

Lesson – 1: Introduction of Bootstrap CSS Framework

What is Twitter Bootstrap?

Bootstrap is a sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript.

History

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

Why Use Bootstrap?

- **Mobile first approach** – Bootstrap 3, framework consists of Mobile first styles throughout the entire library instead them of in separate files.
- **Browser Support** – It is supported by all popular browsers.



- **Easy to get started** – With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
- **Responsive design** – Bootstrap's responsive CSS adjusts to Desktops, Tablets and Mobiles. More about the responsive design is in the chapter Bootstrap Responsive Design.



- Provides a clean and uniform solution for building an interface for developers.
- It contains beautiful and functional built-in components which are easy to customize.
- It also provides web based customization.



- And best of all it is an open source.

What Bootstrap Package Includes?

- **Scaffolding** – Bootstrap provides a basic structure with Grid System, link styles, and background. This is covered in detail in the section **Bootstrap Basic Structure**
- **CSS** – Bootstrap comes with the feature of global CSS settings, fundamental HTML elements styled and enhanced with extensible classes, and an advanced grid system. This is covered in detail in the section **Bootstrap with CSS**.
- **Components** – Bootstrap contains over a dozen reusable components built to provide iconography, dropdowns, navigation, alerts, pop-overs, and much more. This is covered in detail in the section **Layout Components**.
- **JavaScript Plugins** – Bootstrap contains over a dozen custom jQuery plugins. You can easily include them all, or one by one. This is covered in details in the section **Bootstrap Plugins**.
- **Customize** – You can customize Bootstrap's components, LESS variables, and jQuery plugins to get your very own version.

Lesson – 2: Download Bootstrap

It is very easy to setup and start using Bootstrap. This chapter will explain how to download and setup Bootstrap. We will also discuss the Bootstrap file structure, and demonstrate its usage with an example.

Download Bootstrap

You can download the latest version of Bootstrap from <https://getbootstrap.com/>. When you click on this link, you will get to see a screen as below –



Here you can see two buttons –

- **Download Bootstrap** – Clicking this, you can download the precompiled and minified versions of Bootstrap CSS, JavaScript, and fonts. No documentation or original source code files are included.
- **Download Source** – Clicking this, you can get the latest Bootstrap LESS and JavaScript source code directly from GitHub.

File structure

Precompiled Bootstrap

Once the compiled version Bootstrap is downloaded, extract the ZIP file, and you will see the following file/directory structure –

Bootstrap CSS Framework



As you can see, there are compiled CSS and JS (bootstrap.*), as well as compiled and minified CSS and JS (bootstrap.min.*). Fonts from Glyphicons are included, as it is the optional Bootstrap theme.

HTML Template

A basic HTML template using Bootstrap would look like this –

```
<!DOCTYPE html>  
<html>  
  
  <head>  
    <title>Bootstrap 101 Template</title>  
    <meta name = "viewport" content = "width = device-width, initial-scale  
= 1.0">  
  
    <!-- Bootstrap -->  
    <link href = "css/bootstrap.min.css" rel = "stylesheet">  
  
    <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media  
queries -->  
    <!-- WARNING: Respond.js doesn't work if you view the page via file://  
-->  
  
    <!--[if lt IE 9]>  
    <script src =  
"https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>  
    <script src =  
"https://oss.maxcdn.com/libs/respond.js/1.3.0/respond.min.js"></script>  
    <![endif]-->  
  
  </head>  
  
  <body>  
    <h1>Hello, world!</h1>  
  
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->  
    <script src = "https://code.jquery.com/jquery.js"></script>  
  
    <!-- Include all compiled plugins (below), or include individual files  
as needed -->  
    <script src = "js/bootstrap.min.js"></script>  
  
  </body>
```



Bootstrap CSS Framework

```
</html>
```

Here you can see the **jquery.js**, **bootstrap.min.js** and **bootstrap.min.css** files that are included to make a normal HTML file to the Bootstrapped Template. Just make sure to include jQuery library before you include Bootstrap library.

DOTNET Institute

Lesson – 3: Bootstrap Grid System

What is Bootstrap Grid System?

As put by the official documentation of Bootstrap for grid system –

Bootstrap includes a responsive, mobile first fluid grid system that appropriately scales up to 12 columns as the device or viewport size increases. It includes predefined classes for easy layout options, as well as powerful mixins for generating more semantic layouts.

Let us understand the above statement. Bootstrap 3 is mobile first in the sense that the code for Bootstrap now starts by targeting smaller screens like mobile devices, tablets, and then “expands” components and grids for larger screens such as laptops, desktops.

Mobile First Strategy

- **Content**
 - Determine what is most important.
- **Layout**
 - Design to smaller widths first.
 - Base CSS address mobile device first; media queries address for tablet, desktops.
- **Progressive Enhancement**
 - Add elements as screen size increases.

Working of Bootstrap Grid System

Grid systems are used for creating page layouts through a series of rows and columns that house your content. Here's how the Bootstrap grid system works –

- Rows must be placed within a **.container** class for proper alignment and padding.
- Use rows to create horizontal groups of columns.
- Content should be placed within the columns, and only columns may be the immediate children of rows.
- Predefined grid classes like **.row** and **.col-xs-4** are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and the last column via negative margin on **.rows**.
- Grid columns are created by specifying the number of twelve available columns you wish to span. For example, three equal columns would use three **.col-xs-4**.

Media Queries

Media query is a really fancy term for "conditional CSS rule". It simply applies some CSS, based on certain conditions set forth. If those conditions are met, the style is applied.

Media Queries in Bootstrap allow you to move, show and hide content based on the viewport size. Following media queries are used in LESS files to create the key breakpoints in the Bootstrap grid system.

```
/* Extra small devices (phones, less than 768px) */  
/* No media query since this is the default in Bootstrap */  
  
/* Small devices (tablets, 768px and up) */  
@media (min-width: @screen-sm-min) { ... }  
  
/* Medium devices (desktops, 992px and up) */  
@media (min-width: @screen-md-min) { ... }  
  
/* Large devices (large desktops, 1200px and up) */  
@media (min-width: @screen-lg-min) { ... }
```

Occasionally these are expanded to include a **max-width** to limit CSS to a narrower set of devices.

```
@media (max-width: @screen-xs-max) { ... }  
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }  
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }  
@media (min-width: @screen-lg-min) { ... }
```

Media queries have two parts, a device specification and then a size rule. In the above case, the following rule is set –

Let us consider this line –

```
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
```

For all devices no matter what kind with *min-width: @screen-sm-min* if the width of the screen gets smaller than *@screen-sm-max*, then do something.

Grid options

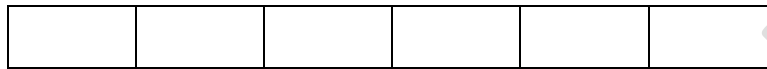
The following table summarizes aspects of how Bootstrap grid system works across multiple devices –

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
# of columns	12	12	12	12
Max column width	Auto	60px	78px	95px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

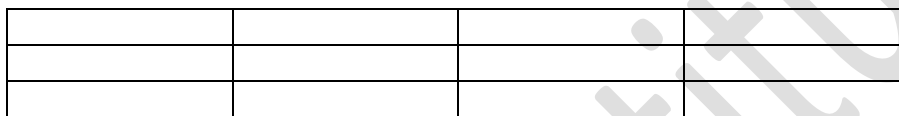
12 Columns



6 Columns



4x3 = 12 Columns



DOTNET Institute

Lesson – 4: Bootstrap CSS & HTML Layout

HTML5 doctype

Bootstrap makes use of certain HTML elements and CSS properties that require the use of the HTML5 doctype. Hence include the below piece of code for HTML5 doctype at the beginning of all your projects using Bootstrap.

```
<!DOCTYPE html>
<html>
  ...
</html>
```

Mobile First

Since Bootstrap 3 has been launched, Bootstrap has become mobile first. It means 'mobile first' styles can be found throughout the entire library instead of them in separate files. You need to add the **viewport meta tag** to the **<head>** element, to ensure proper rendering and touch zooming on mobile devices.

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

- *width* property controls the width of the device. Setting it to *device-width* will make sure that it is rendered across various devices (mobiles, desktops, tablets...) properly.
- *initial-scale = 1.0* ensures that when loaded, your web page will be rendered at a 1:1 scale, and no zooming will be applied out of the box.

Add **user-scalable = no** to the **content** attribute to disable zooming capabilities on mobile devices as shown below. Users are only able to scroll and not zoom with this change, and results in your site feeling a bit more like a native application.

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

Normally *maximum-scale = 1.0* is used along with *user-scalable = no*. As mentioned above **user-scalable = no** may give users an experience more like a native app, hence Bootstrap doesn't recommend using this attribute.

Responsive Images

Bootstrap 3 allows you to make the images responsive by adding a class **.img-responsive** to the **** tag. This class applies **max-width: 100%**; and **height: auto**; to the image so that it scales nicely to the parent element.

```
<img src = "... " class = "img-responsive" alt = "Responsive image">
```

Typography and Links

Bootstrap sets a basic global display (background), typography, and link styles –

- **Basic Global display** – Sets *background-color: #fff;* on the `<body>` element.
- **Typography** – Uses the *@font-family-base*, *@font-size-base*, and *@line-height-base* attributes as the typographic base.
- **Link styles** – Sets the global link color via attribute *@link-color* and apply link underlines only on *:hover*.

If you intend to use LESS code, you may find all these within *scaffolding.less*.

Containers

Use class **.container** to wrap a page's content and easily center the content's as shown below.

```
<div class = "container">  
  ...  
</div>
```

Take a look at the **.container** class in *bootstrap.css* file –

```
.container {  
  padding-right: 15px;  
  padding-left: 15px;  
  margin-right: auto;  
  margin-left: auto;  
}
```

Note that, due to padding and fixed widths, containers are not nestable by default.

Take a look at *bootstrap.css* file –

```
@media (min-width: 768px) {  
  .container {  
    width: 750px;  
  }  
}
```

Lesson – 5: Bootstrap Typography

Bootstrap uses Helvetica Neue, Helvetica, Arial, and sans-serif in its default font stack. Using typography feature of Bootstrap you can create headings, paragraphs, lists and other inline elements. Let see learn each one of these in the following sections.

Headings

All HTML headings (h1 to h6) are styled in Bootstrap. An example is shown below –

```
<h1>I'm Heading1 h1</h1>
<h2>I'm Heading2 h2</h2>
<h3>I'm Heading3 h3</h3>
<h4>I'm Heading4 h4</h4>
<h5>I'm Heading5 h5</h5>
<h6>I'm Heading6 h6</h6>
```

The above code segment with Bootstrap will produce following result –

Inline Subheadings

To add an inline subheading to any of the headings, simply add `<small>` around any of the elements or add `.small` class and you will get smaller text in a lighter color as shown in the example below –

```
<h1>I'm Heading1 h1. <small>I'm secondary Heading1 h1</small></h1>
<h2>I'm Heading2 h2. <small>I'm secondary Heading2 h2</small></h2>
<h3>I'm Heading3 h3. <small>I'm secondary Heading3 h3</small></h3>
<h4>I'm Heading4 h4. <small>I'm secondary Heading4 h4</small></h4>
<h5>I'm Heading5 h5. <small>I'm secondary Heading5 h5</small></h5>
<h6>I'm Heading6 h6. <small>I'm secondary Heading1 h6</small></h6>
```

The above code segment with Bootstrap will produce following result –

Lead Body Copy

To add some emphasis to a paragraph, add class = "lead". This will give you a larger font size, lighter weight, and a taller line height as in the following example –

```
<h2>Lead Example</h2>
<p class = "lead">This is an example paragraph demonstrating
the use of lead body copy. This is an example paragraph
demonstrating the use of lead body copy.This is an example
paragraph demonstrating the use of lead body copy.This is an
example paragraph demonstrating the use of lead body copy.
This is an example paragraph demonstrating the use of lead body copy.</p>
```

Emphasis

HTML's default emphasis tags such as `<small>` sets text at 85% the size of the parent, `` emphasizes a text with heavier font-weight, and `` emphasizes a text in italics.

Bootstrap offers a few classes that can be used to provide emphasis on texts as seen in the following example –

```
<small>This content is within tag</small><br>
<strong>This content is within tag</strong><br>
<em>This content is within tag and is rendered as italics</em><br>

<p class = "text-left">Left aligned text.</p>
<p class = "text-center">Center aligned text.</p>
<p class = "text-right">Right aligned text.</p>
<p class = "text-muted">This content is muted</p>
<p class = "text-primary">This content carries a primary class</p>
<p class = "text-success">This content carries a success class</p>
<p class = "text-info">This content carries a info class</p>
<p class = "text-warning">This content carries a warning class</p>
<p class = "text-danger">This content carries a danger class</p>
```

Abbreviations

The HTML `<abbr>` element provides markup for abbreviations or acronyms, like WWW or HTTP. Bootstrap styles `<abbr>` elements with a light dotted border along the bottom and reveals the full text on hover (as long as you add that text to the `<abbr>` title attribute). To get a slightly smaller font size add `.initialism` to `<abbr>`.

```
<abbr title = "World Wide Web">WWW</abbr><br>
<abbr title = "Real Simple Syndication" class = "initialism">RSS</abbr>
```

Addresses

Using `<address>` tag you can display the contact information on your web page. Since the `<address>` defaults to display: block; you'll need to use

Tags to add line breaks to the enclosed address text.

```
<address>
  <strong>Some Company, Inc.</strong><br>
  007 street<br>
  Some City, State XXXXX<br>
  <abbr title = "Phone">P:</abbr> (123) 456-7890
</address>

<address>
  <strong>Full Name</strong><br>
  <a href = "mailto:#">mailto@somedomain.com</a>
</address>
```

Blockquotes

You can use the default `<blockquote>` around any HTML text. Other options include, adding a `<small>` tag for identifying the source of the quote and right-aligning the blockquote using class `.pull-right`. The following example demonstrates all these features –

```
<blockquote>
  <p>This is a default blockquote example. This is a default
  blockquote example. This is a default blockquote
  example.This is a default blockquote example. This is a
  default blockquote example.</p>
</blockquote>

<blockquote>
  This is a blockquote with a source title.
  <small>Someone famous in <cite title = "Source Title">Source
  Title</cite></small>
</blockquote>

<blockquote class = "pull-right">This is a blockquote aligned to the right.
  <small>Someone famous in <cite title = "Source Title">Source
  Title</cite></small>
</blockquote>
```

Lists

Bootstrap supports ordered lists, unordered lists, and definition lists.

- **Ordered lists** – An ordered list is a list that falls in some sort of sequential order and is prefaced by numbers.
- **Unordered lists** – An unordered list is a list that doesn't have any particular order and is traditionally styled with bullets. If you do not want the bullets to appear, then you can remove the styling by using the class `.list-unstyled`. You can also place all list items on a single line using the class `.list-inline`.
- **Definition lists** – In this type of list, each list item can consist of both the `<dt>` and the `<dd>` elements. `<dt>` stands for *definition term*, and like a dictionary, this is the term (or phrase) that is being defined. Subsequently, the `<dd>` is the definition of the `<dt>`. You can make terms and descriptions in `<dl>` line up side-by-side using class `dl-horizontal`.

The following example demonstrates each of these types –

```
<h4>Example of Ordered List</h4>
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ol>
```



<h4>Example of UnOrdered List</h4>

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
```

<h4>Example of Unstyled List</h4>

```
<ul class = "list-unstyled">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
```

<h4>Example of Inline List</h4>

```
<ul class = "list-inline">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
```

<h4>Example of Definition List</h4>

```
<dl>
  <dt>Description 1</dt>
  <dd>Item 1</dd>
  <dt>Description 2</dt>
  <dd>Item 2</dd>
</dl>
```

<h4>Example of Horizontal Definition List</h4>

```
<dl class = "dl-horizontal">
  <dt>Description 1</dt>
  <dd>Item 1</dd>
  <dt>Description 2</dt>
  <dd>Item 2</dd>
</dl>
```