

Splits Calculator Documentation

By Isabelle Bernal

I decided to approach the splits calculator by first coding it in a language where I feel confident which was Java and then taking that foundation to help me build my code in Javascript. This was very helpful because I was able to write and test my code very quickly. If you compare my Java and Javascript code you can definitely tell I added more to my Javascript code as I was thinking of more and more scenarios that could possibly break my code, I added a little more to the math portion in regards to when the user inputs pace or total time, and I had to add a couple more lines of code to help with the formatting of the output. However, I needed that foundation from my Java code to help me be as efficient as possible with coding in Javascript.

From coding the math portion of the splits calculator in Java to later translating it into javascript to put on the webpage, I have now realized that Javascript and Java may have some more differences than I thought. For instance, javascript only has three data types: const, var, and let, which I found can make it difficult to have an int data type for one specific number. Another major difference is the way you print in Java is way different in javascript and required me to go through a lot of trial and error to figure out how to do it. Once I was able to figure out how to print on the webpage in javascript it was really just an extra step in the beginning of my code and changing the syntax. The last major difference that stood out to me was the user input because not only did I have to change the syntax but since there are only three data types in Javascript I had to make sure to change any string from the user input into an actual number. Using `parseInt()` was very useful throughout my program due to this.

Although there were some major differences between Java to Javascript there were still a couple of similarities that made my Java code very useful. For instance, Javascript and Java have the same syntax when it comes to for loops and if statements so I was able to have a good foundation going into my javascript code. Another similarity is the math operators and as long as all the user inputs were converted into numbers correctly and not left as strings, I was able to move over the majority of the math portion from my Java code into my Javascript code. Converting the user inputs into numbers was very important because if the user inputs are left as strings it can mess up your entire code since it will just put the numbers side by side instead of executing the right math operation.

In my program, I was originally going to make only one text box for each user input but I found it difficult in javascript to grab the times as one string and then try to convert it into a number since the string would be “mm:ss” format. To add on, there is not one singular number in that string there are two numbers and two colons so I decided to make a text box for each number inputted. This made converting from the string value to a number much easier and made calculating easier as well. Something else I wanted to point out as well is that I originally was not going to have a refresh button but I decided to add one that will refresh the page entirely because I realized I did not account for if the user wants to start over after clicking the button. I feel that the refresh button is also convenient as well if the user decides they want to start over with the values they inputted but don't want to go through each text box to change it.