WEB FRONTEND DEVELOPMENT II

WDD 330

Ponder: Team Activity

Overview

This activity will review basic HTML and Javascript tasks. Make sure that you have completed the reading from week 1 so that you can contribute to your team.

Instructions

Complete the following assignment as a team. Designate one team member as the "main driver" and collaborate on their copy of the code. Everyone on the team should be actively engaged in writing the code and contributing to the solution. Once the solution is working, make sure that everyone on the team gets a copy of the code. Each week let someone else be the "main driver" of the coding.

Core Requirements

01 Create a new html file. Add the code necessary to make it a valid document

1. Add a text input and a button. Put an empty div with an ID below those.
2. Add a script tag, write a function that will read the contents of the input, and write them to the div.
3. Call the function when the button is pressed.

02 Modify your HTML file:

1. Write a new function that expects a number as an argument. It should take that number and sum all the numbers up to the number provided. (ie n=5…so it would do 1+2+3+4+5 = 15) It should return that value.

2. When the button is pressed it should read the number from the input. Make sure it is a valid number, call the summing function you just wrote, and output the result to the div.

03 Create an adding machine

1. Add an additional input to your HTML file.

2. Add a + button.

3. Write a function that will take the numbers from each input, add them together, and output the result to a `div`

Stretch Goals

It will be very helpful if you have completed the reading on Ch4: Functions before you do this.

01 Function declaration

Part of your reading this week is about Javascript Functions. In Javascript there are several ways to define a function: with a function declaration, a function expression, or an arrow function. Add some more features (subtract, multiply, etc) to your calculator. Make sure to use each of the three methods at least once in your function definitions. Discuss with your team the differences and if there is one you prefer.

02 Writing 'good' functions

In good program design functions should do one thing and do it well. They should be kept simple. Another good programming practice is to keep your code **DRY** (Don't Repeat Yourself). How did you do with your functions? Does each do one thing only? Is there any repeated code? How could you fix this? If you have repeated lines of code in your functions the most common solution is to pull those repeated lines out and create a new function with them. Do this with your code.

03 Callbacks

In Javascript functions can be passed as the argument into another function. When this is done the function being passed in is called a **callback**.

Discuss with your team how callbacks might be used in programming a calculator to help keep your functions simple and your code **DRY**. If you have time try to implement one of your ideas.

Instructors Solution

As a part of this team activity, you are expected to look over a solution from the instructor, to compare your approach to that one. One of the questions on the I-Learn submission will ask you to provide insights from this comparison.

Please DO NOT open the solution until you have worked through this activity as a team for the one hour period. At the end of the hour, if you are still struggling with some of the core requirements, you are welcome to view the instructor's solution and use it to help you complete your own code. Even if you use the instructor's code to help you, you are welcome to report that you finished the core requirements, if you code them up yourself.

After working with your team for the one hour activity, click here for the instructor's solution.

Submission

When you have finished this activity, please fill out the assessment in I-Learn. You are welcome to complete any additional parts of this activity by yourself or with others after your meeting before submitting the assessment.