Lappeenrannan teknillinen yliopisto

School of Business and Management

Sofware Development Skills

Juha Mursula, 0621738

LEARNING DIARY, FRONT-END MODULE

LEARNING DIARY

29.9.2020 Introduction to Workflow and SASS

First, I installed VS Code without any problems. I already have a GitHub account and Git installed on my computer, so I did not have to install those. However, I have not used GitHub for two years and I had forgotten my password, so I had to set new password. I also noticed that there is a new application, GitHub Desktop, which looks very handy.

I watched the first video and completed the same tasks presented in the video. I installed the 3 add-ons of VS Code that were mentioned in the video. Especially Live Server seemed very practical because it automatically changes the outfit of the web page after you have saved your source code files. I also installed Node.js. The installing program asked to also install some add-ons, but I did not know if they were obligatory to install (nothing was said about add-ons in the video). So, I decided to install them. Finally, I installed Node-Sass from the command line, so I have now the programming environment in use needed in this course.

I also coded the simple code files presented in the video (index.html and main.scss). I learned that VS code is very handy to use because it can guess commands that you are writing and it also completes the code, so you do not necessarily have to remember the syntax perfectly. I also realized the meaning of Sass: it helps the programmer to create style sheets more easily by offering features that does not exist in CSS, e.g. nesting and variables. Node-Sass, in turn, converts the SCSS (Sass) files into ordinary CSS files. Of course, I learned a little bit of syntax too.

30.9.2020 Homepage and Core SASS/CSS

I first watched the whole video. Then I started to watch it again from the beginning and performed the coding tasks presented in the video. It took several hours to do all the tasks. I am not very familiar with HTML or CSS. Several years ago, I took a web programming course, but I have forgotten almost everything. I remembered the basic HTML tags like “header”, “body”, “main” or “li”. In the video the teacher did not explain very well what the different HTML tags are meant for, so I had to go several times to W3Schools web pages to find out what the HTML tags really do. This was very instructive. Now I know for example that “nav” tag is used for making a block of navigation links.

However, following the instructions presented in the video was quite straightforward when coding the HTML file. In addition, VS code helps a lot in coding by completing the code, so the programmer does not usually have to remember the exact syntax. I also noticed that it required 70 rows of HTML code to create the structure of a quite simple web page. That was more than I thought beforehand.

Then about SCSS files. I learned that the syntax remembers many “real” program languages, e.g. all the code will be written inside curly brackets and you must add semicolon after statements. Setting a value for a variable or an attribute is different, however. I also learned that variables and mixins (which are simple functions or macros) can be used in SCSS files. I realized that SCSS (or CSS) will be used for creating the appearance for different HTML tags specified in HTML file. Without CSS the bare HTML page looks very coarse.

While I did the tasks presented in the video, I also often tested other values of the attributes to see how they change the appearance of the web page. Testing was quite easy because VS code shows the alternatives for the value of the attribute. (Later I learned that all alternatives are necessarily not suitable.)

I also looked the CSS file that was generated by SASS. I noticed that all SCSS files were combined into one CSS file and the values of the variables and the mixins were added to the places where they were used in SCSS files. Also, all comments were wiped out.

I had to code the link to the “Fontawesome” web pages in a different way than presented in the video. The video is 2 years old, so it is natural that it is partially out of date.

1.10.2020 Rotating Menu Button

Again, I first watched the video from the beginning to the end and after that I started to do the programming work. The main subject of the third video was JavaScript programming. I have a long experience of programming, but I have not programmed with JavaScript before. Although this is not a JavaScript course, I expect to learn something about the language because it is so essential in web programming. I want to understand when and how JavaScript is used in web programming. In the video, the lecturer mentioned a word “DOM”. I knew beforehand that it means “Document Object Model” and it has something to do with web programming. So, I read a short article about DOM and learned that DOM represents a HTML document as a tree of nodes and provides API that allows to manipulate HTML documents.

In the video it was shown how you can pick up HTML elements into JavaScript code. I learned that it can be done using “querySelector” method with a class name defined in HTML document as a parameter (obviously there are other methods to do that too). I also learned one example how to manipulate HTML elements using JavaScript. Otherwise, the JavaScript coding shown in the video looked like basic programming for me.

Another interesting part in the video was the “metamorphosis” of the menu button. The code itself was quite easy to understand but it was a little surprise for me that the transformation must be done by hand. I expected that there would have existed a built-in function to do the work.

6.10.2020 Menu Overlay and Responsiveness

This time there was a lot of stuff concerning the creation of the appearance of the menu page in the video. However, the things presented in the video were simple and understandable, and it was easy to follow the instructions. Maybe I have also learned something about HTML and SCSS (CSS), which makes the programming more fluent than earlier.

Still, I learned many different and useful SCSS commands. Now I know for example how to place elements in the web page and know many properties that elements can have. I also familiarized myself with a quite imposing “animations” (transitions) that can be done with “translate3d” command. I also learned how to create loops in SCSS (I must say that the syntax is quite complex).

In the video there was also taught how to set the layout of the web page for different devices. In practice, you must only change the appearance of the web page for small laptops, laptops, and smart phones. (Later I learned that sometimes it is useful to code a separate layout for large screens also.) Of course, the need for the changes depends on the layout of the page. You can probably (at least sometimes) design web pages where you do not have to program different appearances for different devices. (Later I learned that e.g. “Flexbox” is designed just for that purpose.)

12.10. Page with CSS Grid

At the beginning of the fifth video it was told that in addition to mixins, you can also create functions. I learned that if you need to return something, you cannot use mixins but you have to use functions. You can call functions in a normal way as in ordinary program languages. I also learned how to change text color so that the text remains readable if the background color has been changed.

I also received a short repetition of how to program a HTML page (although this time we did not create the page from scratch but copied another page for the basis of the new page). Again, it was very nice to notice how handy programming tool VS code is. You do not have to remember all that quite difficult HTML syntax, because VS code completes the command automatically. I also learned some new things about HTML pages, for example “footer” and “p” tags.

CSS grid was the main issue in the video. Now I know how I can divide different HTML elements into rows and columns. The use of CSS grid looks quite simple for the programmer. Of course, the properties of different elements were put up in the video but that was mostly the same as done in previous videos, so I got a little repetition of SCSS coding. An important and useful hint presented in the video was the “sticky footer”: a feature where the footer stays in the bottom of the web page even if the HTML page is too “short” to fill the whole screen.

13.10. Work and Contact Pages

Again, two new HTML pages were created in the video and I increased my understanding a little bit more about how the HTML elements should be placed with respect to each other. I also realized that it is practical to give a class name to HTML elements, so that it is easy to refer to it elsewhere. Of course, you could in many cases use “id” attribute as well, but it is unique, so you can not give same “id” attribute to different elements.

CSS grid was used again in Work/Project page. This time the syntax was little easier, because all elements were the same size and you did not have to use “grid-template-areas” property at all but you only had to tell how many columns (images in the same row) you want. I also learned how to make fancy effects like change the brightness of the image while hovering.

An important note I made was that button styles belonging to images mentioned above were coded outside the code concerning images in the SCSS file. This solution makes it possible to use the same button styles also elsewhere! A necessary thing to know is also that you can create a class in SCSS file (that does not exist in any HTML page) and then extend that class in a class or classes that are created in a HTML page. This looks like a handy way to set same properties to different classes.

The contact page was coded by using “Flexbox” which means that the display is “flexible”, and the elements automatically wrap with respect to the screen size. So, there is no need to do extra programming work to guarantee the responsiveness. Naturally, this kind of technique can only be used with relatively simple web pages.

14.10., 30.10. Website Deployment

The most important thing I learned from this video was that you can deploy your own web page to GitHub (if it does not include back-end software which means that the application runs only in the web browser). I also learned that it is quite easy to register a domain and buy web hosting services. I must seriously consider that when I start to create my own web programming portfolio.

I had some difficulties when deploying the project: most of the images in the deployed web site were not visible at all. Only the background image (turtle) and shrunken pictures were shown properly. So, I had to reduce the size of other pictures as well, so that their file size became smaller. The downside of this operation was that the images in the slide show and parallax page look quite fuzzy now.

This course was very helpful. Now I have a basic understanding about HTML and CSS and I also can use some tools that make the programming of web pages easier. The course was a good starting point for more sophisticated front-end programming.