**CSS Code Review**

* Which if any of the philosophies we discussed this week they might be using (BEM, Atomic, OOCSS, etc)?

Breaking a code into multiple files is one key advantage to using a preprocessor, and forms the basis of any architecture. With Sass, there’s no problem in breaking the code language into the smaller units and organizing them into multiple files and folders as in this case. As we can see here, the Sass has conferred new power on the CSS @import rule, allowing you to combine Sass and CSS files during compilation so they can be sent to the browser as one single file. In this code we see two files, @import 'variables' and @import 'node\_modules/bootstrap/scss/bootstrap' integrated into variables. CSS @import directive allows the programmers to reference one CSS file from another. Importing is handled by the browser and requires additional HTTP requests.

* Are the folders they use well named?

When creating reusable design components, it’s very for the programmers common to scope (as it were) the component’s sub-elements inside the component’s class name. In this case, the elements such as  [button](http://getbootstrap.com/docs/4.0/components/buttons/), [navbar](http://getbootstrap.com/docs/4.0/components/navbar/), or [progress bar](http://getbootstrap.com/docs/4.0/components/progress/) seem to have the sub elements names that maintained the parent elements’ root name.

* Check out the variables partial. Imagine trying to customize Bootstrap without that settings file! Consider the advantages Sass gives to both the organization and customization of Bootstrap.

But what’s really important is how to use them efficiently. The vast majority of CSS and Sass organization systems are based on some concept of user interface “components”or discrete pieces that can be put together to form a complete project. Components can be any size or shape, but they should focus on doing one task independently, and in a reusable way. A button, a drop-down, a calendar are all examples of components that can be reused at different places across a project. Thinking ab

* If you were a developer working on this code and you were asked to work on on something specific...like headline styles...how hard would it be to locate the code and modify it? Would you worry about your changes having unintended consequences after reviewing the class names and architecture of that portion of css?

I think the main thing that would worry me is whether the modification of the file will be limited to the specific part of the element I want to change or it would modify the entire file.

* Do you like how it is organized? Would you change anything?

I like the way it is organized, but I will make sure to name the files and selectors in such way that I would avoid to modify unintentionally.