

**GitHub Repo Link:** <https://github.com/claireelee/puiassignment6b.git>

**GitHub Page Link:** <https://claireelee.github.io/puiassignment6b/>

## Reflection

One usability improvement I made from the last submission was to include a short line in the “Add to Cart” JavaScript function that makes the page scroll up so the user can see the number change in the cart, whereas before they would have had to manually scroll up to see it.

The main source of my trouble during Javascript implementation came from the fact that I wasn’t familiar with Javascript to HTML rendering rules - for example, Javascript doesn’t render with line breaks when mapping outputs to the HTML file, even if the HTML file contains line breaks between the outputs. The best solution I could find in order to fix this was to import that particular section of the HTML file directly into the Javascript file and edit it from there.

Another bug I encountered while I was implementing the “delete item from cart” function was that the delete button would only delete the most recently added item from local storage instead of the item that the user selected to delete. To resolve this issue, I used Javascript to assign a unique number ID to each item added to the cart, then used the ID as the identifier to delete the item. I think this is a particularly good practice to keep note of in the future, in creating any page that contains items that require unique identifiers.

## Programming Concepts

1. I had to use JavaScript constructors to create my item objects with stored attributes. I then used the constructor to call individual attributes of the whole item in the location it was needed (ex. Placing `this.flavor` in the HTML text line that corresponds to flavor, placing `this.glaze` in the glaze HTML line)
2. I used JavaScript to access the local storage in order to store the information from the order page, such as the item count or the attributes of the selected items in the order page, then called them to display on the cart page (ex. Using `window.localStorage.getItem("cartItems")` to get the count of the items in the cart)
3. I used `console.log()`; while implementing numerous functions in order to check that the proper information was being transmitted from one function to another (ex. Validating the number of cart items, validating the correct array of item attributes)
4. I used Javascript if functions to differentiate between certain states and have the site respond accordingly. (ex. Using an if function to validate an “active” vs. “inactive” state when the user clicks a selection/doesn’t click a selection)
5. In my JavaScript file, I worked with both global and local variables depending on whether the variable needed to be called in multiple functions or just within a single function. (ex. The variables storing the current attribute selections were used across multiple functions, so they were global variables.)