

Assignment 6B – Adding Functionality to a Website with JavaScript

Link: https://bfiksel.github.io/homework_5/index.html

Reflection

Throughout Assignment 6B, my javascript skills were tested to their limits. While at times it was frustrating, it was satisfying to be able to figure out, through trial, error, and console(log), what was working and what was not. I finally discovered the utility of the console within the browser, as an area to immediately test code functions and see what the responses were— this reminded me of the IDLE console window we used over the summer Intro to Python coding class, the immediate feedback of which I appreciated.

Some bugs I encountered, and my solution to them:

- Many times when I would try to add a new line or function in the Javascript file, it would cause none of the javascript functions to work at all. This was quite frustrating and made for some difficult debugging.
 - Slowly through trial and error, mostly removing certain parts of the javascript, re-saving, and seeing if functionality was restored in browser, I was able to identify which parts of my code were problematic
- `var.InnerText` was not working for one function, even though it worked for another function with similar content
 - As a solution, I tried `innerHTML` and it did indeed work
- Needing to have the cart number updated on Product page as well as the Cart page was tricky; it worked correctly after items were already in cart, but it was wonky on first load. The onload cart number update function caused the cart number to say “NaN”
 - Solved by having the onload function to update cart number complete only if the boolean is True for at least one item being added to the cart, so on the Product page, it will only pull the Onload function when the user goes back to the product page after adding an item to the cart and leaving the product page
- When pulling the list of items over from product page to cart page, it comes as a string
 - Was able to solve through `JSON.parse` and `JSON.stringify`
- On cart page, the fixed size of the white box could not contain a growing list of cart items
 - Gave the rectangle a min-height and set height to auto, so the container would stretch with cart items added
- I had issues with the “If ____ && ____” operator not working, causing the function not to work
 - After much trial and error, I realized that the whole if statement, including the ‘&&’s needed to be within the same parentheses

- Function for adding the cart items to the actual cart list wouldn't run because it was returning a null value for items of the list of 'items' that had not been created yet
 - I added a condition for only running the updating cart list functionality if the item in the list did not equal null or undefined, but still wasn't working.
 - Had to move that before the condition where it was checking the item at all, and then it worked

Programming Concepts

- Arrays, and arrays within arrays
 - To manage the list of items that the user had added to the cart, including the items' qualities, I utilized an array. When the user added an item to the cart, it registered as a two-item array (one for the color of pillow, and one for the filling of pillow) stored that array within another array, which represented the full list of items that the user had selected
 - Example code utilizing the double array of cartlist2:

```
function cartpageupdate(number) {
  //cart afs dd
  if ((cartlist2[number] !== undefined) && (cartlist2[number] !== null) && (cartlist2[number][0] === "afs" && (cartlist2[number][1] === "dd")))
    document.getElementById("cartitem" + (JSON.stringify(number))).src = "/images/afs-dd-cart.png";
  document.getElementById("x" + (JSON.stringify(number))).src = "/images/x.png";
}
```

- localStorage.set and localStorage.get
 - Used these frequently in order to transfer the list of cart items and the number of items in the cart between pages
 - Pretty much whenever a function pulled an item with localStorage, it would also set the item back with localStorage, so that users could navigate between pages and content would remain the same regardless of any changes
 - Example code utilizing localStorage:

```
function addtocart() {
  console.log("cart list is")
  console.log(cartlist)
  var x = document.getElementById("cartnumber");
  y=parseInt(x.innerText);
  z=y+1;
  x.innerText = z;
  console.log(z);
  window.scrollTo({ top: 0, behavior: 'smooth' });
  localStorage.setItem("cartnum", z);
  console.log("cart number is");
  console.log(localStorage.getItem("cartnum"));
}
```

- onload and onclick in HTML triggering javascript functions
 - Used these throughout, to easily link HTML file to javascript functions
 - Onload was especially useful for populating items that had been stored with localStorage
 - Example code utilizing onclick (Add to Cart button):

```
<div class="buttoncontainer" style="position:absolute; top:920px; left:684px; width:157px; height:60px;">
  <div class="overlay" onclick="pillowtocart2(); addtocart(); cartupdate();" />
</div>
```

- Creating “var” variables for shortcuts to document.getElementById items
 - Used variables for many aspects of the javascript functions, especially as a quick reference for content which I needed to use document.getElementById to refer to
 - An example that utilized this was for updating the cart number on each page in the top right:

```
function addtocart() {
  //console.log("cart list is")
  //console.log(cartlist)
  var x = document.getElementById("cartnumber");
  //console.log(x.innerHTML);
}
```

- If...and statements for conditional functionality
 - Many functions I created checked to see if a condition was true before running. In many cases, I also had to add and “and” functionality to the if statement, which I learned was done using “&&”
 - An example was checking *if* an item in the cartlist array was *not* undefined/null, and if it was indeed a certain color and certain filling:

```
function cartpageupdate(number) {
  //cart afs dd
  if ((cartlist2[number] !== undefined) && (cartlist2[number] !== null) && (cartlist2[number][0] === "afs") && (cartlist2[number][1] === "dd")) {
    document.getElementById("cartitem" + (JSON.stringify(number))).src = "./images/afs-dd-cart.png";
    document.getElementById("x" + (JSON.stringify(number))).src = "./images/x.png";
  }
  //console.log(x.innerHTML);
}
```