Bootstrap and jQuery (Day 17)

Lab and Homework

First - create a basic html page. It should include the jQuery library. Add some simple content and display it to make sure it works.

Next, add the tags required by Bootstrap. The beginning of the file should look like this:

```
!doctype html>
<html lang="en">
<head>
    <title>Bootstrap Playing</title>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">

    <link rel="stylesheet" href="bootstrap-4.3.1/css/bootstrap.css" />
</head>
<body>
  <p>Hello Bootstrap</p>
```

Note that the section above includes the link for the bootstrap.css file. Download bootstrap and unzip it - in the code above (and all subsequent references) the bootstrap files are included locally. The rest of the file looks like this:

```


  <script src="jquery/jquery-3.4.1.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
      integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
      crossorigin="anonymous">
  </script>
  <script src="bootstrap-4.3.1/js/bootstrap.min.js"></script>
  <script src="lodash.js"></script>
  <script>

  $( document ).ready(function() {
    });

  </script>
</body>
</html>
```

In this case the file `popper.js` is being retrieved from a CDN, you can choose to reference it that way or download it locally also.

Start out simple.
Create a `<div>` section at the top of the page. Put a welcome message in it.
Now, surround that `<div>` with another one. If you refresh the page nothing should happen.
Now, create a Jumbotron. Apply classes to the `<div>` sections such that the look like this:

```
<div class="jumbotron">
  <div class="container">
  your message
  </div>
</div>
```

Refresh the page to see the change.

Look at the list of components on the Bootstrap documentation: https://getbootstrap.com/docs/4.3/components/alerts/

Pick out a few of the components and code some simple sections.

- Alert
- Button
- Cards
- Collapse
- Dropdowns

Using div sections, create several message areas that have different background colors. Reference the `bg-` series of classes.

Use `<p>` tags and the `text-` classes to create several areas of colored text.

Combine the `bg-` and `text-` classes to make some custom alert-like sections.

Next, create a basic table using `<div>` sections:

```
<div class="container">
  <div class="row">
    <div class="col">
      My column data
    </div>
    <div class="col">
      My column data
    </div>
    <div class="col">
      My column data
    </div>
  </div>
</div>
```

One additional feature that may be helpful to see the borders is to use border styles. You can either create simple custom ones:

```
<style>
  .border-blue
  {
    border: 1px solid blue;
  }
  .border-red
  {
    border: 1px solid red;
  }
</style>
```

and apply those alternating in each column, or use the built-in Bootstrap border classes:

```
<!-- using custom -->
<div class="col border-blue">

<!-- using bootstrap -->
<div class="col border border-alert">
```

Resize the browser and notice how the column react.

Create several more rows, each with a different number of columns. Continue to resize the browser and notice the behavior.

Next, change the columns to use absolute column sizing using `col-3`, `col-4`, etc. Make sure that sum and limit is 12.

Alter the sum of the columns so that it is less than 12. What happens?

Alter the sum of the columns so that it is more than 12. What happens?

Next, add adaptive sizing. Change the `col-` classes to `col-md-` on various rows. Resize the browser and observe the behavior.

Using the `.offset` classes, create a row that has space to the left and all the column content shown to the right.

Using the `justify-content-` classes, create a grid that forms an "X" on the screen. That is the top row has content at the ends (grid 1 and 12), the second row has one empty space on either end (conent in grid 2 and 11), etc.

Create an imput form using the `form-group` and `form-control` classes. For reference look at the forms section of the Bootstrap documentation. Also create a button on the form and make it a primary button.

Create a button on the page for changing colors. Code several iterations of the button and alter the color of the some of the sections you created earlier by adding or removing classes. For example, if you created a section with `class="alert alert-success"` change that to have an `alert-danger` class.

MVC and MVVM

Some of the best references for MVC are on the Microsoft site. These discuss the "full-stack" version of MVC, meaning a web front end and compiled back end. However they do have very useful descriptions of the main components. Click on the following links and carefully review the articles:

https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-2.2

https://dotnet.microsoft.com/apps/aspnet/mvc

Not to be outdone, here is the spring MVC site.

https://spring.io/guides/gs/serving-web-content/

And finally without going to Angular (b/c there will be several lessons on angular coming up) here is the ember js discussion on "client side" MVC: https://guides.emberjs.com/v1.10.0/concepts/what-is-ember-js/

For MVVM the best reference for web frameworks is the Knockout site:

https://knockoutjs.com/