1. Project Setup

In this section, we will use git to pull down the project files from a remote repository on gitHub to your local computer.

1.1 Download the project files

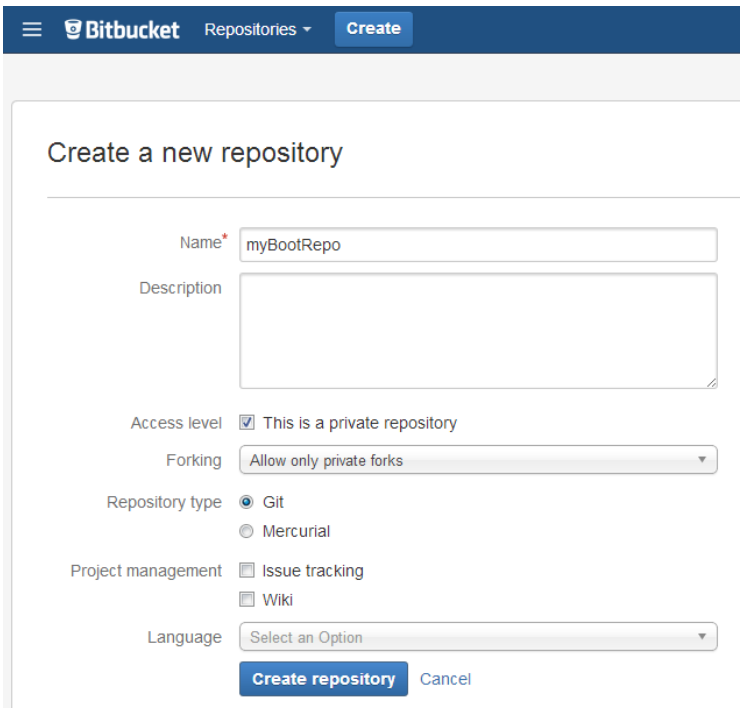
Use the terminal (or GitBash for windows users), and navigate to the directory where the project will be created.

Check out the project by typing the command in bold:

$ **git clone https://github.com/mclarsen/bootstrap.git**
Cloning into 'bootstrap'...
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
Checking connectivity... done

1.2 Create your own repository on BitBucket.org

Login to your account or create one if you have not done so already. Create a new repository by pressing the “create” button on the top menu bar.



Name the repository so we can access it from your local computer. In the figure, the repository is named "myBootRepo". The default settings are fine for this project, but make sure that Git is the repository type.

1.3 Connect to your BitBucket.org Repo

First, we will point Git at the new repository that you just created. From the terminal, type the two commands. Please substitute your actual BitBucket username where *yourUserName* is used. The first command tells Git that, when we update (push) the files, update them on *your* personal repository **not** where you downloaded (cloned) the project from.

```
$ git remote set-url origin https://bitbucket.org/yourUserName/myBootRepo
$ git push origin master
Username for 'https://bitbucket.org': yourUserName
Password for 'https://yourUserName@bitbucket.org':
Counting objects: 19, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (18/18), done.
Writing objects: 100% (19/19), 663.88 KiB | 0 bytes/s, done.
Total 19 (delta 0), reused 0 (delta 0)
To https://bitbucket.org/mclarceny/myBootRepo
 * [new branch]      master -> master
```

Finally, refresh the BitBucket page to verify that your files have been uploaded.

2 Get to know Bootstrap

Bootstrap is a Javascript and CSS framework that was developed by Twitter for use on their own website. However, Twitter decided to make the world a better place and make it freely available. The main benefit of using Bootstrap is to create clean, robust, full featured websites quickly. It has many built in styles and Javascript functions that will make your life just a little easier, but still giving you full control if needed.

2.1 Responsive Design

One of the problems that plagued web developers is getting the page to look good on all screen sizes. The site could be viewed from a mobile device, a 12" monitor at your grandmother's house, or 30" HD widescreen monitor. In the past, one approach to solve this was to code the site twice. One site would be built for desktop/laptop users and another for mobile devices. Responsive design is an attempt to save us web developers from doing just that. No one wants to do twice the work. Bootstrap has a responsive design component that just works. Add one line of code to the page, and you're done.

```
<link rel="stylesheet" type=      "text/css" href="css/bootstrap-responsive.min.css">
```

To see this in action, open up the index.html page from any browser. Now, reduce the width of the window slowly and watch the navigation on the top of the page. When the page cannot display the menu any more, bootstrap creates a button that is a drop down menu. Also, notice how the content of the page begins to stack vertically.

2.2 The Grid and CSS

Bootstrap provides a grid system, using CSS style classes, for content that divides the page into predefined block sizes which a developer can utilize to rapidly layout content on a page. The feature allows for the developer to focus on the content of the page rather than playing around with borders, margins, and content sizes to get intended page layout. Furthermore, there is a robust built-in page navigation component that makes a developer's task more manageable. Finally, there are CSS rules defined for every html element that provide a simple clean appearance to the html elements. A developer can easily override these rules to create a customized look to the site.

Let us take a look at the grid system. Open the file gridDemo.html in a browser, and then open the same file with a text editor so see the structure. You will notice that everything is structured using ** ** tags. The grid allows for 12 "units" for the width of the page. The line:

** span1 **

This creates an area the width of the default page size. The style class, span12, says that we want this section to span the entire width. You can use everything from span1-span12 in any combination that equals 12. In order for Bootstrap to function properly, these ** ** tags need to be surrounded with a ** ** and . We won't get into why here.

3. Content Layout

3.1 Adding Spans

Open the index.html page inside your favorite text editor. The embedded comments are there to help you figure out what various parts of the page are doing, but we will focus on the main Content.

Locate:

```
<div class="container">
    <div class="hero-unit">
        <h1><span style="color:green">CIS</span> 199</h1>
        <br>
        <p> &#8704;x Loves(x,IceCream)</p>
        //INSERT HERE
    </div>
</div>
```

Using the gridDemo.html as an example, create a row with two spans of your choosing that sum up to twelve. This is creating a two column layout for the page. Then, customize the content of the page. Think Anna.

3.2 Navigation

Now, we will add another page and make the navigation functional.

```
<ul class="nav">
                <li class="active"><a href="index.html"><i class="icon-home icon-white" style="padding-
                left:10px"></i>Home</a></li>
                 <li><a href="gridDemo.html">News</a></li>
</ul>
```

The navigation bar is controlled with an unordered list, and each list item becomes a navigation option. The class “active” make the list item appear “selected”. Change the list item “News” to Images. Then, change the href to “images.html”.

Save the page to update the changes to index.html, and then save as images.html to get a new page with the same layout as index. Now we have a new page to work with. The first task is to change the active class, in the navigation, to the current page, images.html.

3.3 Image Carousel and JavaScript

Finally, we will take advantage of one of the many Javascript components provided by Bootstrap. First, we need some images to display. Go get three images off the web and put them in your img directory (or use the ones provided). Here is the predefined html off the Bootstrap website:

```
<div id="myCarousel" class="carousel slide">
      <ol class="carousel-indicators">
            <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
            <li data-target="#myCarousel" data-slide-to="1"></li>
            <li data-target="#myCarousel" data-slide-to="2"></li>
      </ol>
      <!-- Carousel items -->
      <div class="carousel-inner">
            <div class="active item">…</div>
            <div class="item">…</div>
            <div class="item">…</div>
      </div>
<!-- Carousel nav -->
<a class="carousel-control left" href="#myCarousel" data-slide="prev">&lsaquo;</a>
<a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a>
</div>
```

To add you own images in, add an image tag inside the “item” div.

<div class="item" **> **

To add a caption for the image, insert the following code immediately after the image tag inside the “item” :

```
<div class=”carousel-caption”>
        <h4>This the caption title</h4>
        <p>This is the description</p>
</div>
```

Finally, we will manipulate the image carousel by using Javascript (JQuery). If you have not used JQuery before, you will like it. It is basically a Javascript library loaded inside a script tag:

<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>

JQuery allows you grab elements off the page and change their properties is a shorter more powerful form. For example, remember **document.getElementId(“theId”).innerHTML()**? This same statement is JQuery is: **$(“#theId”).html().** Inside the $(‘theSelector’), is the selector. The basic format for selectors is as follows:

To select by Id : $(‘#theId’)

To select by class : $(‘.className’)

Here is a partial list of the attributes of the carousel that we can use:

.carousel('cycle')

Cycles through the carousel items from left to right.

.carousel('pause')

Stops the carousel from cycling through items.

.carousel(number)

Cycles the carousel to a particular frame (0 based, similar to an array).

.carousel('prev')

Cycles to the previous item.

.carousel('next')

Cycles to the next item.

Example of calling Next : $('.carousel').carousel(‘next’)

Normally, we would used these attributes to accomplish inside a javascript function, but, for demonstration purposes, we will used them directly in the web console. So, open the web console in Firefox or Chrome. Enter: $('.carousel').carousel(next’) . Now try one of the other functions.

4 Finishing

Now that we have finished the exercise, let us save the changes we made to the project files since our first push. First, we added some files during the course of the learning module. We have to tell Git to add them to the list of files that it tracks.

git add –A (git, please add all the new files to your list)

git commit –a -m “This is the commit message” (git, please commit all the changes)

git push (git, please transfer all the changes upstream to my repo)

That is it. You are done. During this module, you have been exposed to a number of technologies. You have been introduced to Twitter Bootstrap, Git, and a little Jquery. Hopefully, this will give you the ability to go out and learn more about technologies that are built specifically to help us web developers create better web sites, faster. There will always be a learning curve, but the rewards are well worth it.