Reflection

BUG #1: Children vs ChildNode for remove function

One thing I struggled with for my remove cart function was finding the child of the div block in my HTML file and trying to remove that element once the item is removed, so that that item will no longer show up on the page. At first, I tried using .childNode[i] to find that specific div block, but the end result was not what I expected. I had to click three times on the “remove” button for the block to disappear. After doing some searches on Google, I learned about the distinction between children and childNodes, one of the main ones being that children work upon elements and childNodes work upon nodes including non-element such as comment nodes. In my case, I should have used children instead of childNodes.

BUG #2: Integer vs String

For the price calculation section, I made a mistake and forgot to parse the returned value from localStorage, and so when I add a new item to the cart that was priced at $5, instead of incrementing the total price by 5, there was an additional digit at the end of the original price. I then realized that the number was being treated as a string value instead of an integer, and that is why the “+” function appended the number to the end of the string. I pinpointed where the function went wrong and added JSON.parse() at the beginning and fixed the issue.

BUG #3: Null element on one page

The function for adding to cart spans across two different pages -- the product detail page and the cart page. However, when I was writing the function, I initially used some functions such as querySelector and placed it under the wrong place, so for some pages, there was no such an element with the given id/class on there and the console would indicate an error. I later

Programming Concepts

1. **Local storage vs session storage**
   One of the main programming concepts I learned in this assignment was using local storage and session storage to store data. In this case, I used local storage for the adding cart related functionality, such as products added, price, and count, so that they are visible across different pages and sessions, and the cart values remain the same. Related functions I used including `setItem(key, value)` `getItem(key)`, and `removeItem(key)`.

2. **<template> tag in HTML**
   I used the <template> tag in my cart.html page for each individual cart item. In this case, when a new item is added to cart, I can reuse the template code, make a clone, and add it to the parent div. It is very time-efficient and ensures consistency across different items.

3. **Event Listener**
   I also learned more about using different event listeners, and I used mostly onClick and onLoad listeners for this particular project. For instance, I added an onClick event listener to the add to cart button that will trigger an update to the cart, and there is an onLoad listener for the cart page that will trigger the function to display items according to the cart.

4. **Removing and adding to an array**
   I also learned more about arrays as I used an array to represent my cart. For instance, to remove a product from the array, I used .splice() which specifies which index to start at and how many items to remove. To add a product to cart, I used cart.push() to add that new product to the end of the array.

5. **Representing products as Objects**
   Last but not least, I learned about representing products as Objects using the object constructor. This allowed me to create a new product for the addToCart function using the object constructor.