

Reflection

This assignment was much more interesting and certainly enjoyable to work on and code than the previous few assignments. It was great to connect pieces of the visual and structural elements to some “back-end” that I could manipulate and craft to create a website that was cohesive in some sense. It was great to be able to have a user interact with a button or input, and then to see that data and decision stored somewhere within the JavaScript to be used later. Grabbing the user input proved to be quite difficult for some time. For a while, I was using a select element with multiple option elements within to structure the Quantity selector on the product details page. For some reason, the function I set up to pull the value the user input was not working, but everything was pointing to it working, `console.log()`s, parsing string to integers and all. When I changed the HTML to an input element, it worked magically and easily!

Another thing that was interesting to learn over the course of making various functions and connections between different pieces of the website was that some functions would work well in one specific place, but then be out of place and broken when another function was in place and set up. For example, I have a function that updates the cart counter when items are added to the cart. This function counts the items in the cart off of an array that is established on the product details page, where items are added to the cart and array. This function broke and did not work when I started working on sending the cart to local storage and retrieving each time pages were unloaded and loaded, respectively. Learning how to transfer the functions I already have, such as updating the cart counter, to new situations that provide the same end results was key for this assignment, as I updated some functions to work across pages, rather than within just one page.

Programming Concepts

Within this assignment, I used a few programming concepts that were critical to the running of the website and its functionality. Specifically, **classes**, **constructors**, and **functions** were used to build cart items and to ensure that input from the user was collected and then displayed on the cart page. For example, when a user hits the “Add to Cart” button, a function is called that will do a few things. In this function, the elements that are labeled by size are grabbed and put into a variable. Then, a **loop** iterates over the variable and tests if any of the items are checked (a property of radio

input elements). If yes, the value of that element (set in the HTML) is assigned to the size variable to be put into the Product Class later. This process repeats with the color buttons/elements, as well as pulling the quantity value from the quantity element/input. Later in the function, all these pieces and variables are used to create a product class, and then this product was pushed into an array for showing later. Lastly, the function will run the “UpdateCartNumber()” function to update the cart counter. Lastly, I used **local storage** and transference to **JSON** to move the cart (the array) to different pages to ensure that there was consistency between pages. This piece would send a stringify-ed version of the array to local storage on unload of the page, and then would be retrieved when the next page loaded. Upon retrieval, a function would be run to update the cart counter to the current length of the cart. On the cart page, in particular, another function would be used to iterate over the cart to change the inner HTML of the cart page to display each item in the cart.