Turku University of Applied Sciences

Software Development Skills Front-End, Online course

Louis Lautz, 2003022

LEARNING DIARY

Introduction to workflow and sass

27.1.2023

I learned how to install sass and integrate it into the standard node.js workflow. I already worked a little bit with node, but it was good to get a quick refresher.

I learned about the html syntax autocomplete feature in VS Code 🤯  
Especially the “! + tab” feature to generate HTML boilerplate is amazing! In a previous course I had to always copy paste the boilerplate from the internet.

Homepage and Core Sass/CSS

27.1.2023

The first new thing I learned in this task was the &-operator in sass. It can be nested into a css selector and refers to any element that fits the main selector, but further specifies the selection to the elements with the linked class. I guess this can help with reducing the number of individual selectors needed. In my last course about responsive web design, I often had multiple individual selectors with only a couple attributes inside. I am sure this &-operator can come in handy in those situations.

The next new thing I learned about is the lighten() function. It takes in a color and a factor to lighten the color by. This sounds very useful, because it means that we don’t have to come up with a new color. In fact, it also makes the code more maintainable, because it basically creates a new shade of what even color we input as the first argument. If I were completely change my color scheme, the lighten function would adapt the created shade accordingly, which sounds very useful.

In addition to those things, this lesson goes into more detail about the nesting behavior of sass. Previously I would do something like this: *nav > ul > li > a {}* to select an element in a nested structure. With sass, I can mimic the nested structure of the HTML with a nested structure of sass. This feature makes the selectors a lot clearer, because your eyes don’t have to read every full selector, but rather follow the nested structure downwards, just like they would when looking at HTML.

After the Mixin section around the 21 minute mark, I was curious what else could be done with mixins, as the author of the video only used it for one transition. My curiosity led me to [this](https://www.youtube.com/watch?v=3wXqwX4nu8I) video. It appears that mixins are another way of separating shared attributes to keep the “dry” (don’t repeat yourself). An interesting point a commentor brought up was, that you could just put all of the attributes into a class and apply the class to the elements that would receive the mixins instead. At the moment I don’t see a reason not to do that, so I am intrigued whether there is a truly unique use case for the mixins further down the line. But so far I like the aesthetics of mixins.

The final interesting part that I would not have figured out by myself is the use of the :after pseudo element. I already know how to use pseudo elements but using it in this context is very smart. He basically used it as a shader level on top of a background image, instead of using a separate element for that. Very clever.

The only thing left now is to gather some images and add them to the repository. But I will do that next time.

Rotating Menu Button

30.1.2023

The third lesson taught me how to select DOM elements with JavaScript and add/remove classes.

Then I learned how to chain transform properties to create more complex animations. I did use transform properties before, but I didn’t know that they can take multiple arguments.

Menu Overlay & Responsiveness

30.1.2023

The first new thing I learned in this lesson is the flex-flow propery. Intuitively, I would have used the flex-direction property to set the flex orientation to vertical .In the video, the author doesn’t really explain the flex-flow property, but a quick google search revealed, that it a higher level property that sets the flex-direction and the the flex-wrap property at once. Just like margin sets margin-top, margin-right… etc.

Another new and very interesting use case for sass in this lesson is the part with for loop. In the video, the author wants to give each nav item its own delay to create a staggered transition effect. Traditionally, we would have to select each nav item individually with :nth-child() and set a specific delay. With sass however, we can use a for loop to select the nav items one by one and write general code that will be applied during each iteration to set a new delay for the items.

Lastly, I learned how we can separate media queries with sass, to be able to style mixins rather than media queries.

Page With CSS Grid

30.1.2023

Lesson #5 taught me how use sass functions. They were used in this example to choose an appropriate font color, based on the color of the background. The functions work very intuitively.

Another small thing that I was surprised about, was the part where he moved the import statement for the \_mobile.sass file to the end of the main file.  
I always thought, that the order of CSS statements doesn’t matter, but since we make all of these media queries, which overwrite properties, I guess it makes sense to put them at the bottom of the page, so that they actually overwrite properties.

Responsive Portfolio Website

30.1.2023

In this lesson, the first new thing I learned was the @extend syntax. Although it seems a lot like the standard CSS syntax of using two classes on an element. In this example, we could have used the class btn and btn-dark and it would have worked the same way. So this sass function does not seem that helpful to me. I find it confusing to use a class selector, when that class doesn’t actually exist.

On the contact page I took the liberty to change the hover effect on the contact items. The author changed the padding on hover, which made the button shrink and slightly changes the position of the other buttons, giving the page a “flowy” feel. I noticed though, that when the cursor is very close to the edge of the div, the hover effect is getting switched on and off constantly, because the button moved away from underneath the button. So, instead I decided to achieve a similar effect, by using the transform: scale(); function. It doesn’t achieve the same “flowly” effect, because it doesn’t change the position of the other elements, but it improves the shrinking behavior.

Website Deployment

31.1.2023

The final step of the course exercises was the deployment. I have never done that before, so I was surprised how easy it is to deploy static websites on GitHub pages.