Jonathan Mazurkiewicz

3/8/2023

Class: COP 3813

Assignment: Project 3

Contents:

In the .zip folder, there is:

- Content file (index.html)
- Presentation file / Bootstrap Template (mycss.css)
- Interactive Functionality File (myjs.js)
- One screenshot of the “app” in a web browser (p3_screenshot)
- This report

Explanation:

This project taught me a lot about using Javascript. Initially, I thought it was going to be an extremely easy assignment. Change a couple of things, copy and paste a bit, and then everything would work. However, this turned out not to be the case.

First, I started by finding a suitable Bootstrap template (comments giving credit to the author are listed in the HTML file). After a bit of digging, I found one that was minimalistic and clean in its presentation. I removed several pieces of the old code in the HTML, such as the checkboxes, etc., changed the names and ids of the elements more suited to my purpose, and added in a couple more input boxes to be able to input three numbers. After playing around with the formatting and having a reasonably good looking HTML file, I went off to begin the Javascript.

This is where I spent most of my time on this project. I spent several hours rereading and trying to understand the expected format of event listeners, how to encapsulate the

function inside of another event listener to wait for the DOM to load, and the proper syntax associated with it. To be honest, even after all of my efforts, I am not comfortable in my own ability to recreate another, more complex, event listener. While its general approach is very straightforward (determine the type of event, call a function, access the required variables, and then perform the changes), there are subtle nuances that were deceptively challenging for me to execute. It did not help at all that Brackets is sending me in many different directions based upon the errors that it is producing in the Javascript code. It is unable to recognize "document", it does not know what the keyword "let" is, causing me to search about what could be causing these errors. I have found that the code works perfectly, and similar solutions have been executed by others. I was left to believe that the problem is that Brackets, even though up to date, has an older version of Javascript based upon ECMAScript 5 instead of ECMAScript 6. While this doesn't seem to be logical, these types of errors point in this direction and it made it extremely challenging for me to debug my code.

The next difficulty for me to traverse was finding out how to incorporate the median function into the code. While "Math" contains a min and a max function, and range and mean are very straightforward, median requires the list of elements to be sorted. I did a little bit of research and found out various ways to sort a list of elements. Most were more complicated than I needed them to be. I ended up choosing to implement an in-line function that essentially compares two elements, and based upon the result, sorts the elements in ascending order.

One of the largest problems with my code is that it is not generalizable. There would be many coding tweaks needed to be made in order to allow this code to work effectively for

multiple sizes of inputs (i.e. what to do if there were an even number of inputs for the median, etc.). Amongst possible improvements, here are the things I have come up with.

- Validating the data in case a user only filled out two or less of the inputs, or to only accept positive numbers as inputs, possibly within a certain range.
- Generalizing the code to accept a variable number of inputs
- Asking the user with a checkbox which functions they actually needed calculated.
- Finding a way to display the information graphically for ease of understanding.
- Changing the interface to provide some type of animation or smoothing of transitions to appear more visually appealing.

All in all, the thing I learned most about this assignment is how finicky Javascript code and editors can be. Although conceptually straightforward, it presented challenges that I had not anticipated. Understanding how to wrap a function, or how to properly go from an algorithmic approach to writing the code is more difficult than I had expected it to be. I feel much better now that I was able to struggle through the process than I did prior, but there is still a lot more work to go.