**Sunday April 28th, 2019**

Yesterday I finished the Bootstrap exercises in FreeCodeCamp. Today I’m going to try to get as far as I can in the jQuery exercises. While jQuery seems to be falling out of favor in the world of web development, it’s still a pretty widely used library. Being familiar with it should prove to be useful, particularly in regards to maintaining older projects.

11:26 — The first thing I’m immediately being reminded of when completing these exercises is that everything in jQuery begins with the $ symbol.

11:29 — To select a class using jQuery we use a dot or period “.” just like CSS.

11:40 — To select an id using jQuery we use a #, just like CSS. My first impressions upon being exposed to jQuery again is that adding effects and animations is extremely simple with this library. I can definitely see how in a certain sense it’s like Bootstrap. Both jQuery and Bootstrap are designed to help make the design process easier. Unfortunately, both do have the potential to stunt our growth in native CSS and vanilla Javascript.

11:48 — While we can add classes with jQuery’s addClass() function, we can also remove them using jQuery’s removeClass() function.

11:50 — jQuery has a pretty cool .css() function that allows us to change the CSS of an element!

18:50 — We can add HTML tags and text within an element using the jQuery .html () function.

18:54 — jQuery’s appendTo () function allows us to select HTML elements and append them to another element.

18:56 — If we would like to clone an element, it is possible to do so using jQuery’s clone () function.

18:57 — While I often chained multiple methods together in order to solve FreeCodeCamp’s JavaScript algorithms, the documentation states that chaining is also frequently used in jQuery.

18:59 — I remember having to learn about parent, child, and sibling elements while learning CSS. It turns out jQuery has some convenient functions to target the parents or functions of elements such as parent () and children ().

19:10 — I’ve just finished the FreeCodeCamp exercises for jQuery. They were a good refresher of jQuery. While I learned jQuery with an Andrew Chalkley course a few years ago on teamtreehouse, the first time around I found a lot of it confusing. To be fair, however, my understanding of JavaScript was pretty limited at that time.

Now that I have a decent grasp of JavaScript, understanding jQuery enough to start implementing it in my projects or understanding enough to look at others jQuery code seems a lot more manageable.

17:56 — Just got back from a quick break walking around the city. Let’s start SASS! I briefly did a few teamtreehouse videos to get familiar with SASS a couple years ago, but it’s something I don’t remember much about.

I know it’s used to add features beyond what are available in native CSS and I vaguely remember mix-ins being discussed the last time I learned SASS but that’s about it. I’m sure it’ll come back quickly once I get into the FreeCodeCamp exercises though.

20:02 — It seems there are two slightly different syntaxes available for Sass (Syntactically Awesome Stylesheets). It doesn’t seem to whether someone learns SCSS or straight-up Sass first as transitioning from one to another should be relatively easy once you get the hang of one of them.

20:05 — It turns out just like JavaScript, variables are available in Sass. The only difference is that instead of using var or let or const, Sass variables start with the $ symbol followed by the variable name and a colon. Using variables in Sass can make our websites more maintainable by allowing us to change many elements in our CSS through simply adjusting a single variable in Sass.

20:09 — While this isn’t directly related to Sass, I have to say that the way FreeCodeCamp delivers you all of its lesson in bite-sized chunks before asking you to put everything together in each section’s final projects is truly brilliant.

I once tried to learn to code when I was 13. At that time I was just messing around in Xcode and trying to piece things together from books and random online hello world and flashlight tutorials. Everything felt so insanely complex because it all just came on me at once.

If only FreeCodeCamp had been around 10 years ago…

20:12 — Oh wow, this is cool already. Sass allows you to next your code instead of having to manually type out all of the parent selectors. For example, you could nest you <li> rules inside of your <ul> instead of having to manually select ul li as is typically the case in CSS.

20:18 — It appears that Mixins are the Sass equivalent of JavaScript functions. By utilizing Mixins we can greatly improve our ability to follow the DRY programming principle (“Don’t Repeat Yourself”).

20:25 — I’m going to head to the bathroom and maybe get a quick snack before getting in another pomodoro session or two in before bed.

21:19 — That walk and smoothie break ended up being longer than expected haha. Anyway, I’m back now. Similar to JavaScript, Sass also uses if, else if, and else statements.

21:14 — Apparently using @for we can also add styles in a loop. From what I can tell this seems to be almost the same as for loops in JavaScript.

21:34 — I just constructed a Sass for loop that increments the size of each paragraph by 10px. While I probably won’t be using @for in this fashion on enterprise websites, so far it’s just another feature that makes me feel excited to learn more about Sass!

21:41 — FreeCodeCamp also provides a basic introduction on each loops in Sass. I personally found their explanations of the syntax a little cryptic so I’d recommend watching this video if you’re also struggling: <https://www.youtube.com/watch?v=GHUNtbW22V8>

21:58 — I finally got a rough idea of how to work with each loops in Sass. If someone told me that I needed to incorporate a Sass each loop into my project I would be confident in doing so. If I had to do so in a live coding interview, however, I may be a bit less confident. That’ll come with incorporating these Sass features into my front-end library projects in the coming weeks and months! :D

22:00 — It’s getting late now. I’m going to push this journal to GitHub and call it a day.

Total time spent coding today: 2 hours 14 minutes

Total time spent coding thus far in April 2019: 40 hours 25 minutes