Irene Lin (igl)
Assignment 6B

Reflection (3 pts total)
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.

Issue 1: Incompatible types
Throughout my project I ran into similar issues revolving around incompatible types. When getting the product price from my product details page and trying to perform a mathematical operation (+=) on the obtained innerText the result was NaN (not a number)
My solution was to first google what NaN meant, then find the correct to int method. I found in JS the method is parseInt().
Moving forward I made sure to use typeof in my debugging statements (ie. console.log(typeof X)). This helped me identify incompatible types and fix my code.

Issue 2: Appending child to correct parent
When populating my shopping cart page I decided to create new elements and append each new item to my page. I ran into some issues with appending to the wrong div structure which would then result in four different colors being shown in one product's row rather than in their respective product rows. The issue was getting the right parent element in the first place.
My solution was to provide each new element with a unique ID that corresponded to their index (see populateShoppingCart(), used i value from for loop).

Programming Concepts (5 points)
Demonstrate 5 programming concepts that you learned in Javascript and used in this assignment with an example.

1. Although JS does not require strict typing it is still highly necessary to keep track of what the types of returned items are when trying to perform functions on top of them. An example is the use of parseInt() in addProductPrice().
2. Functions should have distinct purposes and ideally serve only one particular task. An example is the separation of all my adding new element helper functions from populateShoppingCart().
3. Functions should be loosely coupled. An example of this is how calcSubtotal() and populateShoppingCart(). In the future if the store has a store wide discount, it can be easily applied to calcSubtotal() without affecting how any other function works.
4. The creation of objects can help with the grouping and organization of new data. On my website I made use of this to store product details in a shoppingCartItem() object which I later populate and push to local storage.

5. The difference between global variables and variables within functions. Global variables can be accessible by any function and persists after function ends while local variables (defined in function) are constrained to the scope of the function. I use a global int variable for cartItemCount and cartSubtotal which I reference in multiple functions such as populateShoppingCart() and updateSubtotal() and updateCartCount().