Will Kuhlke
PUI
08 November 2020

Assignment 6

Repo: https://github.com/future-will/Assignment_6B
GitHub Pages: https://future-will.github.io/Assignment_6B/

Reflection

This was a rather difficult assignment for me. I have some experience working with JavaScript from the course I took over the summer, but I had no experience passing data between separate HTML pages. The concept of stringifying and then parsing values to different keys wasn't concrete until I was completing this assignment. However, having used these concepts, I think I have a better grasp on using local storage. I started running into bugs when adding different product objects into an array that was then added to local storage. The issue came from the fact that there were essentially 2 separate arrays that I was juggling at any given time: the array of products pulled from local storage and the array that I was building on the product details page. The bug occured when I went to add the array to local storage. I was essentially writing over the previous value in local storage rather than adding to the data that was already present. I was able to overcome this but parsing and then adding the contents of local storage to a fresh array every time a page loaded, and then new products were added to that same array before being placed back into local storage. As you might be able to tell from the way I am writing about this concept, I am still struggling to understand exactly what is happening in the local storage process, but I believe that I have enough practical knowledge to use local storage as a tool in JS.

The second bug I struggled with was using local storage and innerHTML to create HTML elements on the cart page. I had to use w3schools to understand that I needed to loop through the cart items array in order to build multiple divs and their respective child elements. The other issue I had was that because I used IDs to style the cart items, I needed to use the loop to add differentiation (id=\"cartItem" + i + "\") that could be referenced later in the remove function.

Learned Programming Concepts

1. **Key/Value pairs** was a concept that this assignment made clear to me. Arrays made more sense to me at the beginning of this assignment, and I was struggling to see that value of unordered lists of information that were indexed using a string as a key. The best example in this assignment was the key/values in local storage, verses the array created to store the product objects. I see now that their purpose is different, one is to keep a relatively ordered list while the other is more of a storage and reference tool.

2. **Specificity hierarchy** is another concept that was solidified in this assignment. At first they seem like arbitrary differentiators for HTML objects, but having a working understanding around the need to change groups of items versus specific items made this concept crystallize for me. For example on the cart page, I used class selectors to style all of the divs at once, I could use ID selectors to affect individual elements like when running removeCartItem(i).

3. **Booleans** were another concept that I had to gain a better understanding of for this assignment. I had a simple understanding previously of boolean functions, but here I had to combine boolean values together. In my fieldsCheck() function I used the combination of multiple boolean values to ensure that all the required information was selected before allowing the user to add a product to the cart.

4. **Javascript objects and constructors** helped me complete this assignment. I used objects to build new products to add to the cart array. The constructor was called in the addToCart() function and was used to pass the subsequent information (name, size, color, price, quantity, image) into the array to be loaded into the cart page later.

5. **HTML DOM** was a concept learned in class that was made practical to me in this assignment. I think that this idea is also combined with the specificity hierarchy concept mentioned earlier. For example in the strawberrySelect(), blackberrySelect(), crazyberrySelect() and fireOrangeSelect() functions (which I think could have been one function) I used the DOM selector getElementById() to manipulate what image was being featured based on the selected color.