Web Scripting 3

Assignment: 03 – React Calculator

Course Value: 20%

Due Date: To be finished before 1pm on Day 05 of the Web Scripting 3 course

Assignment Description:

Create a calculator using React

Teamwork:

Although not a requirement, I highly encourage all of you to try and work in teams to try and complete this assignment. Working in teams will allow you to pool your knowledge with your fellow classmates and to help each other solve some coding challenges.

Instructions:

1. Create a fully functioning calculator using the React library
2. When your team has completed this assignment, let your instructor know and your instructor will meet with your group during class and give you a grade and some feedback
    a. No need to submit anything. Your instructor will mark your code by running it on your computer

Levels of Difficulty

- This assignment has 3 levels of specifications
    - Base
    - Bonus 01 – Memory Keys
    - Bonus 02 – Decimal and Sign Keys
    - Bonus 03 – Percent and Square Root Keys
- Each level of difficulty achieved will receive a higher grade
- Even if you only complete the base calculator specifications, you will still receive a grade of 17 / 20
    - The above mark of 17 / 20 assumes you have met all of the requirements of the base calculator
- More details on how you will be graded can be found under the “Marking Criteria” section

Calculator Button Data Files

- Included in the assignment files and in the day 03 lesson files are 4 x calculator button data files. These are JavaScript files with a variable that stores an array of calculator button objects.
- Feel free to use any of these data files to create your calculator buttons dynamically with a loop
- Make sure you select the correct file for the version of the assignment you are building
    - Example:
        - If you are building the Bonus 01 version of this assignment, then use the “calculator-bonus-01-button-data.js” file
- Using any of the calculator button data files is optional

General Application Specifications (Applicable to all difficulty levels)

- Create a working calculator where your users can enter numbers (via mouse clicks) and then perform basic calculations on those numbers
- The exact working of your calculator is up to you and your team
    - Search around the internet to see how other online calculator's work
    - Try your operating system's calculator to see how that calculator operates
    - If you have a physical calculator, give that a try to see how a real calculator operates
    - This assignment does not specify a certain way that your calculator should work. This is up to you. Just build your calculator in a way that you think would be easy for your users to operate
- Your calculator should have a clean and easy to use user interface. The interface should be styled with CSS
    - The exact design of your calculator is up to you

Base Specifications

- Create a calculator with the following functionality
- Keys
    - AC – An "All Clear" key that when clicked on clears all numbers and pending calculations
    - C – A "Clear" key that clears the currently displayed number
    - 0 – 9 – keys that allow your user to enter numeric values into the calculator
    - Add, Subtract, Divide and Multiply keys that allow your user to perform calculations on numbers entered into the calculator
    - An "Equals" key that allows your user to perform the calculation
- Display
    - Displays the numbers entered into the calculator and the result of any calculation

Bonus 01 Specifications – Memory Keys

- All the specification of the Base calculator
- Memory Keys
    - Memory Store – Stores the currently displayed number into memory
    - Memory Recall – Recalls the number that is stored in memory
    - Memory Clear – Clears the number stored in memory
    - Memory Plus – Adds the currently displayed number to the number stored in memory
        - Example:
            - If you have the number 5 stored in memory and the number 15 displayed and you press the Memory Plus key, then the new number stored in memory will be 20
    - Memory Minus – Subtracts the currently displayed number from the number stored in memory
        - Example:
            - If you have the number 20 stored in memory and the number 15 displayed and you press the Memory Minus key, then the new number stored in memory will be 5

Bonus 02 Specifications – Decimal and Sign Keys

- All the specifications of the Base and the Bonus 01 calculator
- Decimal Key
    - Allows your user to enter decimal numbers into the calculator
- Sign Key
    - Allows your user to switch a number from a positive number to a negative number and a negative number to a positive number

Bonus 03 Specifications – Percent and Square Root Keys

- All the specifications of the Base, Bonus 01 and Bonus 02 calculator
- Percent Key
    - Allows your user to convert a displayed number to a percent value
- Square Root Key
    - Allows your user to get the square root of a displayed number

Starter Files

- In the day 03 folder you will find some starter files for each version of this project.
- You are free to use these starter files or create your application from scratch. Either option is fine
- If you choose to use the starter files know that the Create React App installation has already been performed
    - You will need to run "npm install" to download the required node modules
- Inside each starter folder you will find a "calculator-button-data.js" file
    - This file contains JSON data that you can optionally use to generate your calculator buttons
    - Using this file is optional

Screenshots and Movie Demos

- The provided screenshots and movie demos are for reference only
- Your calculator does not have to look anything like the calculators shown in the screenshots
- Your calculator does not have operate in the same way that is shown in the movie demos

Multiple Operations

- Performing multiple operations (ex: 2 + 3 * 6 - 2) is NOT a requirement for this assignment
- If, however you would like an additional challenge of making your calculator app perform multiple operations then one possible solution would be to have JavaScript evaluate a string
- You may want to avoid using the "eval()" method for various reasons outlined here:
    - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval#never_use_eval!
- Checkout this stackoverlow Q&A for an alternative way for JavaScript to evaluate a string:
    - https://stackoverflow.com/questions/6479236/calculate-string-value-in-javascript-not-using-eval

Marking Criteria:

This project will be marked out of 20 and will be marked based on the following criteria:

Your highest grade is determined by which version of the calculator you and your teammates were able to complete:

Base Calculator

- User interface and styling of your calculator: **05**
- All technical specifications met: **12**
- Total (out of 20): **17 / 20**

Bonus 01 Calculator

- User interface and styling of your calculator: **05**
- All technical specifications met: **13**
- Total (out of 20): **18 / 20**

Bonus 02 Calculator

- User interface and styling of your calculator: **05**
- All technical specifications met: **14**
- Total (out of 20): **19 / 20**

Bonus 03 Calculator

- User interface and styling of your calculator: **05**
- All technical specifications met: **15**
- Total (out of 20): **20 / 20**