

Exercise

Monday

Introduction to Programming

(/introduction-to-programming)

/ JavaScript and Web Browsers

(/introduction-to-programming/javascript-and-web-browsers)

/ Practice: Calculator UI and Business Logic

Text

Goal: Continue to practice writing JavaScript functions by creating the calculator app.

Warm Up

- What is the difference between business and user interface logic?
- Does "interface" mean the same thing when we talk about a user interface, a door knob as an interface, and an application programming interface?
- How do you provide a function multiple arguments?
- What data type is returned from `window.prompt()` ?
- What does the HTML `<script>` tag do?

Code

Calculator

Create the calculator project that we walked through in the previous lessons. Below are the setup instructions as well as the sample HTML and JS code we used in the walkthrough. Reference the previous lessons as needed.

Project Setup

- Create a `calculator` project folder on your computer.
- Initialize a git repository in your newly created project folder.
- Create `js` and `css` folders in your project folder.
- Create an empty `scripts.js` file in your project's `js` subdirectory.
- Create an empty `styles.css` file in your project's `css` subdirectory.
- Create an HTML file called `index.html` in the root of your project directory, using the HTML below.
- Update your HTML to include the script tag to link to your `scripts.js` file: `<script src="js/scripts.js"></script>`
- Add a `README.md` file.
- Track everything with git and commit the changes.
- Create a calculator repository on GitHub and push your project to GitHub.
- Keep reading for further instructions on how to extend the functionality of this app.

Sample HTML File

index.html

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <link rel="stylesheet" href="css/styles.css" type="text/css">
    <title>Calculator</title>
  </head>
  <body>
    <h1>Calculator</h1>
  </body>
</html>
```

Sample JavaScript File

js/scripts.js

```
// business logic
function add(number1, number2) {
  return number1 + number2;
}

// user interface logic
const number1 = prompt("Enter a number:");
const number2 = prompt("Enter another number:");

window.alert(add(number1, number2));
```

Test Your Scripts in the Browser

When you open your HTML file in the web browser, it runs all scripts connected to it. If you run the project with the starter code above, you should see two prompts for numbers, and then an alert with the calculated result. If the project isn't working:

- Check that you correctly link to your scripts file and that your scripts file includes both the *add()* function and code that actually calls that function.

- Check the JavaScript console to see if there are any helpful error messages that might point you in the right direction.
- Reach out for help from a peer or teacher.

Extend the Functionality of the Calculator App

1. Create functions to subtract, multiply, and divide two numbers.
2. Verify that all four functions work by calling each function. Create a separate alert for the results of each of the function calls.
3. Instead of just displaying the number result of each of the function calls, update your `window.alert(...)` method calls to include a phrase that explains the result. For example, when calling the `add()` function and getting an 8 as a result, the alert should say something like "The addition of your numbers equals 8". Do this for every function.
4. Combine the separate `window.alert()` method calls into one call that states all of the results of each mathematical operation. For example, if the user inputted numbers are 2 and 1, the alert should say: "The addition of your numbers equals 3. The subtraction of your numbers equals 1. The multiplication of your numbers equals 2. The division of your numbers equals 2."
 - Note: you don't have to write out this more complex return in one line. Break it up into as many lines as makes sense to you, and use as many variables as you need.
5. Try something more challenging now. Call all of the calculator functions and have the alert return both the user's inputted numbers and the results. For example, if the user inputted numbers are 2 and 1, the alert should say: "2 + 1 = 3, 2 - 1 = 1, 2 * 1 = 2, and 2 / 1 = 2".
 - Note: you don't have to write out this more complex return in one line. Break it up into as many lines as makes sense to you, and use as many variables as you need.

In upcoming lessons, we'll learn how to make forms in HTML that can gather user input and streamline our calculator applications even more!

[Previous \(/introduction-to-programming/javascript-and-web-browsers/calculator-ui-and-business-logic\)](#)

[Next \(/introduction-to-programming/javascript-and-web-browsers/homework-identifying-and-preventing-microaggressions\)](#)

Lesson 39 of 75

Last updated March 24, 2023

[disable dark mode](#)



© 2023 Epicodus (<http://www.epicodus.com/>), Inc.