Lappeenrannan teknillinen yliopisto
School of Business and Management

Software Development Skills

Leevi Saarinen, 000399177

LEARNING DIARY, FRONT-END MODULE

LEARNING DIARY

3.6.2021

Today I started doing the course. I familiarized myself with the learning goals and objectives of the course. I took a quick glance at the module task list to get a good overview of what I will be working on during this course. I learned what I need to do to complete the course and get a grade.

The first step to complete the course was to get familiarized with Git and choose a code-editor. I am already quite good at using Git as I have used it many times before in software projects. I decided to use vs-code because I have used it in projects before too.

I learned that SASS is a CSS pre compiler. It is a scripting language like CSS. I have heard about SASS before, and I would like to learn how it can be used in frontend developing. It seems to be similar to using CSS, being used in styling.

In node SASS can be used by installing the SASS node package and setting a script in “package.json” that compiles the scss into css. Easiest way in development is to make the script watch changes in the application and run the compiler every time a change occurs.

Responsive design makes the website arrange differently according to the screen size of the device it is used. For example, when the page is used in mobile device the elements often should be positioned vertically and on a bigger screen the elements can be horizontal.

I set up the environment like in the video: VS Code with prettier, Bracket pair colorizer and Live Server extensions.

4.6.2021

During this day I learned a lot more about SASS and CSS. Things I learned included some size units (rem, vh, vw), mixins, variables nesting.

rem is the font size declared in the html tag times the number in front of it. For example, 5rem means 5 times the size of html font size.

If a class is nested inside another class or other selector, it is only applied to the selectors that are its parents. For example, if icons class is nested inside a main selector in CSS, then only the elements with icons class which are inside a main tag are affected by the CSS selector.

Mixins can be used to define re-usable styles. They work similarly to functions in many programming languages. They can include parameters and conditions. Variables in SASS work as in many other programming languages: they are way to reuse a value.
In SASS using & in front of a selector declares a parent selector. It is a special selector that is "used in nested selectors to refer to the outer selector. It makes it possible to re-use the outer selector in more complex ways, like adding a pseudo-class or adding a selector before the parent." (SASS documentation).

27.6.2021

Today I started the third part of the course. I learned how to use JavaScript to manipulate the DOM of a html page. This can be done by selecting the correct DOM objects with querySelector-method. To watch the changes in button an EventListener can be used. addEventListener -method requires the event and function as its parameters. A 'click' parameter is a good event for a button to trigger the event.
By manipulating the DOM different classes can be replaced with other classes in DOM. With JavaScript and SASS, I was able to change the html elements that I am using in my Web App. With transform CSS property animations can be done to a web app.

30.6.2021

I learned how to make a web app responsive using media queries. In SASS media queries can be made using mixins, which are sort of functions in SASS. I learned about mixins in previous parts of the course but now I also learned how to make media queries with them.

means that the web app changes some of its styling according to the size of the screen it is used. It is very important to make web apps responsive in order to get good usability and user experiences for the app.
I also learned how to use functions and for loop within SASS. When I looked at the CSS file that my SASS files have compiled, the code looks so much complicated and complex. The functions and mixins in SASS are so much easier and simpler to write comparing to doing the same things in plain CSS.

2.7.2021

I learned how html components can be positioned by using CSS grid layout. It is a good way to specify where each different html component should be placed within a page. I also learned how to make a page with a lot of html elements responsive to mobile users. This can be done by again using media queries and making the page position every element vertically so that the elements are on top of each other.

As my .scss files have started to grow very large and they contain a lot of nesting, I have started to appreciate the bracket pair colorizer -extension that helps a lot by highlighting different CSS selectors in the file.

6.7.2021

I learned more about the grid layout. It is used by giving a container a property “display: grid”. In grids the number of items in columns need to be specified. Guide to Grid layout: https://css-tricks.com/snippets/css/complete-guide-grid/.

I learned about the flexbox layout, which is another way to arrange html items within a container. It is responsive and mobile friendly because the flex box items arrange themselves based on the size of the page. Flexbox is used by giving a container a property “display: flex”. Flexbox items can also be arranged in an order so that html does not need to be changed. However, I would think, that this could cause some issues as then html items in the DOM are not in the sa me order as they are shown on the web page.

By default, the sibling flexbox items seem to go on a horizontal row and being same sized despite their contents. However, the items seem to be easy to arrange differently with properties related to the flexbox layout such as “flex-direction” and “align-items.” Guide to flexbox: https://css-tricks.com/snippets/css/a-guide-to-flexbox/#flexbox-properties.

19.7.2021

I learned how to deploy a web app to web using GitHub pages. It was really simple as I had already made a remote repository to GitHub for the portfolio project. I only had to install the right node package and add correct home page to “package.json”. Also, I had to write a deploy script. Everything did not go as in developmental environment and a few adjustions had to be made. For example image files with extension “.JPG” did not seem to work when deployed on GitHub Pages so they had to be changed into “.jpg”.