

Adding Functionality to a Website with JavaScript

Elizabeth La

3/25/20

Assignment 6A

1. Prototype- I kept my design pretty simple; I chose to display the items in the cart on the left side and show the cost breakdown on the right side so that the customer doesn't have to scroll down all the way to the end of the shopping list to view the final price. In my ideal world, I would also have a button that allows the user to easily modify their order – i.e. change the quantity of item in the cart—but was not able to implement this in code (at least, not yet!)
2. All the buttons in my products page work, and adding to cart changes the cart icon on the top right of my website. JavaScript was also added to update the price of the item on the products page, based on the quantity selected.

Assignment 6B

1. JavaScript was used to store all items in the cart and wishlist (specifically, through cat backpack and dog backpack products page), and to display it all on the shopping cart page. From this, the total number of items in the customer's cart, along with the total price was calculated. Finally, customers are able to remove items in both the cart and wishlist.
2. Reflection- Something I really struggled with was updating the localStorage, and making sure that all the data was kept and shared throughout the different pages. I used a dictionary to store relevant data (item, color, size, etc) which wasn't that challenging to implement, but I didn't know quite how to initialize it. At one point, I hard coded the initialization of the dictionary on the home page because customers would land on that page when they first visit the website, but then I realized that every time the user goes back to the home page, the cart/wishlist would reset. I then tried to find out how I know nothing has been stored yet in localStorage, and searched if it has a length attribute I can look up if it's "undefined" in the beginning of the session. Then, I learned that I can just check if a variable I would like to keep track of is null, and initialize it at that point to set it! Overall, this assignment was really challenging, but I learned a lot of JavaScript because of it and it was definitely rewarding and cool to see the content of my webpage change through my inputs.
3. Programming concepts-
 1. document.getElementById(): This was extremely helpful in accessing different variables and text. An interesting way I used this was to check that the user has selected both a color and size before adding an item to cart:

```
if (document.getElementById('size').value === "") {
```

```
        alert("Please select the size before adding to cart!");
    }
}
```

2. `innerHTML`: Utilizing this allowed me to display all the items in cart through a for loop—without it, the shopping cart would not at all be based on what the user actually added to cart.

```
    for (var i in items) {
        itemDes = document.createElement("p");
        itemDes.innerHTML = editTxt(i);
        ...
    }
```

3. `JSON.stringify()` and `JSON.parse()`: Using these allowed me to save my dictionary to `localStorage`—at first, I was unsure how to share the data across multiple pages because I thought only strings could be saved to `localStorage`, but using `JSON`'s `stringify` and `parse` methods allowed me to convert my dictionary into a format that's readable by `JavaScript`.

```
    localStorage.setItem("wishItems", JSON.stringify(wishes));
    items = JSON.parse(localStorage.getItem("cartItems"));
```

4. `setAttribute`: This was especially useful when creating new `img` elements, because `img` tags worked differently than other text based tags and I couldn't easily edit the `id` with `innerHTML`. However, using this method allowed me to easily modify different attributes of an `id`.

```
    itemPic.setAttribute("src", findPic(i));
```

5. `appendChild()`: Using this allowed me to actually show the items created by `JavaScript` on the `HTML` page, by simply adding the new `ids` to the end of the parent's node!

```
    shown.appendChild(itemPic);
    shown.appendChild(itemDes);
    shown.appendChild(itemBut);
```