PUI Assignment 6B

Web Prototype with JavaScript

Directions

a. The ability to remove items from the shopping cart.
b. For JavaScript Files:
    1. Interactions should work without error. Use the Developer Tools in Chrome to help you with this.
    2. Indent and comment your code; follow the JavaScript style guides (see above)

Code

https://github.com/sophiaperson/pui_homework_6b

Reflection

Directions

a. You should clearly demonstrate what issues / bugs you encountered, what you learnt from them and how did you resolve them. A good reflection will demonstrate a clear understanding of the issue, and how it may be mitigated in the future.
b. Writing should use appropriate style and clearly to convey the writer's concepts (this includes grammar).
c. Writing should demonstrate reflection on actual events and analyze these events to draw appropriate conclusions.

Not realizing that scripts run every time page is displayed

I initially ran into trouble utilizing sessionStorage to display my cart size notification. Although I understood how to store the cart as an array turned into a string as a value for a key I would call and had had no problems with parsing the details as a JSON, I did not realize that my script was not running on all my webpages and that my script was actually creating a new value for the key every time the window loaded. After I understood the concept that my entire script would run every time I loaded the page, I ensured that the initialization of the key and value pair would only happen if sessionStorage was empty.

Issues with underlying data structure storage

As I was working on my website, I realized that I should have spent more time thinking about the underlying structure of how data would be stored behind the scenes. For example, since I was more concerned with the formatting of the page, I ended up hardcoding the images and HTML formatting of the elements in the products pages, which would make the website difficult to change modularly over time, if I were to change the details of certain products. It also would not be feasible to write the same lines of code by hand for every new product added to the website. Although this could work out for an early prototype, in future iterations, I would want to store the information about which items would be displayed on the products page in an external file, perhaps a JSON file or CSV file, and write code in

JavaScript to parse the file for product information to display. This would also ensure that the MVC framework would not be violated as egregiously.

Issues with asset management and organization

Similarly, considering more of how to proceed with my webpage, I realized that I would want to save my images in a directory in my repo, as to ensure that they would always be visible on my site in case the site they are currently hosted on ever decided to remove the photos from there, and to make the organization of my assets more organized. Had I started in this process earlier, I would have been able to actually include the different images I wanted to for the slideshows on the home page and the product detail pages.

Issues with styling

More in the styling, I also wanted to include more information in the sliding shopping cart notification, but I did not have enough time to implement it. I wanted to display the currently added item with its size, color, quantity, and price, as well as the current cart subtotal. Within the cart page, I would have wanted to style the cart total information such that that section would be fixed on the screen so that the user could have constant visuals on their cart pricing. I also had initially wanted to add functionality to the cart so that the user could change details of their cart items from their cart page for increased user control and efficiency, but I did not have enough time to implement that functionality. In future iterations, I would plan to restyle the cart page and the cart sliding notification. I would also consider how to link the selection details in the cart to the sessionStorage data.

Programming Concepts

Directions

a. Demonstrate 5 programming concepts that you learned in JavaScript and used in this assignment with an example.
b. Writing should use appropriate style and clearly to convey the writer's concepts (this includes grammar).

Data types

I used many different data types in this assignment. For example, I used numbers to store price and quantity data, which I used in calculations. I also used strings when I went to display product name, size, and color. I used a mixture of data types to represent product information in the cart implementation.

Loops

I used loops to iterate through color options and display the one the user had hovered over. I also used loops to calculate the subtotal and total price of the cart items.

Objects

I used an object to store item information for the items the user adds to their cart. This object is pushed into an array, another type of object, that contains all the cart items that user adds. This object is turned into a string that is the value to a key in the sessionStorage. This value is later parsed again as JSON, JavaScript Object Notation.

Model View Controller

I used the Model View Controller (MVC) framework throughout the assignment to separate out the different aspects of the GUI. The HTML mostly handled the model, the CSS the view, and the JavaScript the controller. In a more specific example, with the items in the cart page, I used JavaScript to write the code that would create elements to be displayed in the DOM. For each cart item, I used JavaScript to do the calculations of price and used CSS to style the HTML where the item would be displayed. I also used JavaScript to attach an event handler to the "remove" button for each cart item to allow the user to control whether the item was in their cart. Clicking this control would update the model, which would update the view.

sessionStorage

I used sessionStorage to save the data of what items the user adds to their cart across the multiple pages of the website. This sessionStorage helps display the notification indicating the number of items in the user's cart and the items in the cart displayed in the cart page.