Reflection:

The most challenging thing for me was implementing the remove function because I followed the youtube tutorial to write and store my product list in session storage using a dictionary. (I used the product tag as a key and the products stored inside to be the value: `[products.tag]: products` when I stored the products to session storage.) However, I didn't realize this by myself, so I wrote some useless code by referring to the non-existing array.
Of course, the code did not work. Then I came to Jason to help me look at the code, and it was until then that I knew what I had done wrong. However, I realized that storing the products as a dictionary is easier to parse because I could delete the product with the product tag as the key. One reflection from the mistake is that I should always fully understand the code before implementing it in my design. (At least should try to understand most part)

Another challenging part is making sure the cart number will appear on all pages of the website. I did not realize that beside the window.onload function, in order to item.price

Programing Knowledge I have learned:

1. I learned how to store local and session storage using
   `localStorage.setItem('key', 'value');`
   `sessionStorage.setItem('key', 'value');`
   My code: `sessionStorage.setItem("cartNumbers", JSON.stringify(cartNumber))`

   And I understood the differences between local and session storages (localStorage stores the item even when window is closed, sessionStorage will not and when window is reopened the cart will calculate again)
2. I also learned that I need to JSON.stringify the stored javascript object and convert it to a JSON string if we want to store it into the session or local storage
   a. `JSON.stringify(value)`
   b. My code: `sessionStorage.setItem('totalCost', products.price);`

3. Another thing I learned that when we are accessing the item inside storage, we need to either parse it to an object using JSON.parse(), and if we wanted an integer, we need to parse it into a integer using parseInt()
    a. My code: `cartItems = JSON.parse(cartItems);`
4. One other javascript concept I learned during the process is adding a big chunk of html elements using innerHTML+=`some html codes`. I never knew that was available until I watched it on youtube.

My code: `productContainer.innerHTML  ` `` ` ``

```
           <div class = 'products-wrapper'>

           <div class="products">

               <i class="fa fa-times-circle remove" onclick = "removeItem('${escape(JSON.stringify(item))}')"></i>

               <img src="img/${item.tag}.png" class="product-image">

               <span class="product-title">${item.name}</span>

           </div>

           <div class="product-price">$${item.price},00</div>

           <div class="product-quantity">

               <i class="fa fa-minus-circle"></i>

               <span class="product-number">${item.inCart}</span>

               <i class="fa fa-plus-circle"></i>

           </div>

           <div class="product-total">
```

```
            $${item.inCart* item.price},00

            </div>

            </div>

            `;
```

5. I also learned about the escape function, which encodes a string and converts it to ASCII characters to transport. I used `${escape(JSON.stringify(item))}'` in my code to pass the string I parsed from storage to remove one specific product. Later when I want to access the string, I also need to unescape it using `JSON.parse(unescape(objString))`