

Reflection

I faced a lot of challenges while trying to ensure the JavaScript code I've written was executing correctly, or performing the correct actions. Since the website has HTML, CSS and JavaScript components, it was often difficult to see where the bug is originating from. For example, when I wanted a JavaScript function to run when an HTML element is being clicked, sometimes I would be unsure if the problem arises from the placement of the function call in the HTML file, or if it comes from how the code is written in the JavaScript file. A way that made debugging easier for me was putting alert messages in my JavaScript functions, as a way to check if the code ran to that point. It was also challenging to find which JavaScript methods corresponded with how the HTML is being modified. For example, when I wanted to set the onclick event of an element to a JavaScript function, the JavaScript function would always run without the user ever clicking on the element. Initially I thought it was a problem related to the web page loading, but after searching for some similar errors online, I understood that it was because the function runs when it is being set to the onclick event. I learned that the solution to this issue was to wrap the function you wish to set, inside a generic function, so that it can be returned to the HTML element property.

Programming Concepts

1. JavaScript Arrays
 - a. I declared a shopping cart array for storing items the user will add. I add to the array and also update it when the user makes changes to the cart.
2. localStorage is used to store values with no expiration date
 - a. I used localStorage when storing which product to display on the product details page. I set variables such as name, product image, and price in localStorage. Then I retrieved these variables from localStorage to display on the page.
3. sessionStorage stores variables for one session, or until the tab is closed.
 - a. I used sessionStorage for storing a shopping cart array that contains items the user added to the cart. I retrieve from sessionStorage to display these items, and I update the array when the user deletes an item.
4. JavaScript objects as a collection of properties
 - a. My shopping cart array contains JavaScript objects, these are items with properties such as price, quantity, image. I store these properties in the item objects so I can save what customization the user selected and added to the shopping cart.
5. JSON.stringify
 - a. Since localStorage and sessionStorage cannot store arrays, I used JSON.stringify to convert arrays to JSON strings. I then used JSON.parse to parse these strings back to arrays again.