Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Tiina Kauvosaari, 0550391

LEARNING DIARY, Software Development Skills: Front-end MODULE

**LEARNING DIARY**

28.10.2022  
  
Today I read all the general materials in moodle to start the course. Course seems to be very interesting, and I want to learn a lot!

I already had Visual Studio Code and Git – account. I had to install same extensions and also Node. I had some problems, but I think everything is ok now. I made new repositor called SDS -kurssi in github. I watch first video very carefully and did everything same way. It wasn’t easy, and I don’t understand everything just yes, but I made it. Html and Css is little bit familiar to me from pass, still I felt that everything was new for me. I wanted to do this course in windows environment.

29.10.2022

Today I worked with video number two. There was so much to do today. First I made index.html page ready; I put links (without real links yet). Then I made account to Font awesome – page to get nice icons. I included link from the font awesome page in to index.html. I used that page to have Twitter, Instagram, LinkedIn and GitHub icons. It was really easy to use that; Here is example; <I class=”fab fa-twitter fa-2x”></i>

fa-2x means that icon will be double size.

Secondly, I continued with sass – file. I learned a lot. Background – image, background- color, z -order ( how “deep” is you element), text size (vh and rem), font-family, marginals, heading, links (hover, color changes when you touch with mouse). I also learned what &after means.

I create another sass – file to save all the variables and mixin. Variables starts with $ and mixin starts with @mixin. Variables and mixin helps a lot, because I could only add variable name to example to background-color instead of writing colors every time. Mixin is quite nice also. I think I will learn more of that in next videos.

5.11.2022

Today I worked with video number tree. I created menu - button, rotate it and I add some javascript. I have realized that I have run sass always when I start working and in between working. I type console “npm run sass” to start with it.

First I created own scss – file to Menu – button. It was called “\_menu.scss”. Link to that file have to add to top of the main.scss-file. Importing goes “@import ‘menu’;   
Menu – button is only tree lines in right corner. I defined where it is in the page and the lines, which are in the menu – button. I used easeOut – mixin. Menu includes links (Home, about me, My work and how to reach me) and I have created those in index.html.

The idea was, that when you click menu – button, it rotates and transform to “X”. To make it rotate, I had to use javascript. It was my first time to write script. It was interesting, and I have to learn it more. Javascript was in one file which I created last time. Link to that file is in the end of index.html -file.

7.11.2022

Video number four. I continued with Menu. It was interesting to use kind of for – loop to add navigation links to screen. I also add color to first link (Home) and put hover color to links when you go on top of link (change color to yellow). I defined text-size, color, padding etc to links also.

Transform3d is interesting. You can add something in your website from somewhere, example for top of the screen.

I created @mixin mediaSm, Md, Lg and Xl to “\_config.sass. Then I created a new file “\_media.scss” where I defined what does mediaSm and Md means for website. When screen is small (Md), I change profile picture, change text size, how branding (where picture is) and links comes to screen.

8.11.2022

Video number five. I started working on text colors. I used @function to do it. There is differens between @mixin and @function. Mixin output lines to sass code and it affect to css styles. Function returns a value, and you can use that value to mixin or another function.

Function set-text-color returns color which depends on lightness. In main.scss – file I add the function to color: set-text-color($primary-color);

After that I changed menu – icon (three lines) and link – colors to same way. they all use function and color depend on that function.

then I created about.html – page. First, I just copied all the materials from index.html to about.html. then I changed title and in navigation, I moved nav-item “current” from Home to About me. So, link “About Me” is now in different color.

I added different content to About Me – page. I removed background picture and add new picture of me. I added content using grid – element. It was nice!

With this page I used grid- templates – column. It was easy to modify page with it. Especially mobile – view, where I put content in different order easily with grid.

11.11.2022

Video number six. Page “My work” and “How to contact me”. With My Work – page, I made layout with grid.

*grid – template – column: repeat(3, 1fr)*

fr means fractional unit. Total width is divided by three and every block gets 1/3 of full width.

*grid – template – column: 250 px repeat(3, 1fr)*

would mean, that first there is 250px width area and the rest is divided to 3 areas.

It was very easy to define how you can see pictures with different device, small and big ones. For example I added to \_mobile.scss – page this:

@include mediaXL{ .projects{ grid – template – columns: repeat(4, 1fr); }}

If you name div “project”

<div class=”projects”>

Then you can add styles in css / scss – page by only typing

.projects{

styles… }

I style buttons.

I created Contact Me – page, contact.html. I used grid – style, this time areas.

In html -page, every div has names (bioimage, bio, job1, job2, job3). With grid, I placed them in order;

*grid – template -areas:*

‘bioimage bio bio’  
‘job job2 job3’

21.11

Before I tested video seven material, I created my own page. I watched video number seven, and I saw, that it is not difficult to publish webpage by using github.

Now, my own project is ready, it is time to finnish and publish the page.

I decided to publish my project in Github. So, first I installed gh-pages (npm i gh-pages).

Then I added package.json – file information about the page;

“homepage”: “https//tiinakauvosaari.github.io/SDS-kurssi”

Then I had to add script to the same file:

“deploy”: “gh-page -d dist”

Then I type to terminal;

npm run deploy

It is then published!

I have to remember, if I make any changes, I have to git add/commit/push and deploy.