Reflection

I found storage overly difficult and confusing. I was having a great time coding functionality on one page, but when it came to saving across pages - I needed help every time. I thought it was difficult to understand the timing of localStorage, and when the elements needed to be initialized and recalled, and what made sense when. I also thought that debugging was a bit stressful when it came to storage. I think that because I didn't quite understand it conceptually, I was debugging in circles and coming back to the same console.log solutions/syntax when often it was something high level that I just got completely incorrect. I like to do assignments somewhat last minute to give me the motivation and time pressure to do it well, but I was caught completely off guard by how difficult understanding storage is. In the future, I will start the assignment a lot earlier to account for any conceptual misconstructions that I have. I will also make sure that I understand concepts like storage on a deeper level than just simply knowing their elemental attributes (localStorage.getItem, etc.).

A lot of the smaller issues that I had revolved around the alignment of the elements, and trying to make them standard across the page, and getting the CSS to look exactly how I intended it to. There was a lot more troubleshooting in that space than I had accounted for.

Programming Concepts
1. Local Storage
   a. In order to keep the cart item notification icon consistent across all of the pages, local Storage has to be utilized. So I set the localStorage to set the item after I added a notification.
2. Looping through Elements
   a. In Javascript, I wasn't aware of just how much important it is to consistently loop through items. However, in this assignment, I continually needed to loop through different buttons or different rows to update what I needed to.
3. Calling Functions
   a. There were a lot of instances in which I needed to call a function inside another function. For example, updateCartNoti() is called inside of function(ready) because the function should be implemented right when the page is loaded. It turned out to be a huge asset for me.
4. innerText
   a. When trying to figure out the cart, I realized how important it was to get the text inside elements, so that they can be used for other functions and to give the user information.
   b. Example: document.getElementsByClassName("dot")[0].innerText = noti;
5. The if and else
   a. This was really useful if I wanted to cover an edge case in my website.
   b. Example: if (titles == undefined || titles == null){
         titles = [];}