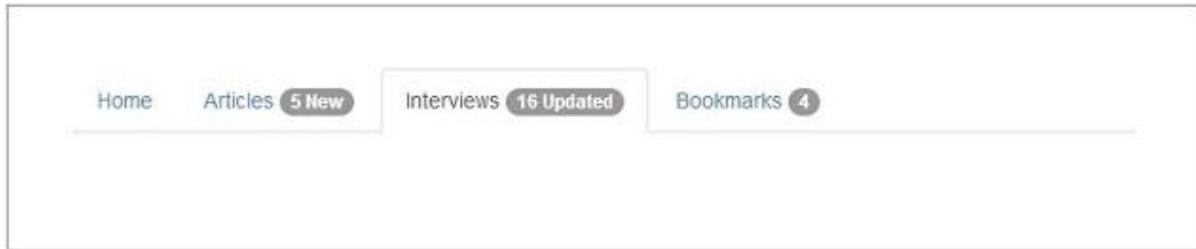
Creating Badges:

We have already seen Badges in [Chapter-5 \(Lists\)](#). Let's put them in the menu bar here.

```
<div>
  <ul class="nav nav-tabs">
    <li><a href="#">Home</a></li>
    <li><a href="#">Articles <span class="badge">5 New</span></a></li>
    <li class="active"><a href="#">Interviews <span class="badge">16
Updated</span></a></li>
    <li><a href="#">Bookmarks <span class="badge">4</span></a></li>
  </ul>
</div>
```

To get with Navs, Read [Chapter-9](#) and [10](#).

If you run the above code, you will find the following output in the browser



Conclusion

In this chapter, we have seen creating different types of Labels using Bootstrap. Hope you understand.

Thanks for reading.

Chapter-15

Progress Bars

Introduction

In some situations, it is necessary to show the user about the process of executions. This can be achieved by using Progress Bars. Bootstrap has some predefined progress bar styles which can be used to create nice progress bars easily.

Using Bootstrap code

We have `progress` and `progress-bar` classes in `bootstrap.css` to create a progress bar.

Normal Progress-Bar:

See the below example of creating simple progress bar,

```
<div class="progress">
  <div class="progress-bar progress-bar-info" style="width: 70%;">
    Please wait...
  </div>
</div>
```

This will shows a progress bar in your browser



Striped Progress-Bar:

Add `progress-striped` to the above normal progress bar code like below,

```
<div class="progress progress-striped">
  <div class="progress-bar progress-bar-info" style="width: 70%;">
    Please wait...
  </div>
</div>
```

Run this code to see a striped progress bar as following



Animated Striped Progress-Bar:

If you want an active progress bar in which stripes are moving linearly, Add `active` to the above striped progress bar.

```
<div class="progress progress-striped active">
  <div class="progress-bar progress-bar-info" style="width: 70%;">
    Please wait...
  </div>
</div>
```

This gives you a running progress bar



You will see the running stripes in your browser.

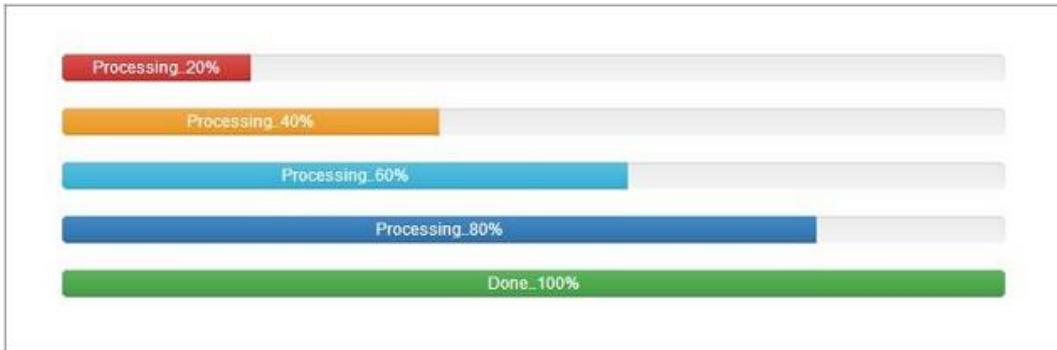
Different colors of Normal Progress-Bars:

In Bootstrap, we have some predefined colors of progress bars,

```
<div class="progress">
  <div class="progress-bar progress-bar-danger" style="width: 20%">
    Processing..20%
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-warning" style="width: 40%">
    Processing..40%
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-info" style="width: 60%">
    Processing..60%
  </div>
</div>
<div class="progress">
  <div class="progress-bar" style="width: 80%">
    Processing..80%
  </div>
</div>
<div class="progress">
  <div class="progress-bar progress-bar-success" style="width: 100%">
    Done..100%
  </div>
</div>
```

Observe the above code in which we used five progress colors of danger, warning, info, default and success.

Run this code and see the following output in your browser,

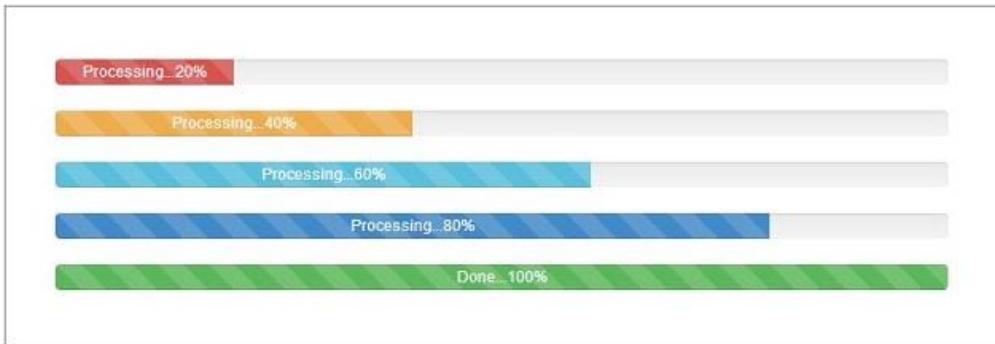


Different colors of Striped Progress-Bars:

As in the same above code, just add `progress-striped` to get the striped bars

```
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-danger" style="width: 20%">
        Processing...20%
    </div>
</div>
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-warning" style="width: 40%">
        Processing...40%
    </div>
</div>
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-info" style="width: 60%">
        Processing...60%
    </div>
</div>
<div class="progress progress-striped">
    <div class="progress-bar" style="width: 80%">
        Processing...80%
    </div>
</div>
<div class="progress progress-striped">
    <div class="progress-bar progress-bar-success" style="width: 100%">
        Done...100%
    </div>
</div>
```

If you run this code, you will see the striped progress bars in your browser,



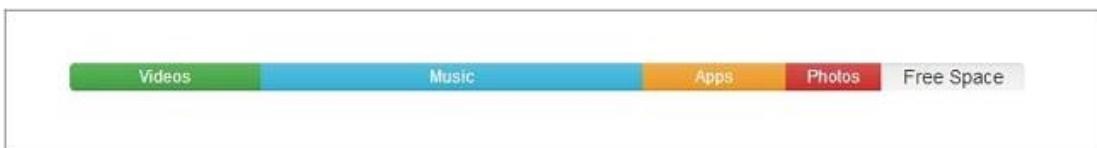
Creating Itunes Progress-Bar

We can also create a progress bar which we see in Apple Itunes.

Observe the below simple code snippet of doing that,

```
<div class="progress">
    <div class="progress-bar progress-bar-success" style="width: 20%">
        Videos
    </div>
    <div class="progress-bar progress-bar-info" style="width: 40%">
        Music
    </div>
    <div class="progress-bar progress-bar-warning" style="width: 15%">
        Apps
    </div>
    <div class="progress-bar progress-bar-danger" style="width: 10%">
        Photos
    </div>
    <div class="text-center">Free Space</div>
</div>
```

In the above code, we are placing different progress-bars in a single progress which gives the following result



Conclusion

In this chapter, we have seen different types of Progress-Bars using Twitter Bootstrap. Hope you understand.

Thanks for reading.

Chapter-16

Utility Components

Introduction

There are some Utility classes in `bootstrap.css`. These are used to create some usable components in our browser.

Using Bootstrap code

We have some classes in `bootstrap.css` that are used in different scenarios. Let's see them one by one.

Well:

`well` is a simple class which looks like `panel` in Bootstrap.

```
<div class="well">
    This is a <a href="#" class="label label-success">Well</a>
</div>
```

Run the above code in your browser to see a well like below



Alert:

`alert` is for showing notifications and alerts in the browser. Below example shows simple alert message

```
<div class="alert alert-warning">
    This is a Warning Notification
    <a href="#" class="close" data-dismiss="alert">&times;</a>
</div>
```

See the above code in your browser



You can hide it by clicking the close mark at the right.

We can also create alerts that are displayed in Google Chrome like below,

```
<div class="alert alert-info">
  Do you want to make <b>Google Chrome</b> as your Default Browser
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <a href="#" class="btn btn-info btn-xs">Yes</a>
  <a href="#" class="btn btn-info btn-xs">No</a>
</div>
```

If you run the above code, you will see the following output in your browser



Let's try creating another one as shown below,

```
<div class="alert alert-success">
  Do you want <b>Google Chrome</b> to..
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <div class="navbar-right">
    <a href="#" class="btn btn-success btn-xs">Remember Password</a>
    <a href="#" class="btn btn-success btn-xs">Never for this site</a>
  </div>
</div>
```

In the above code, we are setting buttons to the right that would be resulted as below



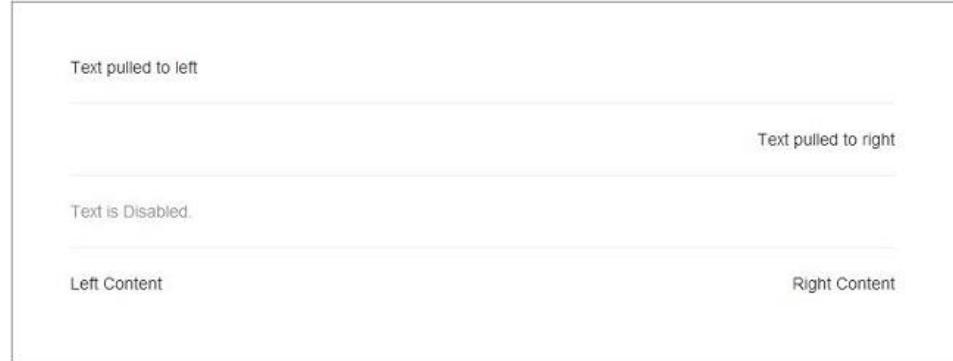
Pull-left, Pull-right, Text-muted and Clearfix:

These are used to adjust your content to keep it where ever you want.

```
<div class="pull-left">Text pulled to left</div>
<br /><hr />
<div class="pull-right">Text pulled to right</div>
<br /><hr />
<div class="text-muted">Text is Disabled.</div>
<br />
```

```
<div class="clearfix">
  <div class="pull-left">Left Content</div>
  <div class="pull-right">Right Content</div>
</div>
```

In the above code, we have used all of them. Run it and see the following output in your browser



Jumbotron:

Jumbotron is like a container in which we can place all our elements. Observe the below code of using **jumbotron**.

```
<div class="jumbotron">
  <h1>Hello, Welcome to Bootstrap</h1>
  <h3>Bootstrap is developed by Twitter Inc. which is used to design the websites with enormous look and feel.</h3>
  <a href="http://getbootstrap.com/" class="btn btn-success btn-lg">Download Bootstrap</a>
  <a href="#" class="btn btn-warning">Learn more on Bootstrap</a>
</div>
```

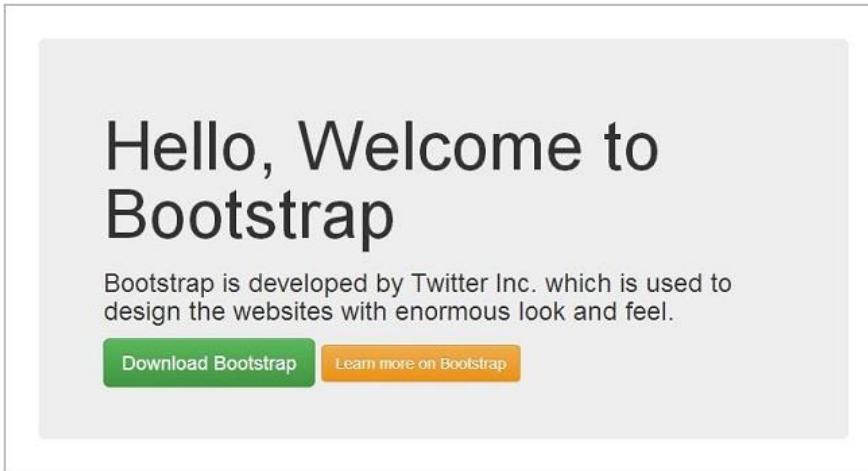
This will give the following result in your browser



We can make it some different such as rounded corners, formatting content etc. For that we have to use `container` like below

```
<div class="container">
  <div class="row">
    <div class="col-xs-8">
      <div class="jumbotron">
        <h1>Hello, Welcome to Bootstrap</h1>
        <h3>Bootstrap is developed by Twitter Inc. which is used to design the websites with enormous look and feel.</h3>
        <a href="http://getbootstrap.com/" class="btn btn-success btn-lg">Download Bootstrap</a>
          <a href="#" class="btn btn-warning">Learn more on Bootstrap</a>
        </div>
      </div>
    </div>
  </div>
```

Now `jumbotron` inherits some of the `container` properties such as rounded corners. Run and see the change as following



Conclusion

In this chapter, we have seen different types of utility components in Bootstrap. Hope you understand it.

Thanks for reading.

Chapter-17

Buttons-I

Introduction

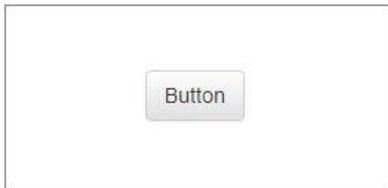
Buttons are the key in any web application to perform an action or to switch pages. There are some predefined buttons in Bootstrap that can be used in our applications.

Using Bootstrap code

We have `btn` class in `bootstrap.css` which is responsible to create Buttons in our applications. Let's create a simple button like below,

```
<div>
    <button type="button" class="btn btn-default">Button</button>
</div>
```

`btn-default` creates a normal simple button in your web page as below,



Different sizes of Buttons:

We have `large`, `small`, and `extra small` button classes in `bootstrap.css`. These are described in the below example,

```
<div>
    <button type="button" class="btn btn-default btn-lg">Large button</button>
    <button type="button" class="btn btn-default btn-sm">Small button</button>
    <button type="button" class="btn btn-default btn-xs">Extra Small button</button>
    <button type="button" class="btn btn-default">Default Button</button>
</div>
```

`btn-lg` (large button), `btn-sm` (small button) and `btn-xs` (extra small) are responsible for large, small and extra small buttons



Types of Buttons:

We have some predefined types of buttons such as `primary`, `success`, `warning`, `danger` and `info` which are of different types in terms of colors. They are shown in the below code

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

The above code in your browser will give different colors of buttons like below

A horizontal row of seven buttons. From left to right: a light gray button labeled "Default", a blue button labeled "Primary", an orange button labeled "Warning", a green button labeled "Success", a teal button labeled "Info", a red button labeled "Danger", and a light blue button labeled "Link Button".

Disabled Button:

In some scenarios, we have to disable the button to the user which hasn't been accessed by him.

```
<div>
  <a href="#" class="btn btn-info disabled">Disabled Button</a>
  <a href="#" class="btn btn-info">Normal Button</a>
</div>
```

`disabled` class above makes your button as disabled like below

A horizontal row of two buttons. The first button is labeled "Disabled Button" and has a darker, more muted color than the second button. The second button is labeled "Normal Button" and has a standard, vibrant light blue color.

Loading Button:

In some applications like making transactions, there is a need to disable button for some time after clicking it and enable automatically. For those type of buttons we need to write a little script in our page like below,

```
<script type="text/javascript">
$(function () {
    $(".btn").click(function () {
        $(this).button('loading').delay(2000).queue(function () {
            $(this).button('reset');
            $(this).dequeue();
        });
    });
});
</script>
```

And we have to give `data-loading-text` attribute in our `button` tag as did in the below code

```
<div>
    <button type="button" class="btn btn-success" data-loading-text="Please
Wait...">Submit</button>
    <button type="button" class="btn btn-warning" data-loading-text="Processing..">Show
Details</button>
</div>
```

Now, let's run this code in your browser to get the following output



In the above figure, it is clearly mentioned that what happens on the button clicks.

Toggle Button:

What to do if need to make the button active on clicking it and toggling back with another click?

Simple, Just use `data-toggle` attribute as we have in the below code

```
<div>
  <button type="button" class="btn btn-info" data-toggle="button">Click Me</button>
  <button type="button" class="btn btn-danger" data-toggle="button">Click Me
too</button>
</div>
```

This attribute will enables the toggling option for our buttons as shown below



Click the buttons and see their 'ON' and 'OFF' status.

Block Buttons:

Observe the below example code in which we added `btn-block` class which is responsible to create the block buttons in our web page

```
<div>
  <button type="button" class="btn btn-success btn-block">Accept</button>
  <button type="button" class="btn btn-danger btn-block">Decline</button>
</div>
```

If you run the above code snip, you might see the following output in your browser



Conclusion

In this chapter, we have seen different buttons of Twitter Bootstrap. Hope you understand.

We will see Button Groups in the next chapter.

Thank you for reading.

Chapter-18

Buttons-II

Introduction

Take a look at [Chapter-17](#) before this to understand easily as this is the continuation of that chapter.

So far we have seen buttons in Bootstrap and now let's do something more with them.

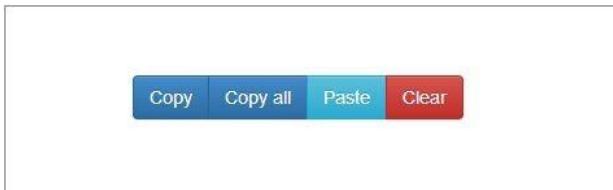
Using Bootstrap code

We have `btn-group` class in `bootstrap.css` that is used to group the buttons.

Observe the below example code of grouping buttons

```
<div class="btn-group">
  <button type="button" class="btn btn-primary">Copy</button>
  <button type="button" class="btn btn-primary">Copy all</button>
  <button type="button" class="btn btn-info">Paste</button>
  <button type="button" class="btn btn-danger">Clear</button>
</div>
```

The above code will gives you the following output in your browser



There is `btn-group-vertical` to arrange and group them in vertical manner.

```
<div class="btn-group-vertical">
  <button type="button" class="btn btn-primary">Copy</button>
  <button type="button" class="btn btn-primary">Copy all</button>
  <button type="button" class="btn btn-info">Paste</button>
  <button type="button" class="btn btn-danger">Clear</button>
</div>
```

Change in the code like above gives the following change in your browser



Different Sizes of Button Groups:

There are different sizes of `btn-group` classes. They are `btn-group-lg` (for large button group), `btn-group-sm` (for small button group) and `btn-group-xs` (for extra small buttons).

```
<div>
  <div class="btn-group btn-group-lg">
    <button type="button" class="btn btn-primary">Copy</button>
    <button type="button" class="btn btn-primary">Copy all</button>
    <button type="button" class="btn btn-info">Paste</button>
    <button type="button" class="btn btn-danger">Clear</button>
  </div>
  <hr />
  <div class="btn-group btn-group-sm">
    <button type="button" class="btn btn-primary">Copy</button>
    <button type="button" class="btn btn-primary">Copy all</button>
    <button type="button" class="btn btn-info">Paste</button>
    <button type="button" class="btn btn-danger">Clear</button>
  </div>
  <hr />
  <div class="btn-group btn-group-xs">
    <button type="button" class="btn btn-primary">Copy</button>
    <button type="button" class="btn btn-primary">Copy all</button>
    <button type="button" class="btn btn-info">Paste</button>
    <button type="button" class="btn btn-danger">Clear</button>
  </div>
</div>
```

If you observe the above code, there are large, small and extra small button group classes which looks like below



Now, for vertical Button Groups,

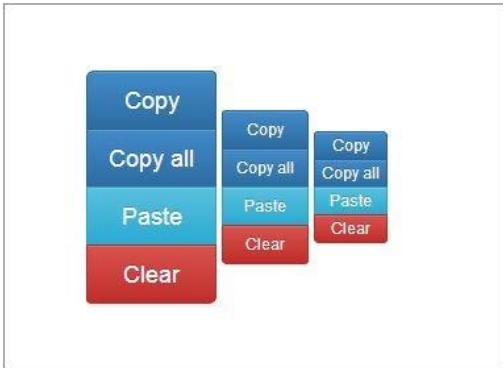
```
<div>
  <div class="btn-group-vertical btn-group-lg">
    <button type="button" class="btn btn-primary">Copy</button>
    <button type="button" class="btn btn-primary">Copy all</button>
    <button type="button" class="btn btn-info">Paste</button>
    <button type="button" class="btn btn-danger">Clear</button>
  </div>
  <div class="btn-group-vertical btn-group-sm">
    <button type="button" class="btn btn-primary">Copy</button>
    <button type="button" class="btn btn-primary">Copy all</button>
```

```

        <button type="button" class="btn btn-info">Paste</button>
        <button type="button" class="btn btn-danger">Clear</button>
    </div>
    <div class="btn-group-vertical btn-group-xs">
        <button type="button" class="btn btn-primary">Copy</button>
        <button type="button" class="btn btn-primary">Copy all</button>
        <button type="button" class="btn btn-info">Paste</button>
        <button type="button" class="btn btn-danger">Clear</button>
    </div>
</div>

```

We have added **vertical** in the above code which result in following output



Checkbox and RadioButton Functionalities:

There are predefined classes for **Checkbox** and **RadioButton** in `bootstrap.css`.

For CheckBox Functionality,

```

<div class="btn-group" data-toggle="buttons">
    <span class="btn btn-danger">
        <input type="checkbox" name="options"> Audi
    </span>
    <span class="btn btn-danger">
        <input type="checkbox" name="options"> Mercedes
    </span>
    <span class="btn btn-danger">
        <input type="checkbox" name="options"> BMW
    </span>
    <span class="btn btn-danger">
        <input type="checkbox" name="options"> Ferrari
    </span>
</div>

```

In the above code, we are using **checkbox** class to perform Checkbox action.

Run the above code in your browser to see the output something like below



If you observe the above figure, Both **Audi** and **Mercedes** are Checked and remaining two are Unchecked.'

For Radio Button Functionality, Just replace **checkbox** with **radio** like in the below code.

```
<div class="btn-group" data-toggle="buttons">
  <span class="btn btn-danger">
    <input type="radio" name="options"> Audi
  </span>
  <span class="btn btn-danger">
    <input type="radio" name="options"> Mercedes
  </span>
  <span class="btn btn-danger">
    <input type="radio" name="options"> BMW
  </span>
  <span class="btn btn-danger">
    <input type="radio" name="options"> Ferrari
  </span>
</div>
```

This code will gives you the RadioButton features in your browser as following



In the above figure, only **Mercedes** is checked and you are not allowed to check multiple which means the radio.

Button Toolbars:

We have **btn-toolbar** class in **bootstrap.css** which is used to create a toolbar. Lets see creating a toolbar with some elements through below code,

```

<div class="btn-toolbar">
    <div class="btn-group">
        <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-home"></span></button>
        <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-book"></span></button>
        <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-gift"></span></button>
        <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-calendar"></span></button>
    </div>
    <div class="btn-group">
        <button type="button" class="btn btn-danger"><span class="glyphicon glyphicon-camera"></span></button>
        <button type="button" class="btn btn-danger"><span class="glyphicon glyphicon-music"></span></button>
        <button type="button" class="btn btn-danger"><span class="glyphicon glyphicon-film"></span></button>
    </div>
    <div class="btn-group">
        <button type="button" class="btn btn-warning"><span class="glyphicon glyphicon-cog"></span></button>
    </div>
    <div class="btn-group">
        <button type="button" class="btn btn-primary"><span class="glyphicon glyphicon-collapse-down"></span></button>
        <button type="button" class="btn btn-primary"><span class="glyphicon glyphicon-collapse-up"></span></button>
    </div>
</div>

```

In the above code, we are using some Glyphicons of Bootstrap as tool elements. If you run this code in your browser, you will see the output something like below



We can also place the Toolbar some elements like in `panel`, `well` etc.

Just observe the below code in which we are going to place our Toolbar in a Panel.

```

<div class="panel panel-primary">
    <div class="panel-heading"><p class="panel-title">ToolBox</p></div>
    <div class="panel-body">
        <div class="btn-toolbar">
            <div class="btn-group">
                <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-home"></span></button>

```

```

                <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-book"></span></button>
                <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-gift"></span></button>
                <button type="button" class="btn btn-success"><span class="glyphicon glyphicon-calendar"></span></button>
            </div>
            <div class="btn-group">
                <button type="button" class="btn btn-danger"><span class="glyphicon glyphicon-camera"></span></button>
                <button type="button" class="btn btn-danger"><span class="glyphicon glyphicon-music"></span></button>
                <button type="button" class="btn btn-danger"><span class="glyphicon glyphicon-film"></span></button>
            </div>
            <div class="btn-group">
                <button type="button" class="btn btn-warning"><span class="glyphicon glyphicon-cog"></span></button>
            </div>
            <div style="float:right;">
                <div class="btn-group">
                    <button type="button" class="btn btn-info"><span class="glyphicon glyphicon-collapse-down"></span></button>
                    <button type="button" class="btn btn-info"><span class="glyphicon glyphicon-collapse-up"></span></button>
                </div>
            </div>
        </div>
    </div>

```

This will places our toolbar in a Panel like in the below picture



Conclusion

In this chapter, we have seen Button Groups and Button Toolbars of Bootstrap. Hope you understand.

We will see some more Bootstrap button actions with JavaScript.

Thanks for reading.

Chapter-19

Buttons-III

Introduction

Take a look at [Chapter-18](#) before this to understand easily as this is the continuation of that.

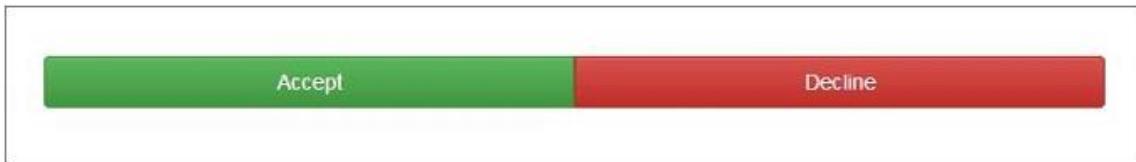
So far we have seen Button Groups and Button Toolbars with Bootstrap and here let's see some more button actions.

Using Bootstrap code

Before going to the JavaScript techniques, we have something called `btn-group-justified` in `bootstrap.css`. Let's see what it does from the below code,

```
<div class="btn-group btn-group-justified">
    <a href="#" class="btn btn-success">Accept</a>
    <a href="#" class="btn btn-danger">Decline</a>
</div>
```

This type will group the buttons and also makes it a block as shown in the below picture



Button Functions with JavaScript:

Toggle Buttons:

We have already seen toggling buttons in [Chapter-17](#). Now here let's do that function with JavaScript.

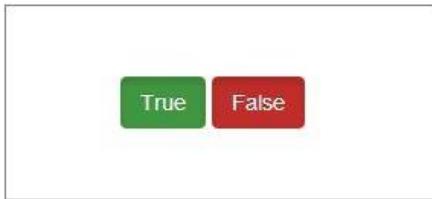
For this we have to write a little script in our page like below,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#toggleBtn .btn").click(function () {
        $(this).button('toggle');
    });
});
</script>
```

And adding buttons like below

```
<div id="toggleBtn">
    <button type="button" class="btn btn-success">True</button>
    <button type="button" class="btn btn-danger">False</button>
</div>
```

Now check this function in your browser and it looks like below output



We can see clearly in the above figure that the buttons are 'ON' (Checked).

Loading Buttons:

Earlier we have seen Loading Buttons in [Chapter-17](#). Here, we are going to perform the same functionality with JavaScript. For loading effect to the button, we have to use below script in our page

```
<script type="text/javascript">
$(document).ready(function () {
    $("#ldBtn .btn").click(function () {
        $(this).button('loading');
    });
});
</script>
```

And give that #ldbtn ID to our buttons as below

```
<div id="ldbtn">
    <button type="button" class="btn btn-warning">Enter</button>
</div>
```

Now run this combination in your browser to see the following result



See, after clicking **Enter** button, it will change its state to **Loading....**

String Buttons:

Now if you want to change the name of the button after the completion of a process, we can achieve it by using below JavaScript code in your page.

```
<script type="text/javascript">
$(document).ready(function () {
    $("#ldBtn .btn").click(function () {
        $(this).button('loading').delay(2000).queue(function () {
            $(this).button('complete');
            $(this).dequeue();
        });
    });
});
</script>
```

And also give that `#ldbtn` ID to our buttons as below,

```
<div id="ldbtn">
    <button type="button" class="btn btn-success" data-complete-text="Loading
Completed">Load</button>
</div>
```

Now see the change by running this combination in your browser. You might get result something like below,



After clicking the `Load` button it will start `Loading...` followed by `Loading Completed` (comes after the delay time).

Changing Tab Contents:

We can also switch the Tab contents with the Button Clicks. For such functionality, we are taking Tab using `nav-tabs` class of Bootstrap and using JavaScript. Include the below script in your page

```
<script type="text/javascript">
$(document).ready(function () {
    $(".nav-tabs a").click(function () {
        $(this).button('loading').delay(500).queue(function () {
            $(this).button('reset');
            $(this).dequeue();
        });
    });
});
```

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

.nav-tabs is the Bootstraps predefined class to create a tab and we are setting functions to [tags of Nav-Tabs](#) in the above script.

And take an example code of creating **Nav-Tabs** as we have below

```
<div>  
    <ul class="nav nav-tabs">  
        <li class="active"><a data-toggle="tab" href="#tabHome" data-loading-  
text="Wait..">Home</a></li>  
        <li><a data-toggle="tab" href="#tabProfile" data-loading-  
text="Wait...">Profile</a></li>  
        <li><a data-toggle="tab" href="#tabSettings" data-loading-  
text="Wait.....">Settings</a></li>  
    </ul>  
    <div class="tab-content">  
        <div id="tabHome" class="tab-pane active fade in">  
            <p>Home Content goes here</p>  
        </div>  
        <div id="tabProfile" class="tab-pane fade">  
            <p>Profile related Content goes here</p>  
        </div>  
        <div id="tabSettings" class="tab-pane fade">  
            <p>Settings related Content goes here</p>  
        </div>  
    </div>  
</div>
```

In the above example, we have three sections and the need is to switch the content of that sections on clicking them. So, Run this combination of code and see the result.



In the above figure, you can see the content switching on the option click.

Conclusion

In this chapter, we have seen different button functionalities using JavaScript. Hope you understand.

Thanks for reading.

Chapter-20

Dropdowns

Introduction

We have already seen dropdowns in previous chapter [Chapter-11](#), but here we are going to see them in detail.

Using Bootstrap code

We have some predefined classes in Bootstrap for dropdown menus. Let's go through the below example code with different types of dropdowns,

```
<div class="btn-group">
  <button data-toggle="dropdown" class="btn btn-primary dropdown-toggle">Menu <span
class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#">Option1</a></li>
    <li><a href="#">Option2</a></li>
    <li class="divider"></li>
    <li><a href="#">Settings</a></li>
  </ul>
</div>
<div class="btn-group">
  <button data-toggle="dropdown" class="btn btn-success dropdown-toggle">Menu <span
class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#">Option1</a></li>
    <li><a href="#">Option2</a></li>
    <li class="divider"></li>
    <li><a href="#">Settings</a></li>
  </ul>
</div>
<div class="btn-group">
  <button data-toggle="dropdown" class="btn btn-warning dropdown-toggle">Menu <span
class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#">Option1</a></li>
    <li><a href="#">Option2</a></li>
    <li class="divider"></li>
    <li><a href="#">Settings</a></li>
  </ul>
</div>
<div class="btn-group">
  <button data-toggle="dropdown" class="btn btn-danger dropdown-toggle">Menu <span
class="caret"></span></button>
  <ul class="dropdown-menu">
    <li><a href="#">Option1</a></li>
    <li><a href="#">Option2</a></li>
    <li class="divider"></li>
    <li><a href="#">Settings</a></li>
  </ul>
</div>
<div class="btn-group">
  <button data-toggle="dropdown" class="btn btn-info dropdown-toggle">Menu <span
class="caret"></span></button>
```

```

<ul class="dropdown-menu">
    <li><a href="#">Option1</a></li>
    <li><a href="#">Option2</a></li>
    <li class="divider"></li>
    <li><a href="#">Settings</a></li>
</ul>
</div>

```

In the above code, we have five types of Dropdown buttons which will look like below in your browser



We can also have default dropdown by using `btn-default` from `bootstrap.css`.

Dropdowns separated with Buttons:

If we want to give separate actions for dropdown and button, we have to code like below,

```

<div class="btn-group">
    <button class="btn btn-success">Menu</button>
    <button data-toggle="dropdown" class="btn btn-success dropdown-toggle"><span
class="caret"></span>&ampnbsp</button>
    <ul class="dropdown-menu">
        <li><a href="#">Option1</a></li>
        <li><a href="#">Option2</a></li>
        <li class="divider"></li>
        <li><a href="#">Settings</a></li>
    </ul>
</div>
<div class="btn-group btn-group-sm">
    <button class="btn btn-info">Menu</button>
    <button data-toggle="dropdown" class="btn btn-info dropdown-toggle"><span
class="caret"></span>&ampnbsp</button>
    <ul class="dropdown-menu">
        <li><a href="#">Option1</a></li>
        <li><a href="#">Option2</a></li>
        <li class="divider"></li>
        <li><a href="#">Settings</a></li>
    </ul>
</div>
<div class="btn-group btn-group-lg">
    <button class="btn btn-danger">Menu</button>
    <button data-toggle="dropdown" class="btn btn-danger dropdown-toggle"><span
class="caret"></span>&ampnbsp</button>
    <ul class="dropdown-menu">

```

```

<li><a href="#">Option1</a></li>
<li><a href="#">Option2</a></li>
<li class="divider"></li>
<li><a href="#">Settings</a></li>
</ul>
</div>

```

In the above code, we have used btn-group classes to group dropdown with button and also giving separate actions.

Run the above code in your browser and you will see the output something like below



These are the sizes of dropdowns.

Dropdowns with JavaScript:

We can also give action to the drop menu using javascript. Observe the below script that we are using to perform drop action

```

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

```

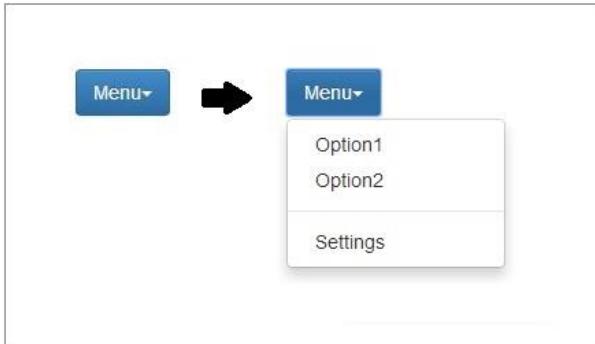
And your dropdown page is,

```

<div class="dropdown">
    <a href="#" class="btn btn-primary menuDrop">Menu<b class="caret"></b></a>
    <ul class="dropdown-menu">
        <li><a href="#">Option1</a></li>
        <li><a href="#">Option2</a></li>
        <li class="divider"></li>
        <li><a href="#">Settings</a></li>
    </ul>
</div>

```

Now run this combination in your browser to get following output as result,



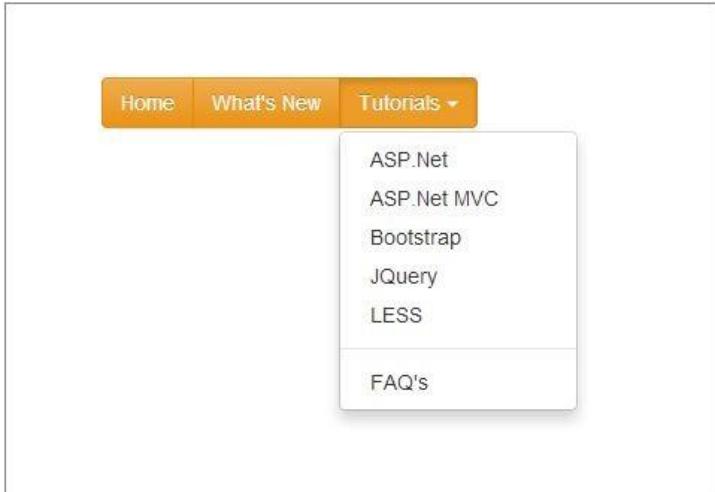
After clicking the **menu**, it will shows you a dropdown.

Dropdowns with more menu elements:

We can also group dropdowns with some other menu elements or buttons. Take a look at the below code which gets you the above scenario

```
<div class="btn-group">
    <button class="btn btn-warning">Home</button>
    <button class="btn btn-warning">What's New</button>
    <div class="btn-group">
        <button data-toggle="dropdown" class="btn btn-warning dropdown-toggle">Tutorials
    <span class="caret"></span></button>
    <ul class="dropdown-menu">
        <li><a href="#">ASP .Net</a></li>
        <li><a href="#">ASP .Net MVC</a></li>
        <li><a href="#">Bootstrap</a></li>
        <li><a href="#">JQuery</a></li>
        <li><a href="#">LESS</a></li>
        <li class="divider"></li>
        <li><a href="#">FAQ's</a></li>
    </ul>
    </div>
</div>
```

If you observe the above code, we are using **btn-group** class to group elements with dropdown. Run the above code in your browser to see the following result



Conclusion

In this chapter, we have seen different types of Dropdowns in Bootstrap. Hope you understand.

Thanks for reading.

Chapter-21

Tooltips

Introduction

Tool-tip shows the small description of the action in the web page. Bootstrap comes with some predefined Tool-tips that might be useful in our applications.

Using Bootstrap code

We have to write little JavaScript to generate tooltips in Bootstrap.

Observe the below code that is used to display normal tooltip in your browser.

JavaScript:

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

View Page:

```
<div class="bar">
    <ul class="tooltip-bar list-inline">
        <li><a href="#" class="btn btn-primary" data-toggle="tooltip" data-original-
title="Home">Home</a></li>
        <li><a href="#" class="btn btn-success" data-toggle="tooltip" data-original-
title="Tutorials">Tutorials</a></li>
        <li><a href="#" class="btn btn-warning" data-toggle="tooltip" data-original-
title="Videos">Videos</a></li>
    </ul>
</div>
```

Run this combination in your browser to see the the following output with tooltips



Toolips with Delay:

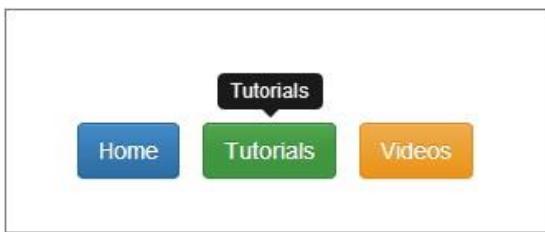
We can also set some delay to the tool-tip. Please go with below script to achieve that,

```
<script type="text/javascript">
$(document).ready(function () {
    $(".tooltip-bar a").tooltip({
        delay: { show : 1000 }
    });
});
</script>
```

And the View Page as follows,

```
<div class="bar">
    <ul class="tooltip-bar list-inline">
        <li><a href="#" class="btn btn-primary" data-toggle="tooltip" data-original-
title="Home">Home</a></li>
        <li><a href="#" class="btn btn-success" data-toggle="tooltip" data-original-
title="Tutorials">Tutorials</a></li>
        <li><a href="#" class="btn btn-warning" data-toggle="tooltip" data-original-
title="Videos">Videos</a></li>
    </ul>
</div>
```

Now run and see the variation in your output in which the tooltip appears after 1 second,



Toolips with Title from JavaScript:

We can also set tooltips from JavaScript that appears if there is no title given in the view code like below

```
<div class="bar">
    <ul class="tooltip-bar list-inline">
        <li><a href="#" class="btn btn-primary" data-toggle="tooltip">Home</a></li>
        <li><a href="#" class="btn btn-success" data-toggle="tooltip" data-original-
title="Tutorials">Tutorials</a></li>
        <li><a href="#" class="btn btn-warning" data-toggle="tooltip" data-original-
title="Videos">Videos</a></li>
    </ul>
</div>
```

And the JavaScript is,

```
<script type="text/javascript">
$(document).ready(function () {
    $(".tooltip-bar a").tooltip({
        title: "Hi, Welcome here"
    });
});
</script>
```

If you observe the above code, we didn't give any title to the **Home**. So it will show the title of the JavaScript like below



Tooltip Positions:

There is a **placement** property in Bootstrap's JavaScript that is used to set the position of the Tooltip.

Take a look at the below Script which explains the tooltip positions,

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

And the code in View Page as follows,

```
<div class="bar">
    <ul class="list-inline">
        <li><a href="#" class="btn btn-primary bar-top" data-toggle="tooltip" data-original-title="Home" data-placement="top">Home</a></li>
        <li><a href="#" class="btn btn-success bar-bottom" data-toggle="tooltip" data-original-title="Tutorials" data-placement="bottom">Tutorials</a></li>
        <li><a href="#" class="btn btn-warning bar-left" data-toggle="tooltip" data-original-title="Videos" data-placement="left">Videos</a></li>
        <li><a href="#" class="btn btn-danger bar-right" data-toggle="tooltip" data-original-title="Settings" data-placement="right">Settings</a></li>
    </ul>
</div>
```

If you run the above combination of code, you will see the output something like below in your browser,



Tooltip actions from JavaScript:

We can also `show`, `hide`, `toggle` or `destroy` the Tooltips manually with some JavaScript properties of Bootstrap.

Let's see the below code which explains such scenarios,

JavaScript:

```
<script type="text/javascript">
$(document).ready(function () {
    $(".show-bar").click(function () {
        $(".tooltip-bar a").tooltip('show');
    });
    $(".hide-bar").click(function () {
        $(".tooltip-bar a").tooltip('hide');
    });
    $(".toggle-bar").click(function () {
        $(".tooltip-bar a").tooltip('toggle');
    });
    $(".destroy-bar").click(function () {
        $(".tooltip-bar a").tooltip('destroy');
    });
});
</script>
```

View Page:

```
<div class="bar">
    <p class="tooltip-bar">
        <a href="#" class="btn btn-default" data-toggle="tooltip" title="Welcome to Home">Home</a>
    </p>
    <div>
        <p>Below are Tooltip actions from JavaScript</p>
        <input type="button" class="btn btn-success show-bar" value="Show" />
        <input type="button" class="btn btn-primary hide-bar" value="Hide" />
        <input type="button" class="btn btn-warning toggle-bar" value="Toggle" />
        <input type="button" class="btn btn-danger destroy-bar" value="Destroy" />
    </div>
</div>
```

Run the above code in your browser and you might see the output something like below



Conclusion

In this chapter, we have seen different properties of Tooltips in Bootstrap.

Thanks for reading.

Chapter-22

Pop Overs

Introduction

So far we have seen Tooltips in Chapter-21. There are some popovers which can be triggered through JavaScript in Bootstrap.

Using Bootstrap code

We have to write little JavaScript code to create Popovers for HTML elements. Let's see one by one starting with normal popover below,

JavaScript:

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

HTML:

```
<div>
    <ul class="popover-pop list-inline">
        <li><a href="#" class="btn btn-primary" data-toggle="popover" title="DotNetFunda"
data-content="Fundamentals of .Net">DotNetFunda</a></li>
    </ul>
</div>
```

Run the above combination of code to get a popover for link like below



We can see the **Title** and **Description** in the popover meanwhile, we can set them in HTML code.

Popovers with Delay:

We can also set some delay to the popovers. For that, we have to use **delay** property of Bootstrap. Take a look at the below example of JavaScript and Html.

```

<script type="text/javascript">
$(document).ready(function () {
    $(".popover-pop a").popover({
        delay: { show : 1000 }
    });
});
</script>

```

And HTML as follows,

```

<div>
    <ul class="popover-pop list-inline">
        <li><a href="#" class="btn btn-success" data-toggle="popover" title="DotNetFunda"
data-content="Fundamentals of .Net">DotNetFunda</a></li>
    </ul>
</div>

```

If you run this code in your browser, you will see the output something like below



Popover with Title from JavaScript:

We can also set title to the popovers from JavaScript as we did for tooltips. See the below JavaScript that is similar to that of tooltips

```

<script type="text/javascript">
$(document).ready(function () {
    $(".popover-pop a").popover({
        title : "Title from Script"
    });
});
</script>

```

And HTML as,

```

<div>
    <ul class="popover-pop list-inline">
        <li><a href="#" class="btn btn-primary" data-toggle="popover" title="DotNetFunda"
data-content="Fundamentals of .Net">DotNetFunda</a></li>
        <li><a href="#" class="btn btn-primary" data-toggle="popover" data-
content="Fundamentals of .Net">DotNetFunda</a></li>
    </ul>
</div>

```

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

This combination will give you the title from JavaScript only if there is no title given in HTML.



Popovers with Triggering:

We can also set the popover to come on **Hover** or **Click**. By default it is on **Click** mode. If you want to make it function on **Hover**, let's go with below approach

```
<script type="text/javascript">
$(document).ready(function () {
    $(".popover-pop_a").popover({
        trigger : 'hover'
    });
});
</script>
```

And HTML as,

```
<div>
    <ul class="popover-pop list-inline">
        <li><a href="#" class="btn btn-primary" data-toggle="popover" title="DotNetFunda" data-content="Fundamentals of .Net">DotNetFunda</a></li>
    </ul>
</div>
```

Now run and see this combination in your browser and it will be something like below



Popover Positions:

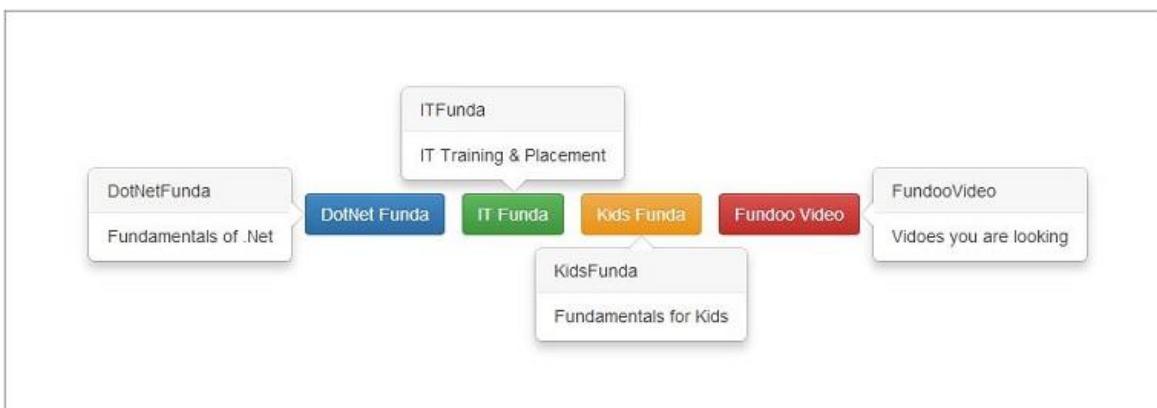
Like Tooltips, we have **placement** property for popovers in Bootstrap. Observe the below code that comes with different positions of popovers

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

And your HTML as,

```
<div>
    <ul class="list-inline">
        <li><a href="#" class="btn btn-primary popover-left" data-toggle="popover" title="DotNetFunda" data-content="Fundamentals of .Net">DotNet Funda</a></li>
        <li><a href="#" class="btn btn-success popover-top" data-toggle="popover" title="ITFundu" data-content="IT Training & Placement">IT Funda</a></li>
        <li><a href="#" class="btn btn-warning popover-bottom" data-toggle="popover" title="KidsFunda" data-content="Fundamentals for Kids">Kids Funda</a></li>
        <li><a href="#" class="btn btn-danger popover-right" data-toggle="popover" title="FundooVideo" data-content="Vidoes you are looking">Fundoo Video</a></li>
    </ul>
</div>
```

If you observe JavaScript above, we have different placements like **top**, **bottom**, **right** and **left**. This combination of code will gives you the following result



Popover actions from JavaScript:

We have `show`, `hide`, `toggle` and `destroy` properties from which we can do actions manually for popovers.

Take a look at the below code to understand them clearly

```
<script type="text/javascript">
$(document).ready(function () {
    $(".show-popover").click(function () {
        $(".popover-pop a").popover('show');
    });
    $(".hide-popover").click(function () {
        $(".popover-pop a").popover('hide');
    });
    $(".toggle-popover").click(function () {
        $(".popover-pop a").popover('toggle');
    });
    $(".destroy-popover").click(function () {
        $(".popover-pop a").popover('destroy');
    });
});
</script>
```

And HTML,

```
<div>
    <p class="popover-pop list-inline">
        <a href="#" class="btn btn-info" data-toggle="popover" title="DotNetFunda" data-content="Fundamentals of .Net">DotNetFunda</a>
    </p>
    <div>
        <input type="button" class="btn btn-primary show-popover" value="Show" />
        <input type="button" class="btn btn-success hide-popover" value="Hide" />
        <input type="button" class="btn btn-warning toggle-popover" value="Toggle" />
        <input type="button" class="btn btn-danger destroy-popover" value="Destroy" />
    </div>
</div>
```

Check out this in your browser and you might see the following actions in your Output



These actions works through JavaScript of Bootstrap.

Conclusion

In this chapter, we have seen different types of popovers in Bootstrap. Hope you understand.

Thanks for reading.

Chapter-23

Alerts

Introduction

So far we have discussed about popovers in Bootstrap in previous chapter and now let's see some alerts in this one.

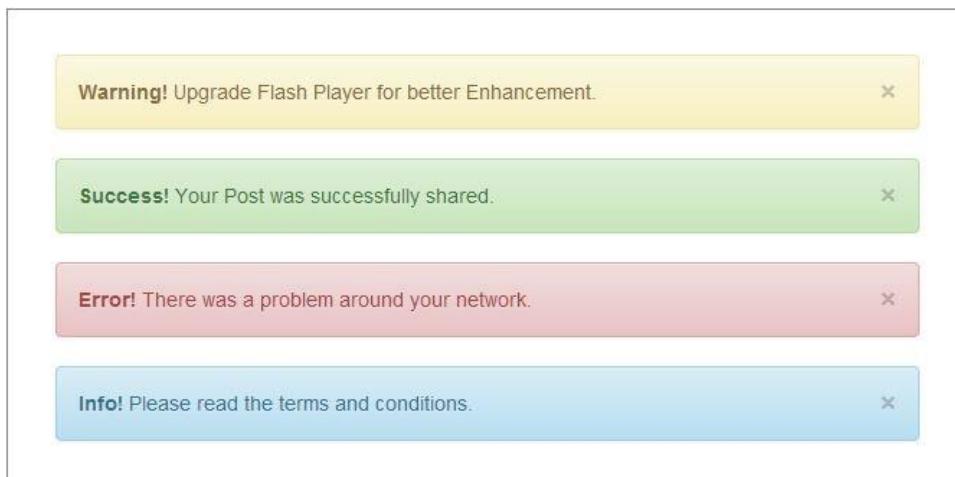
We have already covered alerts in [Chapter-16](#). Let's see them in detail here.

Using Bootstrap code

Let's start with some normal alerts that are available in Bootstrap before going in depth.

```
<div class="alert alert-warning">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Warning!</strong> Upgrade Flash Player for better Enhancement.
</div>
<div class="alert alert-success">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Success!</strong> Your Post was successfully shared.
</div>
<div class="alert alert-danger">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Error!</strong> There was a problem around your network.
</div>
<div class="alert alert-info">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Info!</strong> Please read the terms and conditions.
</div>
```

If you observe the above code, you will come to know that we are using different predefined styles such as `alert-warning`, `alert-success`, `alert-danger`, `alert-info` etc. Which gives you the following result in your browser,



Alerts with Data Attributes:

We can also use other data attributes such as buttons, span etc. for alerts. take a look at the below code in which we are using `span` and `button` instead of '`a`' tag.

```
<div class="alert alert-warning">
    <button type="button" class="close" data-dismiss="alert">&times;</button>
    <strong>Warning!</strong> This alert is using Button to close.
</div>
<div class="alert alert-warning">
    <span class="close" data-dismiss="alert">&times;</span>
    <strong>Warning!</strong> This alert is using Span to close.
</div>
```

Now run this code in your browser and you will see no change in your alert operations



Alert using Different Bootstrap Classes:

We can also create alerts using Bootstrap's predefined classes.

Using Well

Now let's see making alerts through `well` class of Bootstrap below,

```
<div class="well">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Success!</strong> Your Post was successfully shared.
</div>
```

Run this code in your browser to get output something like below

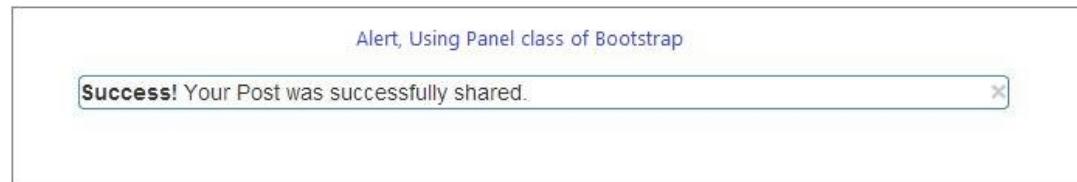


Using Panel

Observe the below code in which we are using `panel` class of Bootstrap instead of `alert`.

```
<div class="panel panel-primary">
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <strong>Success!</strong> Your Post was successfully shared.
</div>
```

Run the above code in the browser to get the following output



Alert using JavaScript:

We can also set alerts using JavaScript like below,

```
<script type="text/javascript">
$(document).ready(function () {
    $(".MyAlert").click(function () {
        $(".alert").alert();
    });
});
</script>
```

And HTML as,

```
<div class="MyAlert">
  <a href="#" class="close" data-dismiss="alert">&times;</a>
  <strong>Warning!</strong> Upgrade Flash Player for better Enhancement.
</div>
```

Run this code in your browser to see the alert message like below



Actually, the above alert fires without JavaScript but if we want to add any additional functionalities, we can do them from JavaScript.

Alert Events from JavaScript:

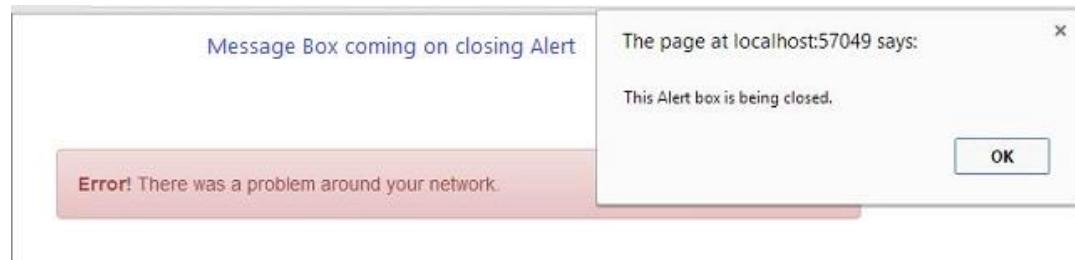
There are two alert events in Bootstrap for JavaScript which are `close.bs.alert` and `closed.bs.alert`. Both have same functionality with slight change that occurs when we use CSS Fade-in, Fade-out etc. actions. Take a look at the below example code of using Alert Event,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#Success").on('close.bs.alert', function () {
        alert("This Alert box is being closed.");
    });
});
</script>
```

And HTML as,

```
<div class="alert alert-danger" id="Success">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Error!</strong> There was a problem around your network.
</div>
```

Run this combination of code in your browser to see the alert event something like below as result,



Closing Alert from JavaScript:

We can also close or remove the alerts from JavaScript.

```
<script type="text/javascript">
$(document).ready(function () {
    $(".close").click(function () {
        $("#Danger").alert('close');
    });
});
</script>
```

In the above code we are using the function '`close`' for `alert` action which allows us to close the alert message.

And HTML is,

```
<div class="alert alert-danger" id="Danger">
  <a href="#" class="close">&times;</a>
  <strong>Error!</strong> There was a problem around your network.
</div>
```

Examine the result by running this combination in your browser which gives you following output



Closing Multiple Alerts at one time:

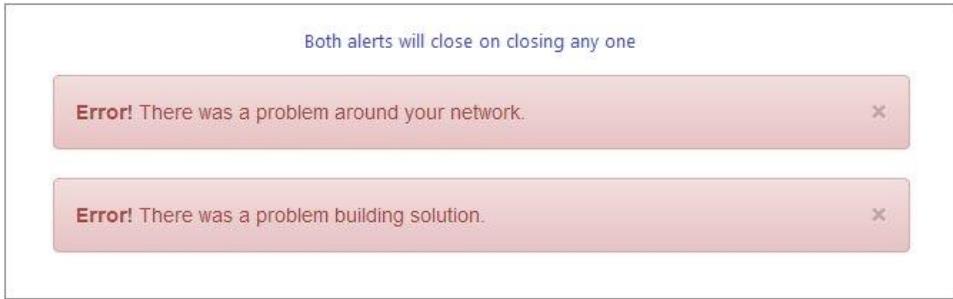
If you want to close all the alert messages at once, follow this approach

```
<script type="text/javascript">
$(document).ready(function () {
    $(".close").click(function () {
        $("#Danger").alert('close');
    });
});
</script>
```

And your HTML,

```
<div id="Danger">
  <div class="alert alert-danger">
    <a href="#" class="close">&times;</a>
    <strong>Error!</strong> There was a problem around your network.
  </div>
  <div class="alert alert-danger">
    <a href="#" class="close">&times;</a>
    <strong>Error!</strong> There was a problem building solution.
  </div>
</div>
```

This combination will give you the following output in which you can close multiple alerts at once



Conclusion

In this chapter, we have seen different types of alerts using Bootstrap in detail. Hope you understand.

Thanks for reading.

Chapter-24

Tabs with Java Script

Introduction

We have already seen Tabs and Pills in Chapter-9 and 10. Let's see using them with Bootstrap's JavaScript.

Using Bootstrap code

Let's see creating simple tabs using Bootstrap first,

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

This will create a Tab like below,



Tabs with Data Attributes:

We can also create tabs with data attributes that works like menus. Please refer the below code for clear view

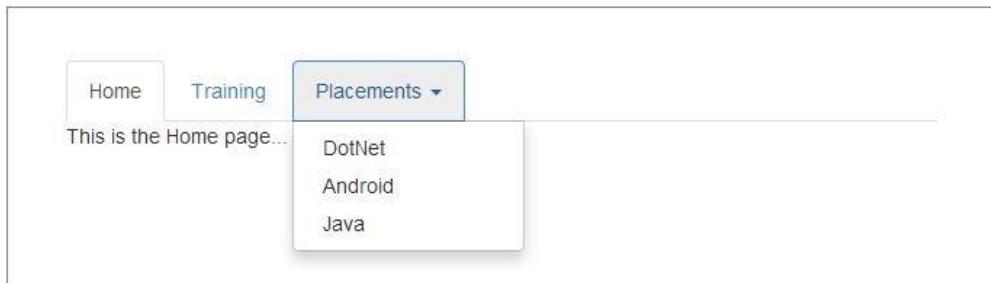
```
<div>
  <ul class="nav nav-tabs">
    <li class="active"><a data-toggle="tab" href="#Home">Home</a></li>
    <li><a data-toggle="tab" href="#Training">Training</a></li>
    <li class="dropdown">
      <a data-toggle="dropdown" class="dropdown-toggle" href="#">Placements <b class="caret"></b></a>
      <ul class="dropdown-menu">
        <li><a data-toggle="tab" href="#dotnet">DotNet</a></li>
        <li><a data-toggle="tab" href="#android">Android</a></li>
        <li><a data-toggle="tab" href="#java">Java</a></li>
      </ul>
    </li>
  </ul>
  <div class="tab-content">
    <div id="Home" class="tab-pane fade in active">
      <p>This is the Home page...</p>
    </div>
    <div id="Training" class="tab-pane fade">
```

```

        <p>Please select your option regarding on Training...</p>
    </div>
    <div id="dotnet" class="tab-pane fade">
        <p>Welcome to DotNet Placements Section...</p>
    </div>
    <div id="android" class="tab-pane fade">
        <p>Welcome to Android Placements Section...</p>
    </div>
    <div id="java" class="tab-pane fade">
        <p>Welcome to Java Placements Section...</p>
    </div>
</div>

```

In the above code, we have used the data attribute `data-toggle` to switch the Tabs. Run the code in your browser to see the following result



Tabs with JavaScript:

We can also switch the tabs without using data attributes. For that we have to write a little JavaScript something like below,

```

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

```

And we are removing `data-toggle` from the above HTML code as below,

```

<div>
    <ul class="nav nav-tabs" id="tabMenu">
        <li class="active"><a href="#Home">Home</a></li>
        <li><a href="#Training">Training</a></li>
        <li class="dropdown">
            <a data-toggle="dropdown" class="dropdown-toggle" href="#">Placements <b
class="caret"></b></a>

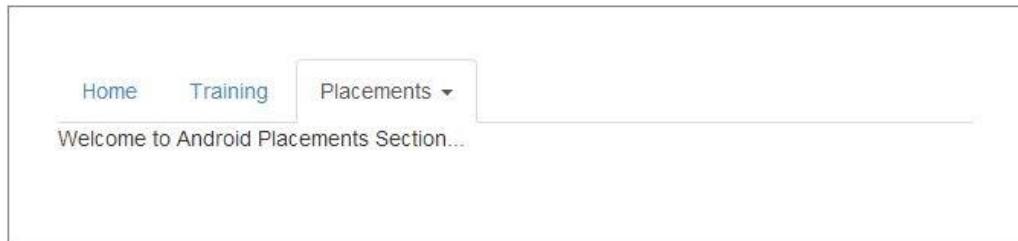
```

```

        <ul class="dropdown-menu">
            <li><a href="#dotnet">DotNet</a></li>
            <li><a href="#android">Android</a></li>
            <li><a href="#java">Java</a></li>
        </ul>
    </li>
</ul>
<div class="tab-content">
    <div id="Home" class="tab-pane fade in active">
        <p>This is the Home page...</p>
    </div>
    <div id="Training" class="tab-pane fade">
        <p>Please select your option regarding on Training...</p>
    </div>
    <div id="dotnet" class="tab-pane fade">
        <p>Welcome to DotNet Placements Section...</p>
    </div>
    <div id="android" class="tab-pane fade">
        <p>Welcome to Android Placements Section...</p>
    </div>
    <div id="java" class="tab-pane fade">
        <p>Welcome to Java Placements Section...</p>
    </div>
</div>
</div>

```

Run this combination in your browser to perform switching Tabs without Data Attributes because of JavaScript



Showing Tab targeted by Selector:

In order to show our tab manually on start or reload, we have to add the following script to the above code

```

<script type="text/javascript">
$(document).ready(function () {
    $('#tabMenu a[href="#java"]').tab('show'); // showing the tab targeted by the selector
});
</script>

```

After adding this code, you will get '**java**' section by default like below

The screenshot shows a navigation bar with three tabs: 'Home', 'Training', and 'Placements'. The 'Placements' tab is highlighted with a darker background and a dropdown arrow icon. Below the navigation bar, the text 'Welcome to Java Placements Section...' is displayed. At the top of the page, there is a message: "'Java' section shown by default because of the Script".

Note: You have to include both Scripts to make it work.

Showing First tab by Default:

In order to show the First tab on page reload, Add the below script to the main source

```
<script type="text/javascript">
$(document).ready(function () {
    $("#tabMenu a:first").tab('show'); // It show the first tab on Reload
});
</script>
```

Now run the code and you will see the first tab in active as below,

The screenshot shows a navigation bar with three tabs: 'Home', 'Training', and 'Placements'. The 'Home' tab is highlighted with a darker background. Below the navigation bar, the text 'This is the Home page...' is displayed. At the top of the page, there is a message: 'First tab is shown on Page Reload'.

Showing Last tab by Default:

If we need to show the last tab on default, Change the Script like below,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#tabMenu a:last").tab('show'); // It show the Last tab on Reload
});
</script>
```

Run this code in your browser to see the Last Tab by default

The screenshot shows a navigation bar with three tabs: 'Home', 'Training', and 'Placements'. The 'Placements' tab is highlighted with a darker background. Below the navigation bar, the text 'Welcome to Java Placements Section...' is displayed. At the top of the page, there is a message: 'Last tab is shown on Page reload'.

Showing Tabs based on Index:

We can also show the Default Tab by using Index like in the below script,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#tabMenu li:eq(1) a").tab('show'); // It shows second tab (0-indexed, like an
array) on Reload
});
</script>
```

Now the above script will shows you the second tab by default in your browser as below,



Conclusion

In this chapter, we have looked into Tabs with JavaScript in Bootstrap. Hope you understand.

Thanks for reading.

Chapter-25

Modals

Introduction

Modals are the form fields which can be shown as dialog or a popup. Bootstrap comes with some predefined Modal Classes that you might see in `bootstrap.css`.

Using Bootstrap code

As we said above, there are so many styles regarding to Modals in `bootstrap.css` which are used to create modals in our application.

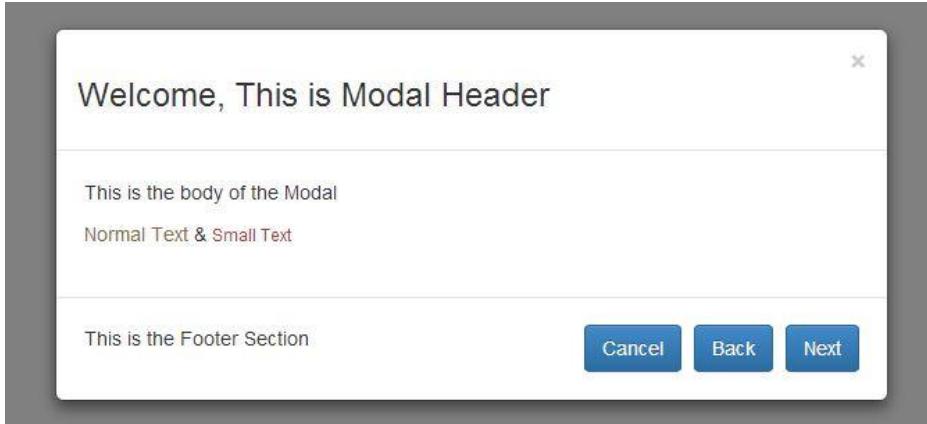
Let's see creating Modal with Data attributes as shown in below code

```
<a href="#modelSave" role="button" class="btn btn-success" data-toggle="modal">Click to Show Modal</a>

<div id="modelSave" class="modal fade">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
        <h3>Welcome, This is Modal Header</h3>
      </div>
      <div class="modal-body">
        <p>This is the body of the Modal</p>
        <span class="text-warning">Normal Text</span> & <span class="text-danger"><small>Small Text</small></span>
      </div>
      <div class="modal-footer">
        <span class="pull-left">This is the Footer Section</span>
        <a href="#" class="btn btn-primary">Cancel</a>
        <a href="#" class="btn btn-primary">Back</a>
        <a href="#" class="btn btn-primary">Next</a>
      </div>
    </div>
  </div>
</div>
```

Observe the above code clearly where we have used so many modal related classes like `heading`, `body`, `footer`, `content` etc.

If you run this you might see the output something like below in your browser,



We can close this as we do for Alerts.

Modal using JavaScript:

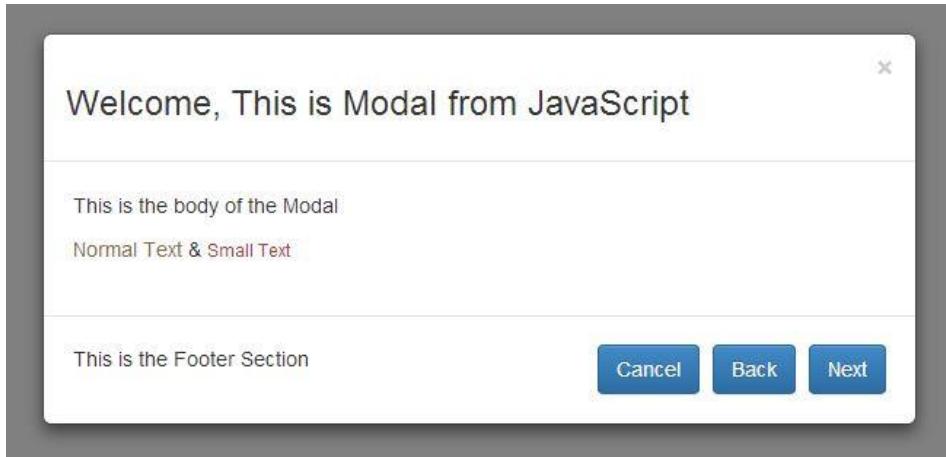
We can also create the same Modal shown above by using a little Script as shown below,

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

And the HTML as,

```
<div id="modalSave" class="modal fade">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
                <h3>Welcome, This is Modal from JavaScript</h3>
            </div>
            <div class="modal-body">
                <p>This is the body of the Modal</p>
                <span class="text-warning">Normal Text</span> & <span class="text-danger"><small>Small Text</small></span>
            </div>
            <div class="modal-footer">
                <span class="pull-left">This is the Footer Section</span>
                <a href="#" class="btn btn-primary">Cancel</a>
                <a href="#" class="btn btn-primary">Back</a>
                <a href="#" class="btn btn-primary">Next</a>
            </div>
        </div>
    </div>
</div>
```

Now run this combination of code in your browser to get the same result as above



Hiding Modal:

In order to hide the model, just use '**Hide**' instead of '**Show**' in the above Script i.e,

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

This will hides your Modal and doesn't shows anything.

Toggle Function:

There is a toggle function in the JavaScript for Modals of Bootstrap. In order to use that, you have to follow the below script,

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

This function will toggles the action of Modal for the Modal Firing element such as Button etc.

KeyBoard Function:

There is another function called **keyboard** (boolean) for Modals which makes use of Keyboard Keys such as **Esc** to close the Modal

```
<script type="text/javascript">
$(document).ready(function () {
    $(".launch-modal").click(function () {
```

```
$("#modalSave").modal({
    keyboard: false
});
});
```

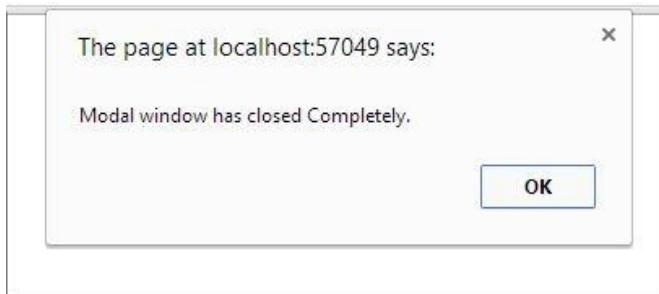
By default it is set to true and use the above Script to make it disabled.

Setting Alert from JavaScript:

We can also set the Alert messages from JavaScript to the Modals. Take a look at below script in which we are setting alert on closing the Modal,

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

This will shows an Alert message after closing your Modal like below,



Use `shown.bs.modal` event instead of `hidden.bs.modal` to show the alert on Modal StartUp.

There is another function called `backdrop` (Boolean), that includes the modal backdrop element. If you specifies `static` to the `backdrop` function, that doesn't closes modal on clicking.

You might also use `modal-lg` and `modal-sm` classes to create larger and smaller modals respectively.

Conclusion

In this chapter, we have seen different types of creating Modals by using Bootstrap. Hope you understand it.

Thanks for reading.

Chapter-26

Accordion and Collapse

Introduction

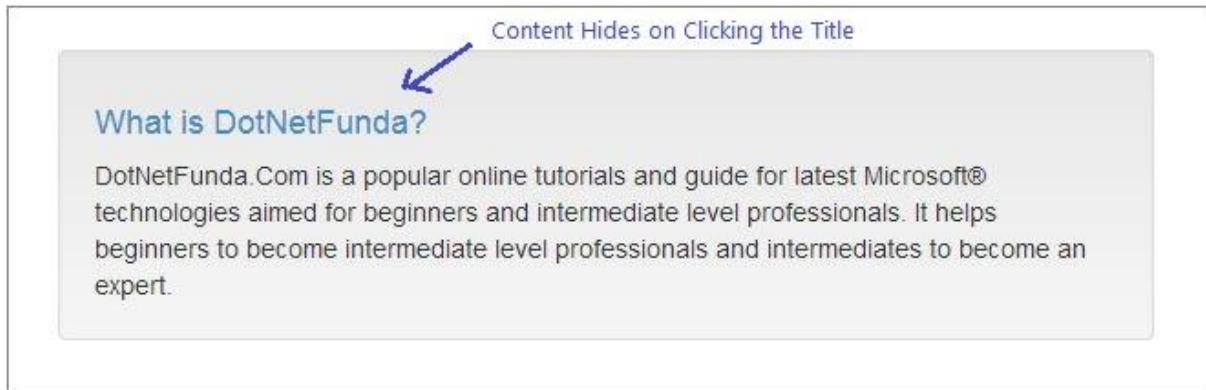
Accordion is the Bootstrap's predefined Style with JavaScript that will allows user to Show or Hide the particular content on his wish.

Using Bootstrap code

Lets start by creating a simple Collapsible form using the below example,

```
<div class="well" id="accordion">
  <div>
    <h4><a data-toggle="collapse" data-parent="#accordion" href="#collapseDNF">What
is DotNetFunda?</a></h4>
  </div>
  <div id="collapseDNF" class="collapse in"> <!-- By default it shows the content
below. remove 'in' to hide the content by default -->
    DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft®
technologies aimed for beginners and intermediate level professionals. It helps beginners
to become intermediate level professionals and intermediates to become an expert.
  </div>
</div>
```

In the above code, we are keeping our entire content in a **Well**. Run this snip in your browser to see the following result



In the above output, Content toggles (Show/Hide) on clicking the title "What is DotNetFunda?"

Using Panels:

As we kept in the well in above example, we can also keep the accordion in the Panels like below,

```

<div class="panel-group" id="accordion">
    <div class="panel panel-primary">
        <div class="panel-heading">
            <h4 class="panel-title">
                <a data-toggle="collapse" data-parent="#accordion" href="#collapseDNF">
                    What is DotNetFunda?
                </a>
            </h4>
        </div>
        <div id="collapseDNF" class="panel-collapse collapse in">
            <div class="panel-body">
                DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. It helps beginners to become intermediate level professionals and intermediates to become an expert.
            </div>
        </div>
    </div>
    <div class="panel panel-primary">
        <div class="panel-heading">
            <h4 class="panel-title">
                <a data-toggle="collapse" data-parent="#accordion" href="#collapseITF">
                    What is ITFunda?
                </a>
            </h4>
        </div>
        <div id="collapseITF" class="panel-collapse collapse">
            <div class="panel-body">
                ITFunda.Com is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related services. It typically hosts study materials, software products / services for buying/selling.
            </div>
        </div>
    </div>
    <div class="panel panel-primary">
        <div class="panel-heading">
            <h4 class="panel-title">
                <a data-toggle="collapse" data-parent="#accordion" href="#collapseKF">
                    What is KidsFunda?
                </a>
            </h4>
        </div>
        <div id="collapseKF" class="panel-collapse collapse">
            <div class="panel-body">
                KidsFunda is the website designed to give the learning fundamentals for kids.
            </div>
        </div>
    </div>
</div>

```

In the above code, we have taken three panels and grouped them by using **panel-group** class of Bootstrap. Run and see the above code in your browser which looks like something as shown below,

What is DotNetFunda?

What is ITFundा?

ITFundा.Com is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related services. It typically hosts study materials, software products / services for buying/selling.

What is KidsFundा?

Using Data Attributes:

We can also use Data attributes such as buttons to create collapsible content. Take a look at the below code snippet that clears on this,

```
<input type="button" class="btn btn-warning" data-toggle="collapse" data-target="#toggleDNF" value="DotNetFundा">
<hr />
<div id="toggleDNF" class="collapse in">
    <p>DotNetFundा.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. It helps beginners to become intermediate level professionals and intermediates to become an expert.</p>
</div>
```

The above code will gives the following output in your browser,

A screenshot of a web browser window. On the left, there is an orange button with the text "DotNetFundा". To the right of the button, a blue arrow points from the text "Clicking this Button" to the button itself. Above the button, the text "Toggles Show/Hide on" is displayed. Below the button, a horizontal line separates it from a collapsed content area. The content area contains the text: "DotNetFundा.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. It helps beginners to become intermediate level professionals and intermediates to become an expert."

Here, we can change the status of the content using Button.

Using JavaScript:

Now, if we want to work out the same function using JavaScript, please follow below approach

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

And HTML as,

```
<input type="button" class="btn btn-primary" value="Toggle Content" />


Run this combination in your browser to experience the same result as we got with Data attributes,



A screenshot of a web browser window. At the top, there is a blue header bar with the text "Toggles Show/Hide Because of JavaScript". Below the header is a button with the text "Toggle Content" in white. Underneath the button is a yellow-bordered box containing the text: "DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. It helps beginners to become intermediate level professionals and intermediates to become an expert." The text is displayed in a black font.



### toggle:



By default, the toggle function is set to true. But if we need to disable the toggle function, just make it to false like below,



```
<script type="text/javascript">
$(document).ready(function () {
 $(".btn").click(function () {
 $("#toggleDNF").collapse({ toggle: false });
 });
});
</script>
```



This will disables the toggling functionality in your browser for the particular id element.


```

show:

This parameter will shows your Collapsing content permanently. Just replace **show** function instead of **toggle** in the above script which looks as,

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

hide:

Similarly there is a **hide** parameter which Hides your content in the browser. Replace **show** with **hide** in the above JavaScript i.e,

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

Showing Alerts on Toggling Accordion (Events):

We can also show the JavaScript Alerts on Toggling Content. Observe the below Script in which we are setting an Alert message on Hiding the Content.

```
<script type="text/javascript">
$(document).ready(function () {
    $("#toggleDNF").on('hidden.bs.collapse', function () {
        alert("Accordion element has been completely closed.");
    });
});
</script>
```

And HTML as,

```
<input type="button" class="btn btn-danger" data-toggle="collapse" data-
target="#toggleDNF" value="Toggle Button" />
<div id="toggleDNF" class="collapse in" style="border:1px solid
orange;padding:5px;margin-top:5px;">
    DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft®
    technologies aimed for beginners and intermediate level professionals. It helps beginners
    to become intermediate level professionals and intermediates to become an expert.
</div>
```

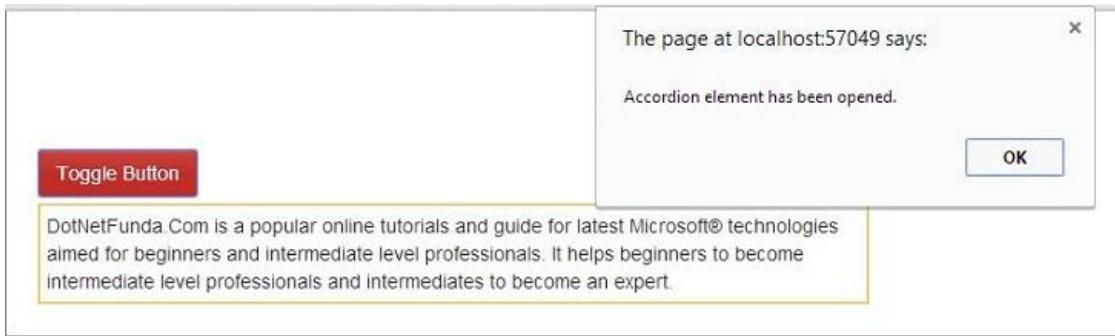
Now Run this combination of code in your browser and try to close/hide the content by clicking button. You will see the following Alert message then



And if you want to show the alert message on showing the content, just modify the above JavaScript like below,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#toggleDNF").on('shown.bs.collapse', function () {
        alert("Accordion element has been opened.");
    });
});
</script>
```

'shown.bs.collapse' will sets any action on Showing the content as in the below output



Conclusion

In this chapter, we have seen different types of Collapsible content in Bootstrap. Hope you understand.

Thanks for reading.

Chapter-27

Carousel Sliders

Introduction

We have seen Sliding Images in many websites as because of using JQuery. Bootstrap comes with predefined CSS and JavaScript for the implementation of Sliding Carousels.

Using Bootstrap code

There are some predefined classes in `bootstrap.css` on behalf of Carousel. We are going to use them to create a Slider. Here, we are not using any Images and so please add below code of styles in your page,

```
<style>
    .item {
        background: #333;
        text-align: center;
        height: 300px !important;
    }
    h2 {
        margin: 0;
        color: #888;
        padding-top: 100px;
        font-size: 50px;
    }
</style>
```

In the above style, we are creating an `item` with `background` and `height` that we are going to use as the default Image.

Lets create a Carousel Sliders with the below code,

```
<div class="thumbnail">
    <div id="DemoCarousel" class="carousel slide" data-interval="2000" data-
ride="carousel">
        <!-- Carousel indicators -->
        <ol class="carousel-indicators">
            <li data-target="#DemoCarousel" data-slide-to="0" class="active"></li>
            <li data-target="#DemoCarousel" data-slide-to="1"></li>
            <li data-target="#DemoCarousel" data-slide-to="2"></li>
        </ol>
        <!-- Carousel items -->
        <div class="carousel-inner">
            <div class="item active">
                <h2>Slide 1</h2>
                <div class="carousel-caption">
                    <h3>This is the First Label</h3>
                    <p>The Content of the First Slide goes in here</p>
                </div>
            </div>
            <div class="item">
```

```

        <h2>Slide 2</h2>
        <div class="carousel-caption">
            <h3>This is the Second Label</h3>
            <p>The Content of the second Slide goes in here</p>
        </div>
    </div>
    <div class="item">
        <h2>Slide 3</h2>
        <div class="carousel-caption">
            <h3>This is the Third Label</h3>
            <p>The Content of the Third Slide goes in here</p>
        </div>
    </div>
    <!-- Carousel Controls -->
    <a class="carousel-control left" href="#DemoCarousel" data-slide="prev">
        <span class="glyphicon glyphicon-chevron-left"></span>
    </a>
    <a class="carousel-control right" href="#DemoCarousel" data-slide="next">
        <span class="glyphicon glyphicon-chevron-right"></span>
    </a>
</div>
</div>

```

If you observe the above code, we have used different types of Carousel elements such as `carousel slide`, `carousel-indicators`, `carousel-inner`, and `carousel-caption` and `carousel control`.

And also `data-interval` for setting Duration.

Now run the above code in your browser and you will see the Sliding Carousel something shown in below figure



Using JavaScript:

We can also create the above Sliding Carousel by using JavaScript. For that we have to write little script to set actions as below,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#DemoCarousel").carousel();

    //Carousel Controls
    $(".left").click(function () {
        $("#DemoCarousel").carousel('prev');
    });
    $(".right").click(function () {
        $("#DemoCarousel").carousel('next');
    });

    //Carousel Indicators
    $(".slide-1").click(function () {
        $("#DemoCarousel").carousel(0);
    });
    $(".slide-2").click(function () {
        $("#DemoCarousel").carousel(1);
    });
    $(".slide-3").click(function () {
        $("#DemoCarousel").carousel(2);
    });
});
</script>
```

In the above script, we have used Carousel functions like `prev`, `next`, and the numbers that indicates Index of the Slide.

Now, you can add your HTML code like below,

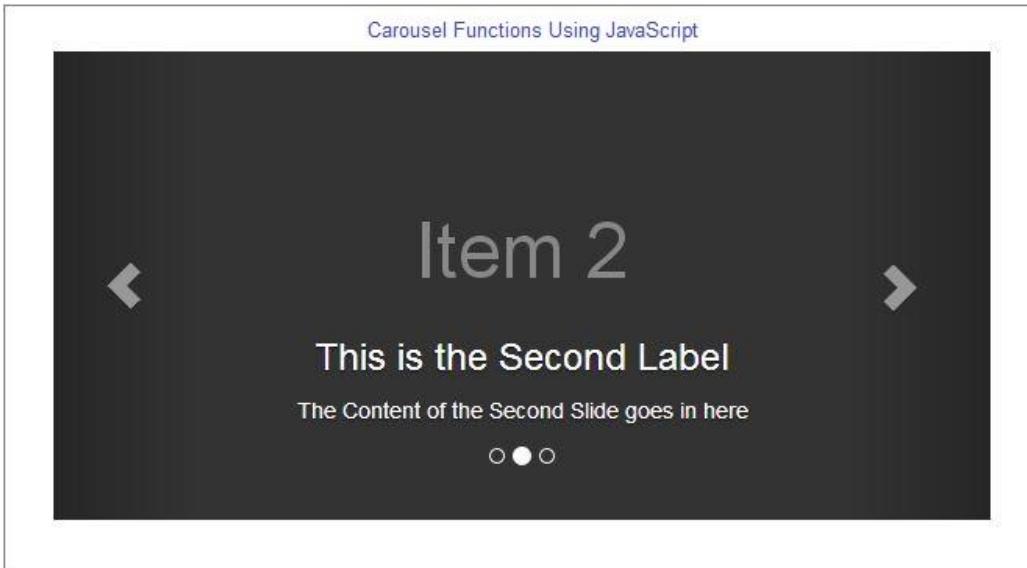
```
<div id="DemoCarousel" class="carousel slide" data-interval="2000" data-ride="carousel">
    
    <ol class="carousel-indicators">
        <li class="slide-1 active"></li>
        <li class="slide-2"></li>
        <li class="slide-3"></li>
    </ol>
    
    <div class="carousel-inner">
        <div class="item active">
            <h2>Item 1</h2>
            <div class="carousel-caption">
                <h3>This is the First Label</h3>
                <p>The Content of the First Slide goes in here</p>
            </div>
        </div>
        <div class="item">
            <h2>Item 2</h2>
            <div class="carousel-caption">
                <h3>This is the Second Label</h3>
                <p>The Content of the Second Slide goes in here</p>
            </div>
        </div>
    </div>
</div>
```

```

        </div>
    </div>
    <div class="item">
        <h2>Item 3</h2>
        <div class="carousel-caption">
            <h3>This is the Third Label</h3>
            <p>The Content of the Third Slide goes in here</p>
        </div>
    </div>
<!-- Controls -->
<a class="carousel-control left">
    <span class="glyphicon glyphicon-chevron-left"></span>
</a>
<a class="carousel-control right">
    <span class="glyphicon glyphicon-chevron-right"></span>
</a>
</div>

```

Now run this combination in your browser to expect the same result like above which looks as



Pause:

There is a pause function for Carousel that is used to pause the sliding motion for a bit.

```

<script type="text/javascript">
$(document).ready(function () {
    $("#DemoCarousel").carousel({
        pause: 'hover'
    });
});
</script>

```

In the above script, we are setting '`hover`' which pauses the Carousel Cycling on Mouse Enter and resumes again on Mouse Leave.

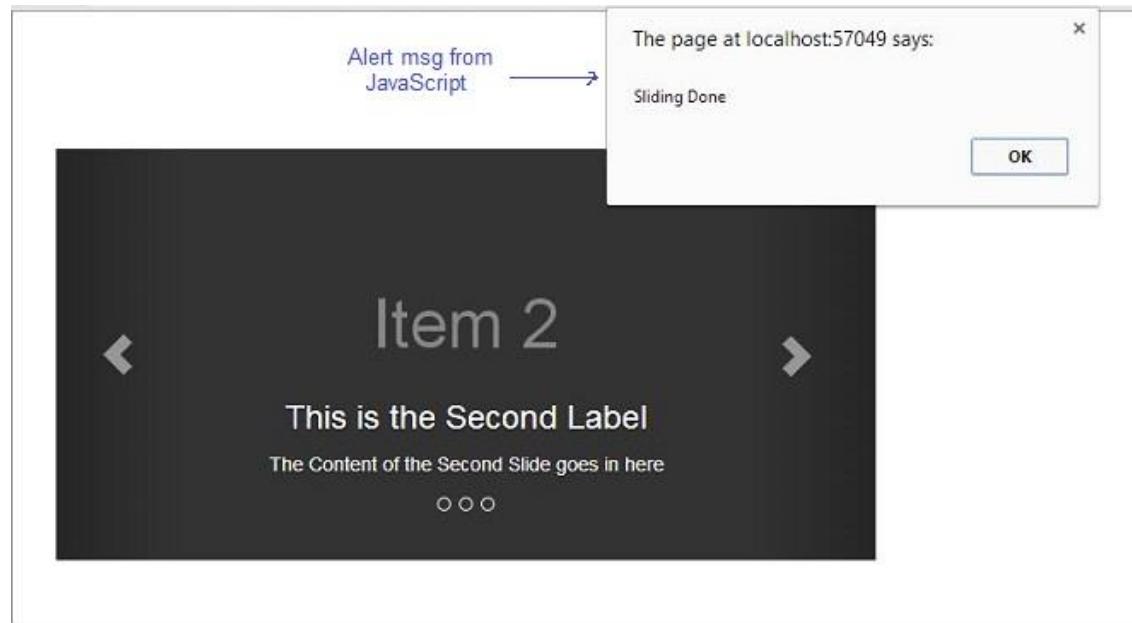
Showing Alert on Sliding:

We can also display an alert message after the completion of sliding. Please look the below script

```
<script type="text/javascript">
$(document).ready(function () {
    $('#DemoCarousel').on('slid.bs.carousel', function () {
        alert("Sliding Done");
    });
});
</script>
```

`slid.bs.carousel` is the event that fires after the completion of slide transition and `slide.bs.carousel` which fires immediately `slide` instance method is invoked.

If you add the above script and run, you will see the output something like below



After the slide transition, it shows JavaScript's `alert` message.

Button Control Functions of Carousel:

We can also set the buttons to control the Carousel actions. Take a look at the below code and use that in your view,

```

<script type="text/javascript">
$(document).ready(function () {
    $("#DemoCarousel").carousel();

    //Carousel Actions
    $(".start").click(function () {
        $("#DemoCarousel").carousel('cycle');
    });
    $(".pause").click(function () {
        $("#DemoCarousel").carousel('pause');
    });

    //Carousel Controls
    $(".prevSlide").click(function () {
        $("#DemoCarousel").carousel('prev');
    });
    $(".nextSlide").click(function () {
        $("#DemoCarousel").carousel('next');
    });

    //Carousel Indicators
    $(".slide-1").click(function () {
        $("#DemoCarousel").carousel(0);
    });
    $(".slide-2").click(function () {
        $("#DemoCarousel").carousel(1);
    });
    $(".slide-3").click(function () {
        $("#DemoCarousel").carousel(2);
    });
});
</script>

```

The above script demonstrates the Carousel functions that are being used from Bootstrap.

Now HTML as,

```

<div id="DemoCarousel" class="carousel slide" data-interval="2000" data-ride="carousel">
    <div class="carousel-inner">
        <div class="item active">
            <h2>Item 1</h2>
            <div class="carousel-caption">
                <h3>This is the First Label</h3>
                <p>The Content of the First Slide goes in here</p>
            </div>
        </div>
        <div class="item">
            <h2>Item 2</h2>
            <div class="carousel-caption">
                <h3>This is the Second Label</h3>
                <p>The Content of the Second Slide goes in here</p>
            </div>
        </div>
        <div class="item">
            <h2>Item 3</h2>

```

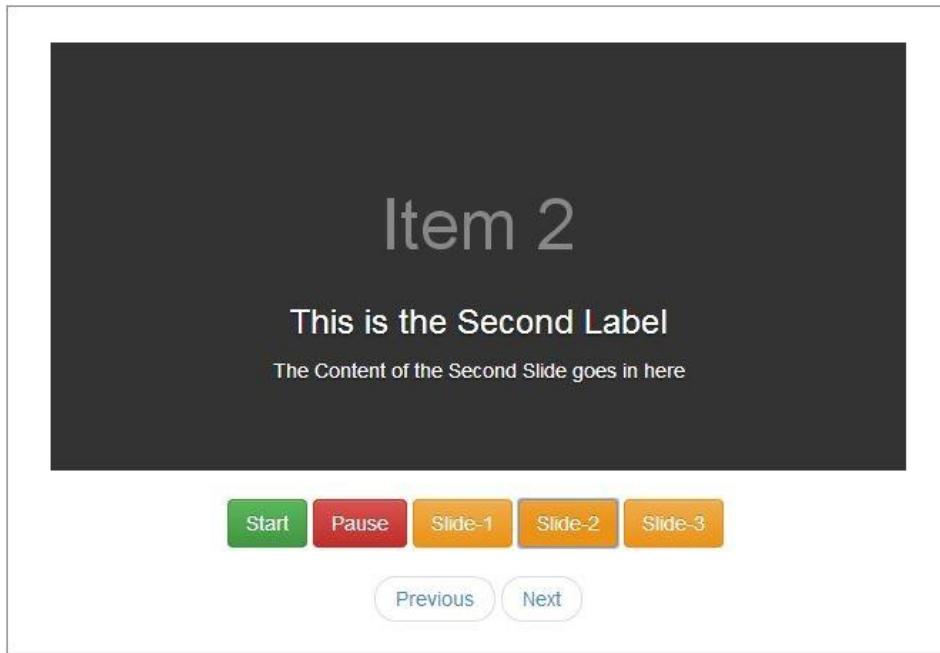
```

        <div class="carousel-caption">
            <h3>This is the Third Label</h3>
            <p>The Content of the Third Slide goes in here</p>
        </div>
    </div>
<br />

<!-- Control Buttons -->
<div style="text-align:center;">
    <input type="button" class="btn btn-success start" value="Start" />
    <input type="button" class="btn btn-danger pause" value="Pause" />
    <input type="button" class="btn btn-warning slide-1" value="Slide-1" />
    <input type="button" class="btn btn-warning slide-2" value="Slide-2" />
    <input type="button" class="btn btn-warning slide-3" value="Slide-3" />
    <br />
    <ul class="pager">
        <li class="prevSlide"><a href="#">Previous</a></li>
        <li class="nextSlide"><a href="#">Next</a></li>
    </ul>
</div>

```

Run this combination of code to create a beautiful Carousel along with controlling buttons like below,



Conclusion

In this chapter, we have seen creating Carousels in different ways using Bootstrap. Hope you understand.

Thanks for reading.

Chapter-28

Scroll Spy

Introduction

Twitter Bootstrap consists of predefined JavaScript to create ScrollSpys.

Using Bootstrap code

Please add the below Style code in your page first in order to understand the ScrollSpy easily.

```
<style>
    .scrollspy-example {
        height:250px;
        overflow:auto;
        position:relative;
    }
</style>
```

You will get it soon of adding above style in our page.

Now lets go and create some sections with ScrollSpy like in the below example code,

```
<div>
    <nav id="navbarScroll" class="navbar navbar-inverse" role="navigation">
        <div class="container-fluid">
            <div class="navbar-header">
                <button class="navbar-toggle" type="button" data-toggle="collapse" data-target=".navbar-scrollspy">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">ScrollSpy</a>
            </div>
            <div class="collapse navbar-collapse navbar-scrollspy">
                <ul class="nav navbar-nav">
                    <li><a href="#Dnf">DotNetFunda</a></li>
                    <li><a href="#Itf">ITFunda</a></li>
                    <li class="dropdown">
                        <a href="#" id="navbarDrop1" class="dropdown-toggle" data-toggle="dropdown">Sports <span class="caret"></span></a>
                        <ul class="dropdown-menu" role="menu" aria-labelledby="navbarDrop1">
                            <li><a href="#football" tabindex="-1">FootBall</a></li>
                            <li><a href="#cricket" tabindex="-1">Cricket</a></li>
                            <li class="divider"></li>
                            <li><a href="#hockey" tabindex="-1">Hockey</a></li>
                            <li><a href="#tennis" tabindex="-1">Tennis</a></li>
                        </ul>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
</div>
```

```
        </ul>
    </div>
</div>
<div data-spy="scroll" data-target="#navbarScroll" data-offset="0" class="scrollspy-example">
    <h1>Education:</h1>
    <hr />
    <h3 id="Dnf">DotNetFunda</h3>
    <p>
        DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. We help beginners to become intermediate level professionals and help intermediate level professionals to become an expert. By following our different sections regularly, we hope you will soon become one of the 'Most Valuable' professional and start shining in your career.
    </p>
    <h3 id="Itf">ITFundा</h3>
    <p>
        ITFundा.Com is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related services. We typically host study materials, software products / services for buying/selling. This website has been brought to you by IT Funda Corporation, a parent establishment of a popular Microsoft(R) technology related knowledge based website www.dotnetfunda.com.
    </p>
    <h1>Sports:</h1>
    <hr />
    <h3 id="football">FootBall</h3>
    <p>
        <b>Football</b> refers to a number of sports that involve, to varying degrees, kicking a ball with the foot to score a goal. The most popular of these sports worldwide is association football, more commonly known as just "football" or "soccer". Unqualified, the word football applies to whichever form of football is the most popular in the regional context in which the word appears, including association football, as well as American football, Australian rules football, Canadian football, Gaelic football, rugby league, rugby union,[1] and other related games. These variations of football are known as football codes.
    </p>
    <h3 id="cricket">Cricket</h3>
    <p>
        <b>Cricket</b> is a bat-and-ball game played between two teams of 11 players each on a field at the centre of which is a rectangular 22-yard long pitch. Each team takes its turn to bat, attempting to score runs, while the other team fields. Each turn is known as an innings.
    </p>
    <p>
        The bowler delivers the ball to the batsman who attempts to hit the ball with his bat away from the fielders so he can run to the other end of the pitch (which is counted as one run) without getting run out (the event in which the fielder throws the ball directly onto wickets or to a player who is near to wickets so he can dislodge them from the ground before the batsman or the non-striker has reached the crease). Each batsman (the other is called non-striker) continues batting until he is out. The batting team continues batting until ten batsmen are out or specified number of overs (defined as 6 countable balls bowled) have been bowled, at which point the teams switch roles and the fielding team comes in to bat.
    </p>
    <h3 id="hockey">Hockey</h3>
```

```

<p>
    <b>Hockey</b> is a family of sports in which two teams play against each
other by trying to maneuver a ball or a puck into the opponent's goal using a hockey
stick. In many areas, one sport (typically field hockey or ice hockey[1]) is generally
referred to simply as hockey.
</p>
<p>
    The first recorded use of the word "hockey" is from the 1773 book Juvenile
Sports and Pastimes, to Which Are Prefixed, Memoirs of the Author: Including a New Mode
of Infant Education, by Richard Johnson (Pseud. Master Michel Angelo), whose chapter XI
was titled "New Improvements on the Game of Hockey".[2] The belief that hockey was
mentioned in a 1363 proclamation by King Edward III of England is based on modern
translations of the proclamation, which was originally in Latin and explicitly forbade
the games "Pilam Manualem, Pedivam, & Bacularem: & ad Canibucam & Gallorum Pugnam". The
English historian and biographer John Strype did not use the word "hockey" when he
translated the proclamation in 1720.
</p>
<h3 id="tennis">Tennis</h3>
<p>
    <b>Tennis</b> is a sport people play individually against a single opponent
(singles) or between two teams of two players each (doubles). Each player uses a racquet
that is strung with cord to strike a hollow rubber ball covered with felt over or around
a net and into the opponent's court. The object of the game is to play the ball in such a
way that the opponent is not able to play a good return.
</p>
<p>
    Tennis is an Olympic sport and is played at all levels of society and at all
ages. The sport can be played by anyone who can hold a racquet, including wheelchair
users. The modern game of tennis originated in Birmingham, England, in the late 19th
century as "lawn tennis".[1] It had close connections both to various field ("lawn")
games such as croquet and bowls as well as to the older racquet sport of real tennis.
During most of the 19th-century in fact, the term "tennis" referred to real tennis, not
lawn tennis: for example, in Disraeli's novel Sybil (1845), Lord Eugene De Vere announces
that he will "go down to Hampton Court and play tennis."[2]
</p>
</div>
</div>

```

Do not get confused with the above code as we are using the `navbar` which we already seen in [Chapter-11](#). The important things are, we have to give the `data-targets` correctly and also need to set the perfect matches of `ID`'s.

Now, run this code in your browser to see the output as follows,

The screenshot shows a dark-themed website with a navigation bar at the top containing links for 'ScrollSpy', 'DotNetFunda', 'ITFundा', and 'Sports'. The main content area is titled 'DotNetFunda' and contains a paragraph of text about the site's purpose of providing Microsoft® technology tutorials for beginners and professionals. Below this is another section titled 'ITFundा' with its own descriptive text.

DotNetFunda

DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. We help beginners to become intermediate level professionals and help intermediate level professionals to become an expert. By following our different sections regularly, we hope you will soon become one of the 'Most Valuable' professional and start shining in your career.

ITFundा

ITFundा.Com is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related services. We typically host study materials, software products / services for buying/selling. This website has been

If you click on the **ITFundा** option in the **navbar**, you will be able to see the **ITFundा** section like below,

This screenshot shows the 'ITFundा' section of the website. It features a dark navigation bar with the same four links as the main site. The main content area is titled 'ITFundा' and contains a detailed paragraph about the website's purpose, mentioning it is an e-commerce platform for software and services, and is run by IT Funda Corporation. Below this is a section titled 'Sports:' followed by a single item, 'FootBall'.

ITFundा

ITFundा.Com is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related services. We typically host study materials, software products / services for buying/selling. This website has been brought to you by IT Funda Corporation, a parent establishment of a popular Microsoft(R) technology related knowledge based website www.dotnetfunda.com.

Sports:

FootBall

And if you go to '**Sports**' and select any sport, then also you will get the corresponding section such as,

Hockey

Hockey is a family of sports in which two teams play against each other by trying to maneuver a ball or a puck into the opponent's goal. One sport (typically field hockey or ice hockey[1]) is g...

The first recorded use of the word "hockey" is from the 1773 book Juvenile Sports and Pastimes, to Which Are Prefixed, Memoirs of the Author: Including a New Mode of Infant Education, by Richard Johnson (Pseud. Master Michel Angelo), whose chapter XI was titled "New Improvements on the Game of Hockey".[2] The belief that hockey was mentioned in a 1363 proclamation by King Edward III of England is based on modern translations of the proclamation, which was originally in Latin and explicitly forbade the games "Pilam Manualem, Pedivam, & Bacularem: & ad Canibucam & Gallorum Pugnam". The English...

Using JavaScript:

Below JavaScript is used to create the ScrollSpy,

```
<script type="text/javascript">
$(document).ready(function () {
    $(".scrollspy-example").scrollspy({ target: "#navbarScroll" })
});
</script>
```

`scrollspy` is the inbuilt method from Bootstrap's JavaScript.

Creating ScrollSpy with Events of JavaScript:

We have some more functionalities such as refresh, remove etc. which are shown in the below script,

```
<script type="text/javascript">
$(document).ready(function () {
    removeThis = function (e) {
        $(e).parents(".section").remove();
        $('[data-spy="scroll"]').each(function () {
            var $spy = $(this).scrollspy('refresh')
        });
    }
    $(".scrollspy-example").scrollspy({ target: "#navbarScroll" });
});
</script>
```

In the above script, we are using remove function which is used to remove a part or a section temporarily.

Add the below HTML along with this script,

```
<div>
    <nav id="navbarScroll" class="navbar navbar-inverse" role="navigation">
        <div class="container-fluid">
            <div class="navbar-header">
                <button class="navbar-toggle" type="button" data-toggle="collapse" data-target=".navbar-scrollspy">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">ScrollSpy</a>
            </div>
            <div class="collapse navbar-collapse navbar-scrollspy">
                <ul class="nav navbar-nav">
                    <li><a href="#Dnf">DotNetFunda</a></li>
                    <li><a href="#Itf">ITFunda</a></li>
                    <li class="dropdown">
                        <a href="#" id="navbarDrop1" class="dropdown-toggle" data-toggle="dropdown">Sports <span class="caret"></span></a>
                        <ul class="dropdown-menu" role="menu" aria-labelledby="navbarDrop1">
                            <li><a href="#football" tabindex="-1">FootBall</a></li>
                            <li><a href="#cricket" tabindex="-1">Cricket</a></li>
                            <li class="divider"></li>
                            <li><a href="#hockey" tabindex="-1">Hockey</a></li>
                            <li><a href="#tennis" tabindex="-1">Tennis</a></li>
                        </ul>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
    <div data-spy="scroll" data-target="#navbarScroll" data-offset="0" class="scrollspy-example">
        <h1>Education:</h1>
        <hr />
        <div class="section">
            <h3 id="Dnf">DotNetFunda <small><a href="#" onclick="removeThis(this)">Remove</a></small></h3>
            <p>
                DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. We help beginners to become intermediate level professionals and help intermediate level professionals to become an expert. By following our different sections regularly, we hope you will soon become one of the 'Most Valuable' professional and start shining in your career.
            </p>
        </div>
        <div class="section">
            <h3 id="Itf">ITFunda <small><a href="#" onclick="removeThis(this)">Remove</a></small></h3>
            <p>
                ITFunda.Com is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related
            </p>
        </div>
    </div>

```

services. We typically host study materials, software products / services for buying/selling. This website has been brought to you by IT Funda Corporation, a parent establishment of a popular Microsoft(R) technology related knowledge based website www.dotnetfunda.com.

```
</p>
</div>

<h1>Sports:</h1>
<hr />
<div class="section">
    <h3 id="football">FootBall <small><a href="#" onclick="removeThis(this)">Remove</a></small></h3>
    <p>
        <b>Football</b> refers to a number of sports that involve, to varying degrees, kicking a ball with the foot to score a goal. The most popular of these sports worldwide is association football, more commonly known as just "football" or "soccer". Unqualified, the word football applies to whichever form of football is the most popular in the regional context in which the word appears, including association football, as well as American football, Australian rules football, Canadian football, Gaelic football, rugby league, rugby union,[1] and other related games. These variations of football are known as football codes.
    </p>
    </div>
    <div class="section">
        <h3 id="cricket">Cricket <small><a href="#" onclick="removeThis(this)">Remove</a></small></h3>
        <p>
            <b>Cricket</b> is a bat-and-ball game played between two teams of 11 players each on a field at the centre of which is a rectangular 22-yard long pitch. Each team takes its turn to bat, attempting to score runs, while the other team fields. Each turn is known as an innings.
        </p>
        <p>
            The bowler delivers the ball to the batsman who attempts to hit the ball with his bat away from the fielders so he can run to the other end of the pitch (which is counted as one run) without getting run out (the event in which the fielder throws the ball directly onto wickets or to a player who is near to wickets so he can dislodge them from the ground before the batsman or the non-striker has reached the crease). Each batsman (the other is called non-striker) continues batting until he is out. The batting team continues batting until ten batsmen are out or specified number of overs (defined as 6 countable balls bowled) have been bowled, at which point the teams switch roles and the fielding team comes in to bat.
        </p>
        </div>
        <div class="section">
            <h3 id="hockey">Hockey <small><a href="#" onclick="removeThis(this)">Remove</a></small></h3>
            <p>
                <b>Hockey</b> is a family of sports in which two teams play against each other by trying to maneuver a ball or a puck into the opponent's goal using a hockey stick. In many areas, one sport (typically field hockey or ice hockey[1]) is generally referred to simply as hockey.
            </p>
            <p>
                The first recorded use of the word "hockey" is from the 1773 book Juvenile Sports and Pastimes, to Which Are Prefixed, Memoirs of the Author: Including a New Mode of Infant Education, by Richard Johnson (Pseud. Master Michel Angelo), whose chapter XI was titled "New Improvements on the Game of Hockey".[2] The belief that hockey
```

was mentioned in a 1363 proclamation by King Edward III of England is based on modern translations of the proclamation, which was originally in Latin and explicitly forbade the games "Pilam Manualem, Pedivam, & Bacularem: & ad Canibucam & Gallorum Pugnam". The English historian and biographer John Strype did not use the word "hockey" when he translated the proclamation in 1720.

```
</p>
</div>
<div class="section">
    <h3 id="tennis">Tennis <small><a href="#" onclick="removeThis(this)">Remove</a></small></h3>
    <p>
        <b>Tennis</b> is a sport people play individually against a single opponent (singles) or between two teams of two players each (doubles). Each player uses a racquet that is strung with cord to strike a hollow rubber ball covered with felt over or around a net and into the opponent's court. The object of the game is to play the ball in such a way that the opponent is not able to play a good return.
    </p>
    <p>
        Tennis is an Olympic sport and is played at all levels of society and at all ages. The sport can be played by anyone who can hold a racquet, including wheelchair users. The modern game of tennis originated in Birmingham, England, in the late 19th century as "lawn tennis".[1] It had close connections both to various field ("lawn") games such as croquet and bowls as well as to the older racquet sport of real tennis. During most of the 19th-century in fact, the term "tennis" referred to real tennis, not lawn tennis: for example, in Disraeli's novel Sybil (1845), Lord Eugene De Vere announces that he will "go down to Hampton Court and play tennis." [2]
    </p>
    </div>
</div>
</div>
```

Above HTML code is similar to the example we have seen earlier in this chapter in which adding the `remove` functions to different sections.

This combination will give you the output something like below,

The screenshot shows a web page with a dark header bar containing navigation links: ScrollSpy, DotNetFunda, ITFundA, and Sports ▾. Below the header, there are two main content areas. The first area is titled "DotNetFunda Remove" and contains the text: "DotNetFunda.Com is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. We help beginners to become intermediate level professionals and help intermediate level professionals to become an expert. By following our different sections regularly, we hope you will soon become one of the 'Most Valuable' professional and start shining in your career." The second area is titled "ITFundA Remove" and contains the text: "ITFundA.Com is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related services. We typically host study materials, software products / services for buying/selling. This website has been". Both sections include a "Remove" link next to their respective titles.

Observe the above output that, we can remove the sections temporarily by clicking on 'Remove'.

We have set 'Remove' function to all the sections i.e,

The screenshot shows a navigation bar with 'ScrollSpy', 'DotNetFunda', 'ITFundu', and a 'Sports' dropdown menu. Below the navigation, the word 'Cricket' is followed by a blue 'Remove' link. The main content area contains a paragraph about Cricket, mentioning the pitch, teams, and scoring. A scroll bar is visible on the right side of the content area.

Displaying Current Status:

We can also display the status of the current section. For that, please add the below script and a HTML tag to the above code,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#navbarScroll").on("activate.bs.scrollspy", function () {
        var currentView = $(".nav li.active > a").text();
        $("#show").empty().html("Currently you are at - " + currentView);
    });
});
</script>
```

'active.bs.scrollspy' is the predefined bootstrap Event which is responsible to get the current state of the ScrollSpy.

```
<h3 id="show" class="text-success">Currently you are at - DotNetFunda</h3>
```

Now, adding the above tag with 'show' id from above JavaScript will makes the script work in your page.

If you add these code to your page and run, you will see the Current state of ScrollSpy like in the below output,

The screenshot shows a web page with a navigation bar at the top containing links for ScrollSpy, DotNetFunda, ITFundu, and Sports. The 'Sports' link is highlighted with a dropdown menu open, showing categories like FootBall, Cricket, Tennis, and Badminton. Below the navigation, there are two main content blocks. The first block is titled 'FootBall Remove' and contains a paragraph about football. The second block is titled 'Cricket Remove' and contains a paragraph about cricket. At the bottom of the page, a green banner says 'Currently you are at - Sports FootBall'.

FootBall Remove

Football refers to a number of sports that involve, to varying degrees, kicking a ball with the foot to score a goal. The most popular of these sports worldwide is association football, more commonly known as just "football" or "soccer". Unqualified, the word football applies to whichever form of football is the most popular in the regional context in which the word appears, including association football, as well as American football, Australian rules football, Canadian football, Gaelic football, rugby league, rugby union,[1] and other related games. These variations of football are known as football codes.

Cricket Remove

Cricket is a bat-and-ball game played between two teams of 11 players each on a field at

Currently you are at - Sports FootBall

Try going to different sections like below,

The screenshot shows a web page with a navigation bar at the top containing links for ScrollSpy, DotNetFunda, ITFundu, and Sports. The 'Sports' link is highlighted with a dropdown menu open, showing categories like FootBall, Cricket, Tennis, and Badminton. Below the navigation, there are two main content blocks. The first block is titled 'Tennis Remove' and contains a paragraph about tennis. The second block is titled 'Currently you are at - Sports Tennis'.

Tennis Remove

Tennis is a sport people play individually against a single opponent (singles) or between two teams of two players each (doubles). Each player uses a racquet that is strung with cord to strike a hollow rubber ball covered with felt over or around a net and into the opponent's court. The object of the game is to play the ball in such a way that the opponent is not able to play a good return.

Tennis is an Olympic sport and is played at all levels of society and at all ages. The sport can be played by anyone who can hold a racquet, including wheelchair users. The modern game of tennis originated in Birmingham, England, in the late 19th century as "lawn tennis". [1] It had close connections both to various field ("lawn") games such as croquet and bowls as well as to the older racquet sport of real tennis. During most of the 19th-century in fact,

Currently you are at - Sports Tennis

So., you will get the Current state of the ScrollSpy for all the sections.

Conclusion

In this chapter, we have seen different types of ScrollSpys in Bootstrap. Hope you understand.

Thanks for reading.

Chapter-29

Affix

Introduction

So far we have seen Scroll Spy in Bootstrap, and now let's see **Affix** in this chapter.

Using Bootstrap code

At first, lets see how to add **Affix** class of Bootstrap to our HTML with the below code,

```
<div data-spy="affix" data-offset-top="60" data-offset-bottom="200">
  <ul class="nav nav-pills">
    <li class="active"><a data-toggle="tab" href="#one">Section One</a></li>
    <li><a data-toggle="tab" href="#two">Section Two</a></li>
    <li><a data-toggle="tab" href="#three">Section Three</a></li>
  </ul>
  <div class="tab-content">
    <div id="one" class="tab-pane fade in active">
      <h1>Welcome to the First section</h1>
    </div>
    <div id="two" class="tab-pane fade">
      <h1>Welcome to the Second section</h1>
    </div>
    <div id="three" class="tab-pane fade">
      <h1>Welcome to the Third section</h1>
    </div>
  </div>
</div>
```

The above will gives the following output in your browser,



This is something like **Tabs** that we have seen before.

Using JavaScript:

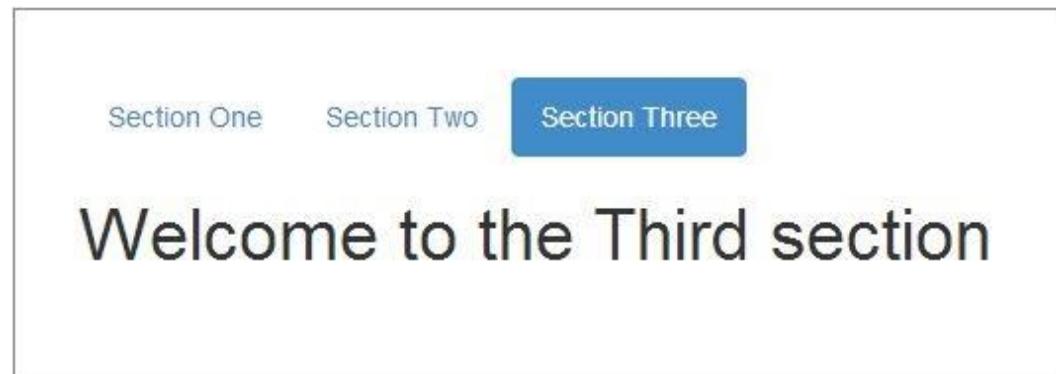
We can also use the Bootstrap's JavaScript to create Affix as below,

```
<script type="text/javascript">
$(document).ready(function () {
    $("#navAffix").affix({
        offset: {
            top: 140, bottom: 200
        }
    });
});
</script>
```

And HTML as,

```
<div id="navAffix">
    <ul class="nav nav-pills">
        <li class="active"><a data-toggle="tab" href="#one">Section One</a></li>
        <li><a data-toggle="tab" href="#two">Section Two</a></li>
        <li><a data-toggle="tab" href="#three">Section Three</a></li>
    </ul>
    <div class="tab-content">
        <div id="one" class="tab-pane fade in active">
            <h1>Welcome to the First section</h1>
        </div>
        <div id="two" class="tab-pane fade">
            <h1>Welcome to the Second section</h1>
        </div>
        <div id="three" class="tab-pane fade">
            <h1>Welcome to the Third section</h1>
        </div>
    </div>
</div>
```

This combination will give the same output as above i.e,



Affix Demonstration:

Now see the clear picture of **Affix** here. Please add below Style sheet to your view,

```
<style>
    ul.affix {
        position: fixed;
        top: 0px;
        left: 50px;
        width: 200px;
    }
    ul.affix-top {
        position: static;
    }
    ul.affix-bottom {
        position: relative;
    }

    /* First level of nav */
    .sidenav {
        margin-top: 50px;
        padding: 10px;
        background-color: #f1f1f1;
        border-radius: 6px;
        border: 1px solid #0094ff;
    }

    /* All levels of nav */
    .sidebar .nav > li > a {
        display: block;
        color: #0094ff;
    }
    .sidebar .nav > li > a:hover,
    .sidebar .nav > li > a:focus {
        text-decoration: none;
        background-color: #ffffff;
    }

    /* Nav: second level */
    .sidebar .nav .nav {
        margin-bottom: 10px;
    }
    .sidebar .nav .nav > li > a {
        padding-top: 3px;
        padding-bottom: 3px;
        padding-left: 25px;
        font-size: 90%;
    }
    #sections {
        background-color: #f1f1f1;
    }
</style>
```

Dont get confused from above Style sheet as we are using them in the below code.

```

<div data-spy="scroll" data-target="#navAffix">
  <div class="container">
    <div class="jumbotron">
      <h1>Bootstrap Affix</h1>
    </div>
    <div class="row">
      <nav id="navAffix" class="sidebar col-md-2">
        <ul class="nav sidenav" data-spy="affix" data-offset-top="200">
          <li class="active">
            <a href="#education"><strong>Education</strong></a>
            <ul class="nav">
              <li><a href="#dotnetfunda">DotNetFunda</a></li>
              <li><a href="#itfunda">ITFunda</a></li>
              <li><a href="#kidsfunda">KidsFunda</a></li>
            </ul>
          </li>
          <li>
            <a href="#sports"><strong>Sports Zone</strong></a>
            <ul class="nav">
              <li><a href="#football">FootBall</a></li>
              <li><a href="#cricket">Cricket</a></li>
              <li><a href="#hockey">Hockey</a></li>
              <li><a href="#tennis">Tennis</a></li>
            </ul>
          </li>
        </ul>
      </nav>
      <div id="sections" class="col-md-10">
        <article id="education">
          <h2><a href="#">Education</a></h2>
          <div
id="dotnetfunda"><h3>DotNetFunda.Com</h3><strong>DotNetFunda</strong> is a popular online
tutorials and guide for latest Microsoft® technologies aimed for beginners and
intermediate level professionals. We help beginners to become intermediate level
professionals and help intermediate level professionals to become an expert. By following
our different sections regularly, we hope you will soon become one of the 'Most Valuable'
professional and start shining in your career.</div>
          <div id="itfunda"><h3>ITFunda.Com</h3><strong>ITFunda</strong> is an
e-commerce website that furnishes a platform to the interested buyer and seller to
interact and transact for the software or software related services. We typically host
study materials, software products / services for buying/selling. This website has been
brought to you by IT Funda Corporation, a parent establishment of a popular Microsoft(R)
technology related knowledge based website www.dotnetfunda.com.</div>
          <div id="kidsfunda"><h3>KidsFunda.Com</h3><strong>KidsFunda</strong>
is an entertainment website for kids with learning fundamentals to improve their skills
and knowledge.</div>
        </article>
        <br />
        <article id="sports">
          <h2><a href="#">Sports</a></h2>
          <div id="football">
            <h3>FootBall</h3>
            <p>
              <strong>Football</strong> refers to a number of sports that
involve, to varying degrees, kicking a ball with the foot to score a goal. The most
popular of these sports worldwide is association football, more commonly known as just

```

"football" or "soccer". Unqualified, the word football applies to whichever form of football is the most popular in the regional context in which the word appears, including association football, as well as American football, Australian rules football, Canadian football, Gaelic football, rugby league, rugby union,[1] and other related games. These variations of football are known as football codes.

```
</p>
</div>
<div id="cricket">
    <h3>Cricket</h3>
    <p>
        <strong>Cricket</strong> is a bat-and-ball game played
between two teams of 11 players each on a field at the centre of which is a rectangular
22-yard long pitch. Each team takes its turn to bat, attempting to score runs, while the
other team fields. Each turn is known as an innings.
    </p>
    <p>
```

The bowler delivers the ball to the batsman who attempts to hit the ball with his bat away from the fielders so he can run to the other end of the pitch (which is counted as one run) without getting run out (the event in which the fielder throws the ball directly onto wickets or to a player who is near to wickets so he can dislodge them from the ground before the batsman or the non-striker has reached the crease). Each batsman (the other is called non-striker) continues batting until he is out. The batting team continues batting until ten batsmen are out or specified number of overs (defined as 6 countable balls bowled) have been bowled, at which point the teams switch roles and the fielding team comes in to bat.

```
</p>
</div>
<div id="hockey">
    <h3>Hockey</h3>
    <p>
        <strong>Hockey</strong> is a family of sports in which two
teams play against each other by trying to maneuver a ball or a puck into the opponent's
goal using a hockey stick. In many areas, one sport (typically field hockey or ice
hockey[1]) is generally referred to simply as hockey.
    </p>
    <p>
```

The first recorded use of the word "hockey" is from the 1773 book Juvenile Sports and Pastimes, to Which Are Prefixed, Memoirs of the Author: Including a New Mode of Infant Education, by Richard Johnson (Pseud. Master Michel Angelo), whose chapter XI was titled "New Improvements on the Game of Hockey".[2] The belief that hockey was mentioned in a 1363 proclamation by King Edward III of England is based on modern translations of the proclamation, which was originally in Latin and explicitly forbade the games "Pilam Manualem, Pedivam, & Bacularem: & ad Canibucam & Gallorum Pugnam". The English historian and biographer John Strype did not use the word "hockey" when he translated the proclamation in 1720.

```
</p>
</div>
<div id="tennis">
    <h3>Tennis</h3>
    <p>
        <strong>Tennis</strong> is a sport people play individually
against a single opponent (singles) or between two teams of two players each (doubles).
Each player uses a racquet that is strung with cord to strike a hollow rubber ball
covered with felt over or around a net and into the opponent's court. The object of the
game is to play the ball in such a way that the opponent is not able to play a good
return.
    </p>
    <p>
```

Tennis is an Olympic sport and is played at all levels of society and at all ages. The sport can be played by anyone who can hold a racquet, including wheelchair users. The modern game of tennis originated in Birmingham, England, in the late 19th century as "lawn tennis".[1] It had close connections both to various field ("lawn") games such as croquet and bowls as well as to the older racquet sport of real tennis. During most of the 19th-century in fact, the term "tennis" referred to real tennis, not lawn tennis: for example, in Disraeli's novel *Sybil* (1845), Lord Eugene De Vere announces that he will "go down to Hampton Court and play tennis".[2]

```
</p>
</div>
</article>
</div>
</div>
</div>
</div>
```

Run this code in your browser and you will see the following output,

The screenshot shows a web page titled "Bootstrap Affix". At the top, there is a header section with the title "Bootstrap Affix". Below the header, there is a navigation bar with two main categories: "Education" and "Sports Zone". Under "Education", there are links to "DotNetFunda.Com", "ITFunda", and "KidsFunda". Under "Sports Zone", there are links to "ITFunda.Com", "FootBall", "Cricket", "Hockey", and "Tennis". The main content area contains three sections: "Education" (with a brief description of DotNetFunda), "ITFunda.Com" (with a brief description of ITFunda), and "KidsFunda.Com" (with a brief description of KidsFunda). The "Education" section is currently active, indicated by a blue background.

Now scroll down and see the change because of Affix,

- Education**
 - DotNetFunda
 - ITFundu
 - KidsFunda

- Sports Zone**
 - FootBall
 - Cricket
 - Hockey
 - Tennis

Education

DotNetFunda.Com

DotNetFunda is a popular online tutorials and guide for latest Microsoft® technologies aimed for beginners and intermediate level professionals. We help beginners to become intermediate level professionals and help intermediate level professionals to become an expert. By following our different sections regularly, we hope you will soon become one of the 'Most Valuable' professional and start shining in your career.

ITFundu.Com

ITFundu is an e-commerce website that furnishes a platform to the interested buyer and seller to interact and transact for the software or software related services. We typically host study materials, software products / services for buying/selling. This website has been brought to you by IT Fundu Corporation, a parent establishment of a popular Microsoft(R) technology related knowledge based website www.dotnetfunda.com.

KidsFunda.Com

KidsFunda is an entertainment website for kids with learning fundamentals to improve their skills and knowledge.

Sports

FootBall

Football refers to a number of sports that involve, to varying degrees, kicking a ball with the foot to score a goal. The most popular of these sports worldwide is association football, more commonly known as just "football" or "soccer". Unqualified, the word football applies to whichever form of football is the most popular in the regional context in which the word appears, including association football, as well as American football, Australian rules football, Canadian football, Gaelic football, rugby league, rugby union,[1] and other related games. These variations of football are known as football codes.

Cricket

And also if you select any content, the **side nav** will be fixed and not scrolled down.

- Education**
 - DotNetFunda
 - ITFundu
 - KidsFunda

- Sports Zone**
 - FootBall
 - Cricket
 - Hockey
 - Tennis

Sports

FootBall

Football refers to a number of sports that involve, to varying degrees, kicking a ball with the foot to score a goal. The most popular of these sports worldwide is association football, more commonly known as just "football" or "soccer". Unqualified, the word football applies to whichever form of football is the most popular in the regional context in which the word appears, including association football, as well as American football, Australian rules football, Canadian football, Gaelic football, rugby league, rugby union,[1] and other related games. These variations of football are known as football codes.

Cricket

Cricket is a bat-and-ball game played between two teams of 11 players each on a field at the centre of which is a rectangular 22-yard long pitch. Each team takes its turn to bat, attempting to score runs, while the other team fields. Each turn is known as an innings. The bowler delivers the ball to the batsman who attempts to hit the ball with his bat away from the fielders so he can run to the other end of the pitch (which is counted as one run) without getting run out (the event in which the fielder throws the ball directly onto wickets or to a player who is near to wickets so he can dislodge them from the ground before the batsman or the non-striker has reached the crease). Each batsman (the other is called non-striker) continues batting until he is out. The batting team continues batting until ten batsmen are out or specified number of overs (defined as 6 countable balls bowled) have been bowled, at which point the teams switch roles and the fielding team comes in to bat.

Hockey

That is the use of **Affix** in Bootstrap.

Events of Affix:

Below list are the Events of Affix in Bootstrap which are used in different scenarios.

Event type	Description
affix.bs.affix	This event fires immediately before the element has been affixed.
affixed.bs.affix	This event is fired after the element has been affixed.
affix-top.bs.affix	This event fires immediately before the element has been affixed-top.
affixed-top.bs.affix	This event is fired after the element has been affixed-top.
affix-bottom.bs.affix	This event fires immediately before the element has been affixed-bottom.
affixed-bottom.bs.affix	This event is fired after the element has been affixed-bottom.

Conclusion

In this chapter, we have seen Affix in Bootstrap. Hope you understand.

Thanks for reading.