Alyssa Nielsen
ANNHX3 / 14211627
Exploration 1 Documentation
CS 4830

# Exploration #1 – Sass

For my first exploration assignment, I wanted to explore Sass. I enjoy web design, and want to try to make a decent personal site, so I decided I would try to learn some more advanced ways of doing CSS. I have never used or even seen Sass used, so I used a tutorial on TeamTreehouse.com called Sass Basics since I have a subscription and the tutorial seemed fairly in depth, with quizzes and mini-tests at the end of each lesson. I have submitted my source code, which includes all of the HTML, CSS and SCSS files I created to learn the fundamentals of Sass, and I have hosted the HTML file from the project that I created using the tutorial challenges, with some modifications I made for fun, on my AWS instance at ec2-54-152-146-70.compute-1.amazonaws.com. It is a very basic example, but it did introduce me to a ton of very useful tools for website CSS design. Each of the sections of the tutorial I practiced on the main.scss and a style.scss file, with respective HTML and CSS files that correspond to them. The style.scss file is the one I hosted, as it was the fake demo "blog" I slowly built throughout the tutorial challenges, and the main.scss and corresponding files I used to follow along as I practiced writing the examples and running them myself throughout the videos. I commented out code we didn't use from one section to the next as I built up so I could see my previous work.

To begin, I went to Sass-lang.com and went to the install page, where I found out I need to go to rubyinstaller.org to install Ruby first. After installing Ruby, I followed the Sass site instructions and used the Ruby command prompt to install Sass. I also downloaded the Scout app from scout-app.io as well, it is a type of GUI for Sass, but it really didn't add much to the experience for me so I ended up just using the command line. I set up a basic HTML file, and a Sass file, and through the command line I set up a watch on my folder containing my Sass file, so that any updates to my file would be output by Sass into a CSS file in the same folder. I split screened to work on all three simultaneously, which seemed to be the best way to watch the changes in real time on save. I began learning about the nesting feature of Sass. I like to take a quick look at documentation ( http://sass-lang.com/documentation/file.SASS_REFERENCE.html ) when working with something unfamiliar, and Sass's documentation is pretty decent. I looked up most of the things I learned in each section as I went in Sass's documentation, as well as some other sites I will include as I used them.

The nesting feature was the first one I learned, and it is nice for organizing your CSS in a way that makes more logical sense and makes it less redundant. I used the '&' and '>', both of which are neat features – I looked further into these features at https://css-tricks.com/the-sass-ampersand/. This portion of the tutorial it was suggested I download Modernizr at https://modernizr.com/ which I did, and it checks the browser for what all HTML, CSS, and Javascript features the browser can run and track and auto-fill in the tags for you. Pretty cool.

Alyssa Nielsen
ANNHX3 / 14211627
Exploration 1 Documentation
CS 4830

Next, I moved on to the section where I learned how to make my code DRY using variables, mixins and extending selectors in Sass. This was a pretty simple and intuitive concept, but a very useful one when working with CSS. It works the same as a variable does in many other programming languages, as a placeholder for a value, so that you only need to change the value once but can reuse it in many places that wouldn't need to change. Another way to promote code reusability with Sass is using mixins. Mixins allow you to make a function essentially that can be called with @mixins and the function name within your Sass / CSS file and allows you to add parameters as well. This is a great feature to have at your disposal. I looked at https://www.sitepoint.com/sass-basics-the-mixin-directive/ for further documentation regarding mixins. I looked into, but didn't practice much, the math and include options for mixins, too, in the tutorial and documentation. The @extend property I also found really useful, as it allows you to extend a selector's set of CSS properties from one to another. Also great for reusability. Another topic in this section was functions, and making them with Sass is simple and again very similar to many other programming languages.

The next section I delved into was about color palettes and what you can do with color, again not super mind-blowing or exciting, but useful nonetheless when working on designs. There are nice features like complementary, lighten, and darken you can use, as well as tons of others. I learned about importing SCSS files into a file, using @import, which is great for when you use many separate pages for your CSS like you are supposed to in order to separate out different areas of design. This also allows you to use Sass libraries, like Compass and Bourbon. I looked up Bourbon on http://bourbon.io/, and discovered I needed to install it via my Ruby command line again. It was a very simple couple of commands and it was installed. It includes many mixins for Sass pre-made for you, which is nice. I like anything that makes my life easier.

I thought the tutorial was good, and gave me a good idea of what Sass is about. I would feel comfortable jumping into a real project and trying to use this, and plan to skim the documentation more to see what else I can do with Sass, as I am sure this barely scratched the surface.