PUI 2019 - Dennis Check - Assignment 6B

Reflection

I encountered several major bugs or errors in logic during the construction of the cart add and remove functions.

My first major stumbling block had to do with local storage. The idea of creating an array of objects and pushing new objects to local storage was clear to me from the outset, but when I began trying to load the objects on the cart page, I found that my local storage was persistently being cleared between pages. Using console.log and stepping through my code in Chrome developer tools, I realized that this was happening because I was pushing an empty cart array (the array's initial position) to local storage each time I loaded the detail page. I solved the issue by crating a logic gate; an empty array is now only written and pushed to local storage on page load IF local storage's value is null (i.e., if there is no array existing.) I was also able to help a couple classmates who were having this same problem.

In populating my cart, I tried a few methods before hitting on one that made sense to me. First, I tried keeping the first cart item hidden until the first item was added, then cloning that initial cart item for each subsequent item in the cart array. I had trouble following the loop logic and was not sure how to modify that first item, so I switched tactics. Instead, I constructed new divs from scratch for each for each cart item, and used innerHTML to modify the content based on object attributes from the local storage array. This proved more straightforward, after I learned how to use single quotes rather than double quotes to turn my HTML into 'plaintext' in that section.

I made use of W3 Schools' resources to learn about various JS functions and found they usually addressed my exact questions. I also looked at Stack Overflow posts to get ideas on how to solve larger problems, such as appending new nodes.

I encountered a major problem when building the remove functionality (the proximate cause of my late submission), by somehow pushing the string "deleteArray" to local storage as an array (I think.) I stepped back as far as I could with undo but the damage was not undone; eventually I went step by step and fixed several logic errors in my JS using console.log. This made me realize I should be committing to GitHub more often to have stable versions available if I muck up my code in an inscrutable way.

5 concepts

1. innerHTML - I used this to edit the HTML of each cart item's div, which is powerful when combined with concatenations of javascript expressions. For instance, I used `document.getElementById("holder"+index).innerHTML =` *HTML + JS + HTML* as the basis for the populate function.

2. localStorage - This is the basis for keeping the item information from the detail page to the cart page; I learned how to add to it and remove from it, and how to keep from accidentally clearing it with expressions meant to initialize an array. For example, I used `localStorage.setItem("cartArray", JSON.stringify(cartArray));` to store the cart array

3. Constructors - I used a constructor to set up my cart objects and give them the ability to take on certain attributes.

```
function cartItem(quantity, fabric, stuffing, price) {
        this.quantity = quantity;
        this.fabric = fabric;
        this.stuffing = stuffing;
        this.price = price;
}
```

was the chunk that set up these items, and later cartItems were pushed into local storage for later access.

4. JSON - I used JSON to pass data between the user interface and local storage, and learned about the inverse functions stringify and parse to convert between storable and usable forms of data. For instance: cartArray = JSON.parse(localStorage.getItem(“cartArray")); and localStorage.setItem("cartArray", JSON.stringify(cartArray));

5. Chrome developer tools - This was my first experience using some parts of the developer view extensively; I had mostly used it for inspecting HTML elements in the past, but I eventually realized it allows you to monitor local storage without using console.log statements, as well as clearing local storage for testing. I used it to understand issues with what was or was not being pushed to local storage.