Sarah Chen
PUI

One of the challenges I faced was not fully understanding functions and methods in javascript and how they could be used. Even though I followed a tutorial, a lot of the steps had to be changed because the code didn't work for me. This led to a lot of Googling and searching for possible causes of the bugs, and possible workarounds. Sometimes, I didn't understand why an error occurred or what it meant, and so I had to figure it out myself or ask for help. I often went to documentation sites to figure out what the methods were intended to do, and what inputs they can take. From there, I had a decent foundation to figure out why my code was bugging and how I could fix it. For example, I had trouble figuring out how to delete items from the cart because I would run into errors when I tried to pass the item to the remove function. After searching for other methods, I stumbled on "this" on StackOverflow. Then, I was able to look into how "this" worked, and how I could use it. As a result, I was successfully able to call the remove function with its correct input (the color) with this.getAttribute("name"). I have learned that the internet is a great resource for both debugging and troubleshooting, and will likely be using it extensively in the future.

Another challenge was shifting the CSS and HTML to what I had already. For instance, the shopping cart from 6A was hardcoded, so I had to erase all of my work and reimplement it in javascript. Fortunately, a lot of the CSS styling could be saved, even if I had to make some positioning changes. Still, it was hard for me to let go of the work I had done before, even if the interaction was a cool new feature that I liked. Even though I encountered a lot of errors and things that didn't work the way they should have, seeing a working final product that is interactive is very satisfying.

1) A programming concept that I learned the nuanced differences between localStorage and sessionStorage. Even though we discussed it during lab, it remained an abstract concept before I actually used it in my own projects. localStorage can be good for saving things across tabs, but also made it difficult to debug. As a result, I ended up using sessionStorage for all of my storage purposes so that I could reset everything when I opened a new tab. I used it to store various things I need across pages, such as the items in the cart.
    a) Example: sessionStorage.setItem("allCartProducts", JSON.stringify(cartItems));
2) Another programming concept I learned was that I could change HTML code in my javascript file using .innerHTML, such as when I set the subtotal amount to what all of the items would cost. Because I used .innerHTML, it allowed the amount to update as more things were added to the cart.
    a) Example: subPrice.innerHTML = cartCost;
3) Another programming concept I learned was the importance of changing types of some variables. For instance, I needed to change the number of items in the cart to integers each time I retrieved it. Otherwise, I wouldn't be able to add the number of items added to the cart to the total because the types weren't compatible.
    a) Example: shoppingCartCount = parseInt(shoppingCartCount);

4) Another programming concept I learned was the importance of using the console for debugging, especially while I was working with javascript. When I wanted to check what a variable was or what a function was outputting, I would use console.log either directly in the console or in my code files. This way, I was able to see if the code I had written did what it was supposed to. For example, I was able to use console.log to check the state of the items in my cart and see if I was successfully adding new items in.
    a) Example: console.log(cartItems);
5) Yet another programming concept I learned was the usefulness of if-else statements. This really helped me in areas where I wanted to do one thing if a certain condition was true, and another if it was false. One of the most common ways I used it was for my function to do a certain thing if no inputs were made (no items in the cart, nothing selected, etc.), and then use the else statement to handle if the if statement was false. I was also able to just use the if statement on its own without using else. For instance, I used it to set the subtotal in the shopping cart when there are items selected.
    a) Example: if (subPrice != null) {subPrice.innerHTML = cartCost;}

Tutorials used:

1) https://www.youtube.com/watch?v=IY5UN82FZ2Q&list=RDCMUCADAkBGiLWIPkCu8D1R1M6g&index=5
2) https://www.youtube.com/watch?v=l2M5q2te234