CPSC1520 – JavaScript Exercise 2: Introduction to Functions

Introduction

Functions are the building blocks of almost everything we will do in the course.

At their core functions are collections of statements that allow us to break down more significant problems into smaller ones and will enable us to reuse code.

In this exercise, we will use a combination of functions that we create and JavaScript built-in functions to create our "Multiplying Calculator," which will take two values, multiply them and update the dom with our result.

Exercise Step 1 – Create the "getElementValue" function.

1. Link your JavaScript file in your HTML, and print a console.log that prints out "javascript calculator linked."
2. Create a function named "getElementValue" that will take one parameter named "selector."
3. The function should **return the innerText** of the element selected (using the "selector" parameter)
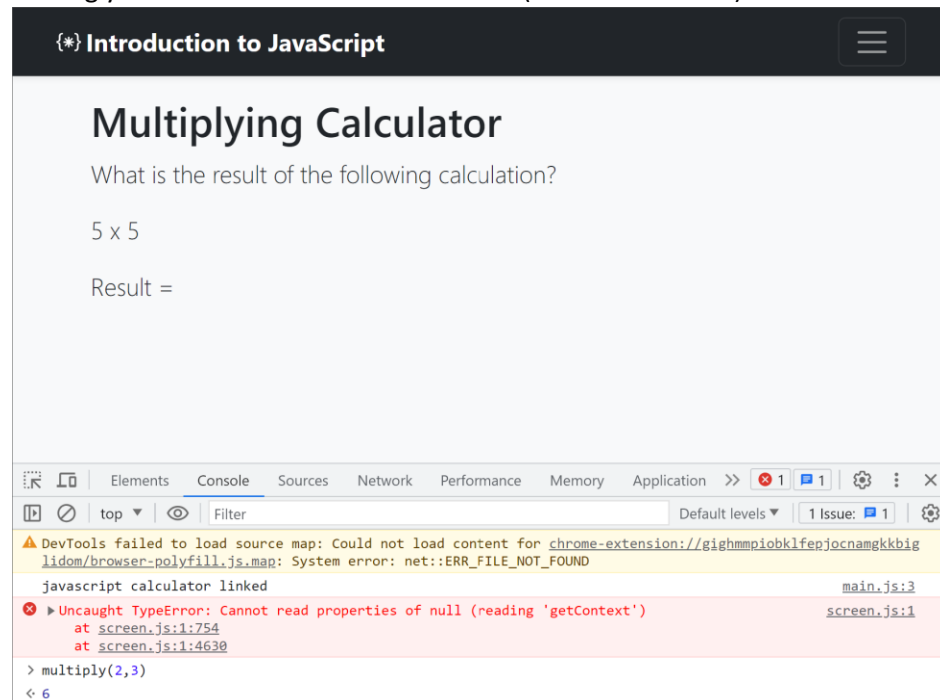   a. If you test your element it should look like the output of the following picture. (Note the console.)
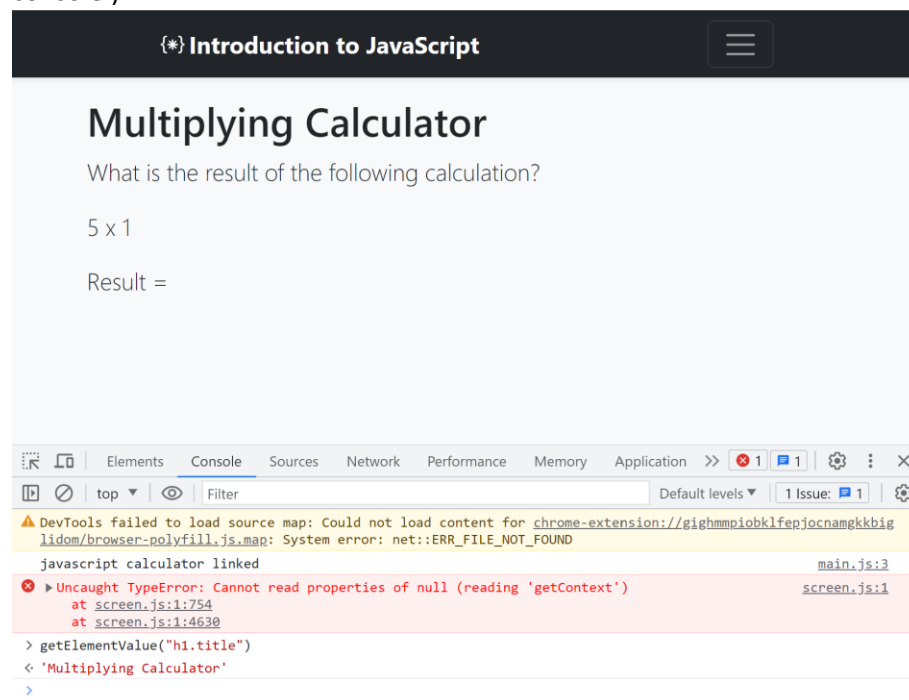
4. Once you have completed this, you can move on to the next step.

Exercise Step 2 – Create a "multiply" function

1. Create a function named "multiply" that will take two parameters: "firstValue" and "secondValue".
   a. Note when you call this function, you'll need to **pass in numbers, not strings.**

2. The function will return both numbers multiplied together. If you're confused about how to do this and want some reference https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Multiplication.
   a. Testing your function should look like this. (Note the console)

3. Once you have completed this, you can move on to the next step.

## Exercise Step 3 – Updating the Result in the Dom

1. Create a function named "updateResult" that doesn't take any parameters.
2. The first two things we want to do is get both number values that you see in the dom.
   a. Use the function "getElementValue" to get the values
3. Since these values from the dom are string values, we will use the built-in function "parseInt" to convert the string into a number. For reference (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/parseInt)
4. After we convert both input values to numbers, call the "multiply" with the values from step 3 as the arguments to the function (you should pass two arguments).
   a. Store the result of the function in a variable named "result."
5. Select the element with the id of "result" and update the innerText to the new value.

6. Call the "updateResult" function so that the result is updated every time you reload the page. The result of your application should look like this:
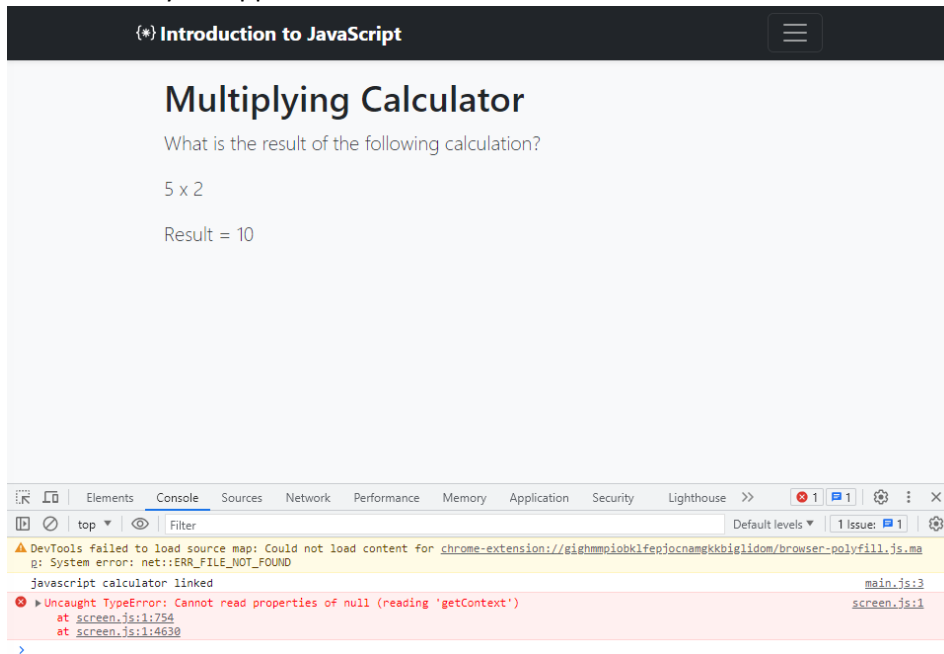
Exercise Step 4 – Push up your code to github (accepting this assignment)

1. Open the link given and accept the assignment. Your link should look something like this. Note the image will be different because you'll accept the assignment specified.
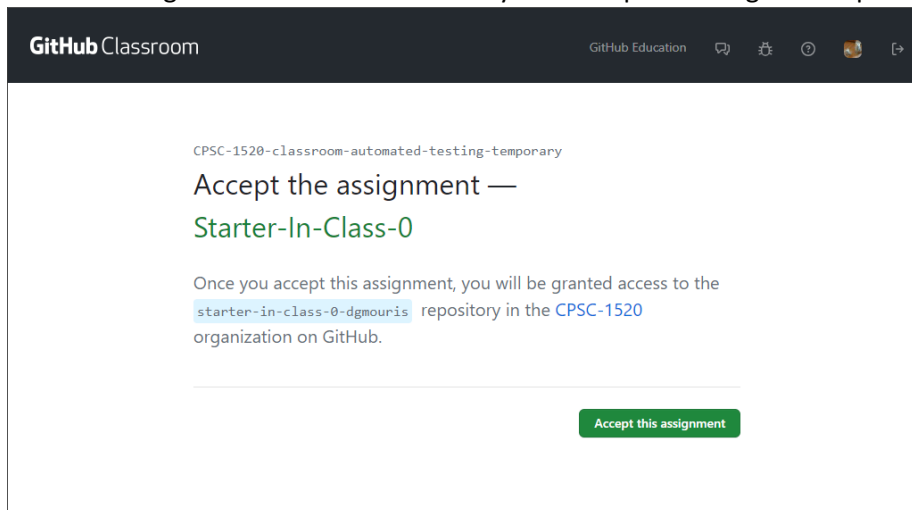
You'll see a page like this.

You accepted the assignment, **Starter-In-Class-0** . We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates.

Note: You may receive an email invitation to join CPSC-1520 on your behalf. No further action is necessary.

One you're repo is ready the page should look like the following.

You're ready to go!

You accepted the assignment, **Starter-In-Class-0**.

Your assignment repository has been created:

https://github.com/CPSC-1520/starter-in-class-0-dgmouris

We've configured the repository associated with this assignment (update).

Note: You may receive an email invitation to join CPSC-1520 on your behalf. No further action is necessary.

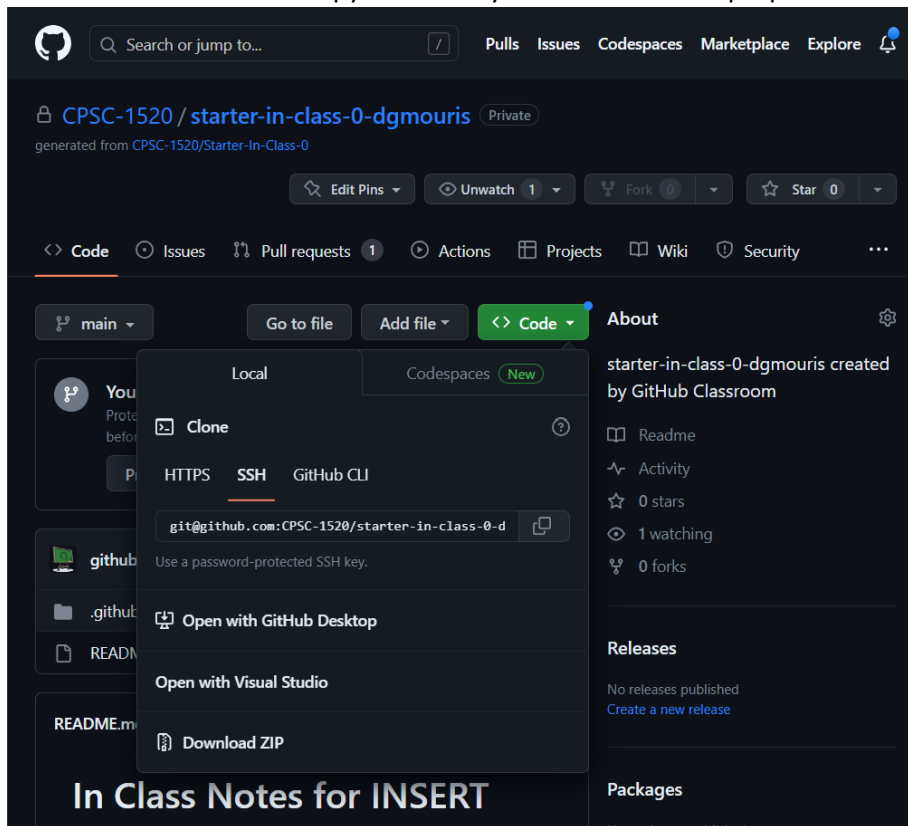2. You should see the page below once you click on the link highlighted in blue. Click the button that says "Code." You'll need to select "HTTPS" unless you've set up "SSH" (you can also set up

GitHub CLI". Click on the copy icon once you've selected the proper icon.

3. Clone the repository in your console (or if you're using GitHub Desktop) using the "git clone REPO_URL" command.

```
DMOURIS@W309-DMORR2 C:\Users\dmouris\temp
$ git clone git@github.com:CPSC-1520/starter-in-class-0-dgmouris.git
Cloning into 'starter-in-class-0-dgmouris'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 0), reused 2 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp
$ 
```

And go into this folder.
4. Make your changes, then add them to staging (using "git add .") and commit them (using "git commit -m "CHANGE THIS MESSAGE"). Once committed, push them up to GitHup (using "git

push") it should look like below.

```
DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git add README.md

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git commit -m "Made changes"
[main 9532c1b] Made changes
 1 file changed, 1 insertion(+), 3 deletions(-)

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 373 bytes | 373.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:CPSC-1520/starter-in-class-0-dgmouris.git
   b6ef88e..9532c1b  main -> main

DMOURIS@W309-DMORR2 C:\Users\dmouris\temp\starter-in-class-0-dgmouris
$ []
```
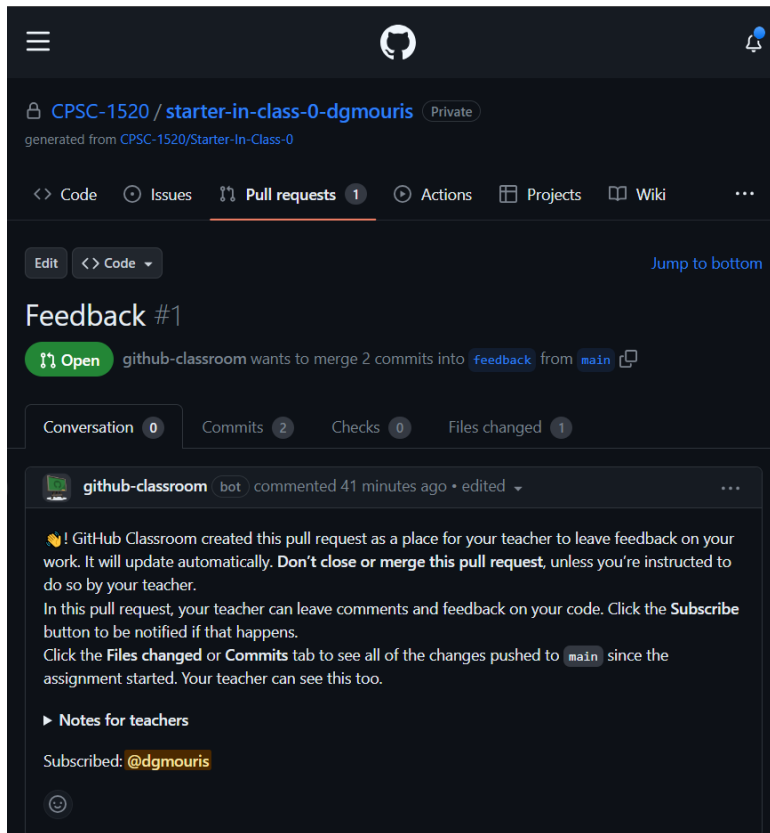
5. If you click "Pull Requests" and then the first item called "Feedback" you should see your commit (seen at the bottom).
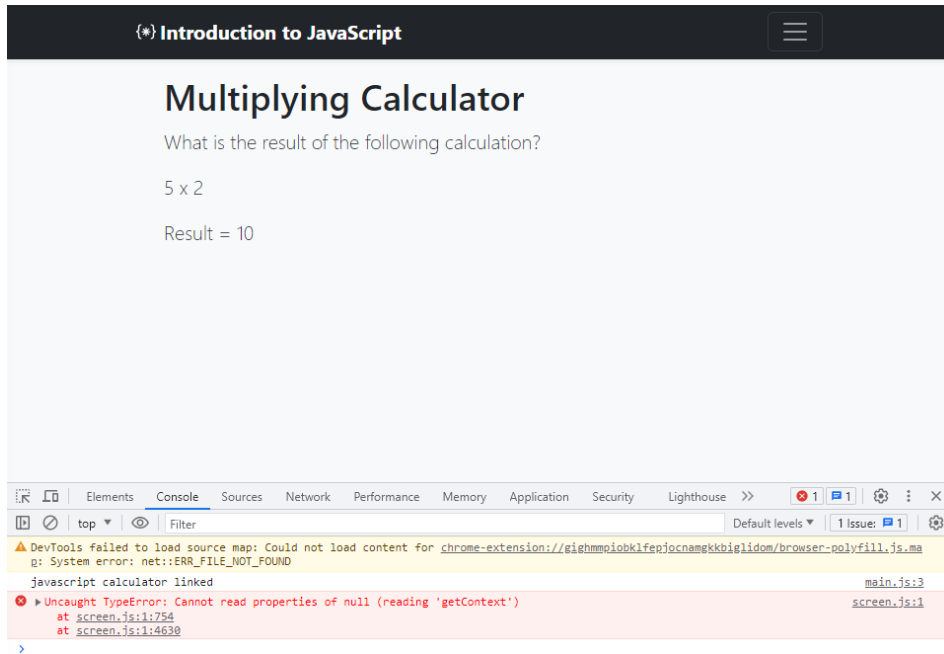
6. The link of your repository to Moodle.

Grading

I'll give full marks if:

- The name of **your Netlify app is renamed to your name**.
- All functions have the correct names and parameters.
- What I see should be exactly what I see in the picture below (your numbers will be different).

If you don't follow the instructions, you'll get a zero. There are no marks in between.