Gothberg, L.
Assignment: How-To
Class: CS 290, Kothari, M.

Bootstrap: An Open-Source HTML, CSS, and JS Framework

What is Bootstrap?

Bootstrap is a library of pre-defined CSS (Cascading Style Sheets) and JS (JavaScript) definitions that makes building a website significantly less obtuse. CSS is the syntax that allows for formatting of individual elements on a web page via the use of HTML (HyperText Markup Language, the "code" used to build websites) tags (which define what a particular does, such as an anchor link to another page), classes (used multiple times on multiple different in a given page), and unique identifiers (typically only used for a single specific element of a page). JS is a dynamic programming language that is typically used to add interactive elements to a web page (i.e. drop-down selection menus, submit buttons on a form, or the ability to update various parts of a given webpage).

While not a proper API (Application Program Interface), Bootstrap functions in much the same way. By linking to the source code, it allows the end user to call on the pre-defined classes and features included in the Bootstrap Framework. The primary difference is that a proper API is usually used to send data to another computer (typically called a "server") which will do something with that data, then (usually) send some other data back to the user. Bootstrap is a "read-only" collection of functions.

There are some other aspects to Bootstrap, such as the ability to use Less and Sass to extend the pre-defined CSS, but I won't go into them here for the sake of brevity.

How to use Bootstrap:

The first step is including the Bootstrap Library into your new (or existing) website project. There are multiple was of doing this, with various benefits and drawbacks to each. For the purposes of this document, I'll cover what I think is the simplest option, using the CDN (Content Delivery Network) method[i]. Because using this method involves linking your website project to a pre-compiled version of the Bootstrap CSS and JS libraries hosted on other machines, it's very easy to do. Simply include the following snippets inside the <head></head> tags of your HTML code:

```
<!-- Bootstrap -->
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
```

This provides access to the CSS portions of Bootstrap, which do most of the heavy lifting when it comes to formatting your webpage. You'll probably be extending or adding to the the things defined in this library as well, so having a link to your own CSS file is wise. Just be sure that the link to you CSS is placed after the link above.

Similarly, you'll need the following link right before the closing </body> tag in your HTML:

```
<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<!-- Latest compiled and minified Bootstrap JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mCGD8wGNIcPD7Txa"
crossorigin="anonymous"></script>
```

As with the CSS, you'll likely be adding your own bits of JS, so having a link to that is a good idea. In this case, however, it doesn't matter if you place the link before the above code or after.

Using Bootstrap:

This is where things begin to get interesting. It does require digging through the documentation to find the right thing, but doing so is part of any API or Library. Once you've discovered the correct CSS class, it's a simple matter of using them as you would any custom CSS class or ID definition you wrote yourself. By far the most useful portion of the Bootstrap Framework is what they define as the "Grid System". It allows you to define a container of a specific width that will determine how the items inside that container are placed on the page, and is built to be dynamic so it works on devices of various sizes.

Example:

The following will create a block element that has the class “container” and “col-md-4”, which is a wrapper for whatever you choose that will be about a third the width of the browser window (the Grid System is defined as being 12 columns wide). The “md” in “col-md-4” tells the browser that for windows approximately 970 pixels wide, this element is shown as normal. The “4” tells the browser that the container should be 4 “columns” wide (out of the possible 12), or 1/3 the size of the window.

```
<div class=”container col-md-4”>
    <!--INSERT CODE HERE -->
</div>
```

Other variations include “col-xs-*”, “col-sm-*”, “col-lg-*”, and “col-xl-*” (* being the number of columns you wish to choose, anything between 1 and 12, inclusive). The only difference is how wide the columns will appear. Using “sm” will set a minimum column width of about 62 pixels, where using “lg” will set a minimum column width of about 97 pixels. Which one you use will depend on what kind of device the webpage is intended to be viewed on. If possible, all columns will expand equally above the minimum width to provide equal spacing on a larger device.

Example:

The following is a way to define a dropdown menu activated by clicking on a button. Doing all of this in CSS and JS would take a full page or so of code if you were to type it out manually, but it is entirely possible. This is just another example of the benefits of using a pre-defined library.

```
<div class="dropdown">
    <button class="btn btn-default dropdown-toggle" type="button" id="dropdownMenu1" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="true"> Dropdown <span
class="caret"></span>
    </button>
    <ul class="dropdown-menu" aria-labelledby="dropdownMenu1">
            <li><a href="#">Action</a></li>
            <li><a href="#">Another action</a></li>
            <li><a href="#">Something else here</a></li>
            <li role="separator" class="divider"></li>
            <li><a href="#">Separated link</a></li>
    </ul>
</div>
```

Demonstrations of the above snippets can be found here. (NOTE: This is the framework for a currency conversion website that is currently under construction, so it is in no way complete.) Other examples of the features included in Bootstrap like the “Jumbotron” image header can be viewed here.

[i] Full instructions for the other installation methods can be found on the Bootstrap Website.