Sohee Cho
Week 3 HW

1. ***Choose* one form** and **describe** what you have ***just created*** by **attaching** the **addEventListener()** method to the ***div form button*** and what ***everything does*** in the **code** you ***just added***, piece by piece.
    - My code:

```
- const addNums = () => {
-    let num1 = document.getElementById("num1").value;
-    let num2 = document.getElementById("num2").value;
-    let sum = parseInt(num1, 10) + parseInt(num2, 10);
-    document.getElementById("result").innerHTML = sum;
- };
-
- const btnAdd = document.getElementById("add");
- btnAdd.addEventListener("click", addNums);
```

        - In my index.html I removed the onclick attribute and added the new JavaScript to the div form button. By attaching “addEventListener()” I have created a click event, which means an external function is triggered when the user “clicks” the button. I am referencing a function, which is external and in this case the function is “btnAdd.addEventListener(“click”, addNums);

2. Describe the ***difference*** between a **referenced function** and an **invoked** (aka **called**) **function**. Please ***provide*** a code **example** for ***both*** a **referenced function** and an **invoked function**. You may use part(s) of the answer to the ***first question*** in your homework as example(s), if ***applicable***.
    - The difference between a referenced function and an invoked function is that a referenced function is external. An invoked function is used immediately when the site loads.

        - example of

3. Describe the ***difference*** between **named functions** and **function expressions**. Give one (code) example for ***each***.
    - Named functions is defined by a function declaration. In order to create the function, I use the “function” keyword and add the name of the function, so it is defined.
        - example: function sohee(student) { class };
    - Function expressions are a named or anonymous function. It is not hoisted, so it can’t be used before it is defined.
        - example: let sohee = function(student) { class };

4. Describe the ***difference*** between **arrow functions**, **named functions**, and **anonymous functions**. Give one (code) example for each.
    - Arrow functions are a simpler version of a function. The syntax of the arrow function consists of zero or more parameters, an arrow => and then concludes with the function statements.
        - Example:

```
const multiplyNums = () => {
let num3 =
document.getElementById("num3").value;
let num4 =
document.getElementById("num4").value;
let multiply = num3 * num4;
document.getElementById("result2").innerHTML =
multiply;
```

    - Named function is a function declaration if it appears as a statement.
        - function cityTechStudent(name) {
            - name = “Sohee”;
            - return “My name is ${name}!”)
            - cityTechStudent( )

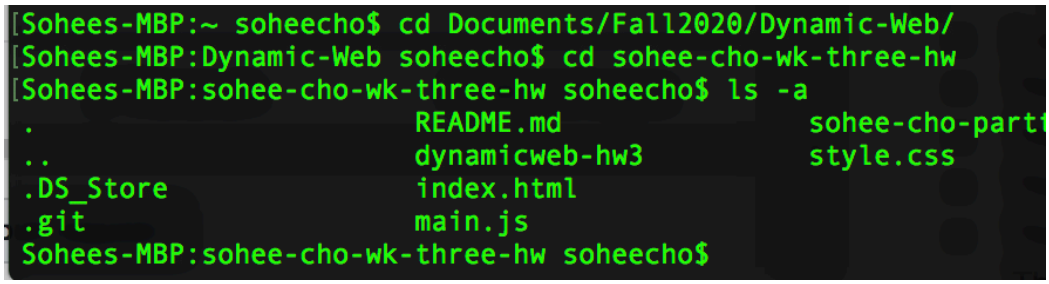
- Anonymous function is a function definition that is not bound to an identifier. They are often used for constructing the result of a higher-order function that needs to return a function.
    - const addNums = (a,b) => a + b;
    - const total = addNums(2,3);
    - console.log(total);

5. ***Trigger*** a **click event** by the ***click*** of a **button** using the **addEventListener()** method. You have ***already*** **added** the **addEventListener()** ***method*** to your **Arithmetic Forms** project, but ***now*** I want you to ***add*** it to something **new**. I have ***created*** a **repository** called **toggle-square** which you can use as a ***reference***. However, I want you to **change** the **square** to a ***circle***, and ***call*** **your exercise** ***toggle-circle***. I also want you to use ***different*** **colors** than I did. And I want you to ***go to town*** with the **styling**! I am ***also*** **providing** a ***bonus*** **function** called **clearBg()**. You will probably want to ***re-use*** the **body** of the **function** (what is ***between*** the {}) in other projects. What it does is "***refresh***" the **browser window** and ***clear*** whatever was ***created*** with **JavaScript**. In other words, ***whatever*** DOM **manipulation** was **done** with **JavaScript**. Since it is all on the ***client side***, the **manipulation** does ***not*** persist. When the **page** is ***refreshed***, the **changes** to the **DOM** ***disappear***. When you ***do*** **re-use** this **function** in ***other*** **projects**, just make sure to provide a ***new***, **descriptive name** to the ***function*** that **describes** what it is ***doing*** in that **particular project**. In **toggle-square**, I ***named*** the **function** **clearBg()** because ***essentially*** it is **clearing** the **background color change** of the **square** that JavaScript ***executed*** on the DOM.
    - File in folder as "hw-togglebutton"
6. **Describe** what **window.location.reload()** ***means/does*** exactly. I have ***provided*** a couple of **links** in the **"Helpful Resources"** ***section*** that **break down** what it **means/does. The links** should ***help*** you **come up** with an ***answer*** to this **question**.
    - The "window.location.reload()" reloads the document, which

refreshes the page on the browser.

7. **Describe** the ***commands*** you need to **initialize** a **local git repository** on your computer ***inside*** a **project** you want to ***push*** to **Github** for the FIRST time, ***add*** any files or directories to the **staging area**, ***commit*** those files, and then ***push*** them to **Github**. **List *each command***, starting from the ***first command*** that **initializes** your **local Git repository** at the ***root*** (top) of **your project**, ***all the way*** to the **final command** which ***pushes*** your **project's local repository** to ***Github***.
    - I create a repository on GitHub
    - Then I use "cd" to go into the directory file of my project
    - Then I use "git init" which initializes the Git Repository
    - Then I use "git add ." which adds my files to the Git index
    - Then I use "git commit -m 'first commit'" to commit the added files
    - Then I use "git remote add origin" along with the link to my repository to copy it to my repository
    - I check the status by using "git status"
    - And finally I use "git push -u origin master" to push all the files to GitHub
    - Voila!

9. ***Which folder*** inside your **project** ***represents*** your **local Git repository** and ***confirms*** that you have actually ***initialized*** **Git** in your **project**? **Hint:** it is a ***hidden folder*** that starts with a . (dot). When you ***initialize*** **git** in your ***projects***, ***execute*** the **ls -a command** to ***list*** all the **files** and **folders** that ***reside*** **inside** your **project folder**, ***including*** **hidden files**. That is what the **-a flag** is for. To reveal any **hidden** . (dot) **files** or **folders** in your project's **root directory**.
    - For this week's homework, the ".git" folder represents my local Git repository and confirms that I initialized it in my project.
    -



10. What ***else* signals** to you that you have ***initialized* Git** in your (project) **folder**? **Hint:** it ***represents*** your **default local branch** which has the ***same name*** as your **default remote branch** on **Github**. To ***test*** this, go into a folder that you do NOT care about, that has been "**Git**" ***initialized***. ***Check out*** what is **present** in the **Terminal** window. Then type the ls -a command followed by hitting the enter/return button. After you have determined what is present, **type** the **command rm -rf .git** in **Terminal followed by *hitting*** the **enter/ return** key. After you have done that, ***execute*** the **ls -a** command again in **Terminal** and see ***which files*** and ***folders*** are **printed out**. Has ***anything* disappeared**? In ***addition***, has **anything else *disappeared*** from the **Terminal window** as a ***result*** of **rm -rf .git**?
    - The master branch signals that I have initialized Git in my folder.