Course Assignment

Javascript 1

Emil Strøm | Front-End Development | 03.03.2019

Brief

This document includes my report on the course assignment as well as a reflection over the methods and approach I used to complete the assignment, my decisions, technical report, possible flaws and/ or improvements that I might've missed, not had time to do, or wish I had done differently or more efficiently.

The task

The task was to use the attached resources to complete the following assignment on various pages. Create functions that fetches an API, and convert it to JSON format. Display the API contents on the page, and create a search functionality. Create specifics on the API content, and create a contact form and validate it with a function.

Planning stage

In this section of the document I will reflect, detail and discuss my thoughts and decision making, as well as the process behind the planning stage of the website.

I had a quick look at the assignment text, and had to take some time to read-up and talk to my teachers about how to complete some of the elements as I felt like I didn't understand everything completely. After getting clarification from my teachers, I went ahead and planned the code and how I could best utilize what I knew to complete the task, as well as researching other options or measures to complete the assignment.

Development stage

In this section of the report I'll be covering my efficiency, tools, strategies and other elements that made the site into what it is.

I started with the index page, and I immediately ran into trouble as I didn't quite understand how to fetch the API as I hadn't understood the necessary elements to complete the task. I got some advice and some reading material from MJ to complete the first task, and after a little guidance I quickly managed to complete the first part of the index page. After completing the fetch part, I created a function that included a for loop, to iterate through the array of items in the converted .JSON file and created a function that would implement necessary HTML elements to all of the 99 cards that had to be added to the page. I then created an if statement to check if the cards had images or not, if they did not it would have a placeholder attached to it. See line 16-18 in the script.js.

Further I had trouble completing the search functionality so I decided to move on the second part of the assignment: card-specific page.

On the card-specific page, I also had some trouble in the beginning with understanding how to attach the id to the fetch URL, but after some quick googling I managed to figure it out. I then created a function that again checked to see if the ID had an image attached to it or not (see line 18-20 on specific.js). After completing that I decided to use variables to declare where the new information should be displayed (see line 21-22 in specific.js) I then used the result.card. To fetch the rest of the information and display it accordingly.

After completing the card-specific part of the assignment, I moved on to the form as I thought this would be the easiest task. I quickly remembered that Regular Expressions were not the easiest to deal with, however I managed to create patterns that checked for input for the name fields, validates the phone number and email-address. The last assignment I used Regular Expressions to validate, most domains was excluded due to me not paying attention to the entire pattern, this time any domain should work.

I then decided to add a visual error to the input fields as well as displaying the included error message in the HTML. The reason I decided to add the visual errors was because I thought that it will always be better for a user to have a visual representation of where they have gone wrong, to avoid confusion. I'll talk more about this later in the report.

Finally, I proceeded with the about page, which I personally found to be the easiest task at hand after completing all the other parts of the assignment. I decided to create a function that would first edit the attribute of "moreInfoContent" class style so the "Display: none;" would become "Display: block;" So that the "tooltip" would be displayed, I then created a second attribute editor to change the "onclick" to my second function which has the exact opposite of the first function. Essentially the second function closes the box, and changes the "onclick" back to its original state. I personally found this to be the fastest and easiest solution, at least for me.

General Reflection

In this section I will have a general reflection over my final product, what I could've done differently and what I could've done better.

Generally, I could've narrowed the code down quite a lot, at least on some of the pages. More specifically I think that the contact.js and script.js could've been a lot neater and less code could've been used to complete the task. However, I wanted to display extra knowledge and "skill". I tried to add some elements or functions that differed from the set assignment, to display that I know how to do what's required and then some. However, I'm afraid that while doing so, I also unintentionally made the code harder to read and longer than necessary. From what I remember in the lessons

“Less code is better”. At the same time, usability comes first. And I honestly, think that my decisions improves the usability of the page, especially the parts where I added better visual cues for the user to understand what’s gone wrong and where.

One specific part of the assignment I think I could’ve done better at was the search function in the script.js. I’m not happy with the end-result due to function always returning as “false” due to the for loop checking my if statement against all 99 cards. I tried to add “fix” to it, but I couldn’t quite figure out how to bypass the for loop iterating over all 99 cards. I believe there’s a way to fetch only the specific card, and I tried to google that, unfortunately the fetch() has a lot of variables and options and it simply got too complicated for me.

I also believe that I could’ve used psuedo-code here, to remind myself and anyone else looking at the code what the code is for. I tried to do this in a way by using comments, but again, psuedo-code would’ve been simpler to understand than a block of text.

Over-all, I’m pleased with the end-result, but I definitely think it can be done in a more efficient way, especially if we were allowed to use jQuery for the opening, closing, hiding, showing and deleting of elements.