

Homework 6B: Reflection

Reflection

One of the largest bugs I faced during this assignment was obtaining the correct type (i.e. string vs integer, array vs string, etc.). Many of these bugs were a consequence of using local storage, which I quickly learned only stores strings. When I first had to build the cart quantity feature— i.e. the total quantity of items in the cart being displayed above the cart icon in the navigation bar, the number displayed wouldn't render properly. Instead of the quantity increasing each time a new item was added to the cart, the quantity would be appended as a string. For example, if 2 items were in the cart to start and you added 6, instead of changing the quantity display to $2+6=8$, it would show 26. After some investigation, I realized that the numbers were being added to a local storage variable, which turned them into strings. Thus, rather than the computer calculating $2+6$, it was calculating $"2" + "6" = "26"$ by 'string addition' standards. I resolved this issue by using `parseInt()` and totaling the new quantities before adding them to local storage. Then, whenever I would get the local storage quantity, I would `parseInt` as well.

Another issue I had was with updating the local storage cart array whenever local changes were made to quantity or glaze after adding to that cart. Since these changes were made locally on the cart page, they wouldn't persist if someone refreshed the page or left the page and came back. Thus, I had to find a way to update their quantity and glaze in the local storage array. This was originally a difficult task as I couldn't figure out the logic I needed to obtain the index of the affected bun bun item array within my larger local storage all items array. I learned that I needed to match the index in the order my individual item divs were laid out in the "Your Orders" cart box to the array with the same index in the larger local storage cart array. To resolve the issue, I used a for loop to iterate through each "item" class div in the "items-wrapper" container div, and break the for loop when the index of the item we were currently on was exactly equal to the current node (i.e. item div) that had a quantity or glaze change. Then, I used the found index to edit the item array at that index in the larger local storage array to reflect the change made to glaze or quantity. This worked well and I am glad the solution wasn't too difficult in terms of code fixes.

Overall, this assignment presented me with a few challenges, with the most noteworthy described above. I faced problems not only with code but also with the logic behind certain functionality. Nonetheless, this homework helped me grow as a UX developer and I look forward to applying what I have learned from this assignment to my future code endeavors.

Programming Concepts

1. I learned how to remove entire div elements from a document
 - a. You can get the current element selected that needs to be removed and do `“.remove()”`
 - b. Shown in https://github.com/glaborwit/glaborwit.github.io/blob/master/PUI2020/homework_6/js/cart.js on `deleteItem(curr)` function
 - c. Demonstrated on webpage http://gabriellelaborwit.com/PUI2020/homework_6/cart.html when items are in cart. You can press the “remove” button
2. I learned how to use `localStorage` by doing `getItem` and `setItem()`
 - a. This is demonstrated all over the website when items are in the cart because the items persist across different pages and are shown in the “Your Orders” box on the cart page http://gabriellelaborwit.com/PUI2020/homework_6/
 - b. https://github.com/glaborwit/glaborwit.github.io/blob/master/PUI2020/homework_6/js/cart.js: examples of this concept used are throughout this file
3. I learned how to `JSON.stringify()` and `JSON.parse()` to successfully add arrays to local storage and pull them out
 - a. https://github.com/glaborwit/glaborwit.github.io/blob/master/PUI2020/homework_6/js/cart.js examples of code using `JSON.stringify()` and `JSON.parse()` used throughout this file and other JS files
 - b. Seen in cart file (http://gabriellelaborwit.com/PUI2020/homework_6/cart.html) behind the scenes to set items in local storage cart and above the cart icon
4. I learned how to use `() =>` for onclick events
 - a. Used in https://github.com/glaborwit/glaborwit.github.io/blob/master/PUI2020/homework_6/js/cart.js file in `populateOrdersTab()` function towards bottom
 - b. Used to implement remove item functionality on the cart page which can be seen on web page http://gabriellelaborwit.com/PUI2020/homework_6/cart.html when items in cart
5. I learned how to use `.parentElement` to get from an element clicked to the entire div to edit something in that bigger div
 - a. Used in function `getCurrProductName(curr)` in https://github.com/glaborwit/glaborwit.github.io/blob/master/PUI2020/homework_6/js/cart.js

[ork_6/js/cart.js](#) to get a products name from the inner HTML when clicked on another part of inner div

- b. Behind the scenes work, but results in cart page working
(http://gabriellelaborwit.com/PUI2020/homework_6/cart.html)