Github repo: https://github.com/shalinidrao/homework_6
Github Page: https://shalinidrao.github.io/homework_6/

Shalini Rao

Homework 6 Reflection

One problem I had was understanding how local storage works in order to store all the cart items to be displayed and how to save each item. Objects ended up being the most logical way to do that, since I could create the necessary attributes that would be pulled from the inputs the user gave when they added an item to the cart. This way, whenever the “Add to Cart” button was pushed, the cart counter would be incremented, the button text would change, and a new Bun object would be created for the newly added item. I had a problem where only one item would show on the cart, no matter how many I added. I realized I needed an array to store all the added items, and then display them by parsing through the array and formatting the information stored in each object.

I also struggled with understanding how appending a child works. I had problems formatting each cart item in the same way I had it in the static page, with a horizontal rule beneath each item. The “Add more items” button was appearing before all the items, instead of after. I was appending to the cart wrapper, so I fixed the issue but creating an empty div in the cart wrapper, and appended to that div instead of the cart wrapper as a whole. I also appended a horizontal rule to this empty div, instead of to the product itself.

Programming Concepts

Objects

I needed a way to hold all the information the user put in for their options for glaze and quantity. To bundle them all together in one item, I made a Bun object that stored the bun flavor, glaze, and quantity. A new object would be created each time the user pressed the “add to cart” button with the options they had selected. I could then pull each of these attributes of the object for displaying the cart in the desired format.

Local Storage

I used local storage to keep track of the number of items in the cart as well as the items the user has added to the cart. I use the getItem function to obtain the number of items in the cart to display next to the cart in the navigation bar. Each time the user pressed the “add to cart” button, a function would be called to increment the cart counter and update the variable in local storage. It would also stringify the newly created Bun object, add that to the cart items array, and set the cart array in local storage to the newly modified array.

Arrays

As mentioned above, I used an array as a global variable to store all my cart items. Each time a new item was added, a new object would be made and pushed into the array. This also made it possible to display all the chosen items in the cart, allowing the user to add multiple items and

see them all in their cart. It was very interesting to learn that I could store objects in the array by using the stringify method and then parsing it later when I pulled each item out.

For Loops

For loops in Javascript work similarly to other languages, but the counter variable was interesting to figure out, especially with let and var both being ways to define a variable. I ended up rewatching the recitations to understand the best way to keep a counter for the for loop. I used a for loop to parse through my cart items array one at a time to display each item on the page.

DOM Manipulation

This was by far the most interesting part of using Javascript on this website. I used multiple Javascript functions to build new divs, create text elements, and set attributes of items to be displayed on the cart page. It was especially difficult to keep track of the nested divs necessary for the grid format I made on the cart page. I kept my existing static cart page on the side to keep track of what elements to create in what order, and in what order to append them to my cart div. I used the appendChild() function to add each new element to a div (each column), and then added those columns to the cart wrapper. It was helpful to draw out a node tree to keep track of what nodes were children of others.