Assignment 6B Reflection
Aziz Ghadiali

Issues/bugs I encountered

Syntax errors

Some of the most common syntax errors that I encountered while writing my code were misspelling variables and functions and occasionally having an extraneous curly brace. One of the reasons that I had this issue was that I created variables that had similar names to each other and were not distinct enough leading me to sometimes write the wrong thing. I can fix this in the future by trying to be more descriptive with my variable names. Another reason I think that I encountered these issues was that my class/id names were also sometimes not very clear or descriptive of the HTML element(s) they corresponded to. In the future, I would again focus on being more intentional in the names that I use for ids and classes so that they are easier to understand. One of the other reasons that I think that I had issues with misspellings was that once I got an idea on how to solve a particular issue, I began to just write out the code to get the idea down and then go back and correct the syntax. For example, I ran into an issue when I accidently used a single equals sign in a series of conditional statements without realizing it. I don’t think that this process really caused any significant problems, however, in the future for more complicated projects I would probably write down pseudo code first and plan out each part a bit better prior to typing out the code so that I could focus more on making sure I am writing things that are syntactically correct and to save time.

Runtime errors

I had a few runtime errors while writing my code but they never really posed a significant issue. They usually occurred because I had misdefined the scope of a variable or writing in the wrong JS function. Again, I would probably avoid these issues in the future by planning out my code design in advance so that I could accurately scope the necessary variables and understand the functions I would need to use to complete different actions. However, I also do feel like some of these things are inevitable while coding and that is why just using Google to resolve these minor issues feels equally as effective.

Logic errors

I ran into some logic errors mostly while working with local storage. Most of it was related to me trying to understand how the data was stored and retrieved and so I often tried to use functions that accessed the wrong items, performed incorrect operations (such as treating a string as an int before casting it), and sometimes just forgot to populate certain fields. I think that in the future reading a bit more detail about how the structures I am using function would definitely help to prevent some of these issues.

Five JavaScript concepts

Conditionals

The first JavaScript concept that I learned how to use was conditionals specifically for the purpose of determining how to populate the contents of my cart. For example, in my JavaScript code (product-page.js), I used a series of four conditional statements to determine what product image to use based on the selected color of the item. I run these statements everytime a color is changed so that the referenced product image is always updated.

Local Storage

The second concept I learned was local storage. I had understood what the concept was prior to this project but actually used it in code for the first time. In my project, I used local storage to store the information selected for a particular product item (color, size, quantity, cost, and image url) every time the add to cart button was clicked. Using the get method, I pulled the information from local storage to populate the items shown on my cart page by creating a new cart item for every item in the local storage with the specific selections for that item.

Dynamically rendering DOM elements

The third concept I learned was dynamically rendering DOM elements using the create element, set attribute methods and the innerHTML property. For example, I used dynamic rendering to create new DOM elements for each of the tags in my cart items. To do this I started outlining the structure of my cart items and worked backwards to populate all of the higher level divs with their internal content and adding them to the DOM. Using this concept I was able to actually use the data in my local storage to actually render the cart items on the webpage.

Scoping variables

The fourth concept that I learned was variable scoping (global vs. local). For example, in my code I used global variables (using the var declaration) such as “productItems” (contains the data retrieved from local storage) to store values that needed to be accessed in multiple places in the script. I also used localized variables (using the let declaration) for variables such as my for loop indexes since they were not globally scoped.

Event Handling

The fifth concept that I used was event handling to run a function when a button is clicked. For example, I used an event handler to detect when a remove button is clicked and remove the appropriate item from the cart. To do this, I created a for loop that goes through a list of all of the remove buttons in the cart and listens for a click. When a remove button is pressed, I store the specific id of the remove button and use that to match it to the correct cart item. I then remove the cart item, update the local storage, and then update the HTML.