

Implementation Challenges or Bugs

- **Accessing desired elements in the DOM file**
 - While I had expected this to be the easiest part of implementing websites with JavaScript, I encountered several issues when trying to get DOM elements. 1) I did not know that `getElementById()` is only a method on the document object, so I ran into bugs when I tried to call it on other types of objects. In order to resolve this issue and correct my misunderstanding, I had to use Chrome's developer tools and perform a few hypothesis testings using `console.log()`. 2) Since it was the first time for me to use JavaScript to make changes to an HTML template(for a shopping cart item), I had to think hard about how to get the grand-parent of a child element in a complicated tree structure. Again, I managed to resolve this issue by inspecting the code of my website and doing a few testings with the `parentNode` property through `console.log()`.
- **Determining when to read from and write to local storage**
 - My usual coding habit is to write down the pseudocode or basic algorithm before starting to write the actual code. However, when I first started writing the JavaScript parts, I had trouble imagining when to access the local storage. I approached this challenge by thinking from a higher or more abstract level. I first separated my JavaScript code into three different documents: one for the product detail page, one for the shopping cart page, and one for updating the shopping cart icon on every page. From there, I had a clearer overview of my work and started to implement each web page without using the local storage. Finally, once I had the basic functions for each web page implemented, it became easy for me to check my website and determine when and where to insert code for accessing and updating the local storage.

Coding Concepts

1. **JavaScript Objects** are containers that store properties (key-value pairs). They are useful for keeping my code structured and avoiding unnecessary repetition.
 - a. Example: I specifically used JavaScript objects to store properties of a shopping cart item, so every time a user clicks on "Add to Cart," a new "Product" object is created with the user's selected properties.
2. **Local Storage / Session Storage** is a web storage object that allows key-value pairs to be saved in the browser.

- a. Example: This concept is essential to my shopping website since it allows data to be stored and transferred across multiple web pages. Without using `localStorage/sessionStorage`, I cannot keep the number of shopping cart items consistent when users navigate through the different web pages. My shopping cart page has two states (empty shopping cart vs. filled shopping cart with at least one item). I had to access and check the local storage every time the page loads in order to determine which state of the shopping cart to display to users.
- 3. **HTML “Template”** is useful for displaying and modifying repeated HTML elements.
 - a. Example: Although I used the `<div>` tag instead of the `<template>` tag, I relied on this same concept in order to format and add new shopping cart items to my shopping cart page. So in my HTML file, I first created an item template with the desired tree structure and then used CSS to hide this initial empty template. When a user clicks on “Add to Cart,” I use `item_template.cloneNode(true)` to copy the item template and pass in the appropriate HTML properties according to user selection.
- 4. **Window.onload** allows a function to be called as soon as the webpage has been loaded.
 - a. Example: Using `window.onload` and `window.addEventListener('load', function(){})`, I was able to perform tasks as soon as the page finished loading and update the shopping cart quantity and state according to the most recently updated local storage.
- 5. **ParentNode** is a property that returns the parent node of the specified node, as a Node object.
 - a. Example: I specifically used the `parentNode` property in my `removeCartItem()` and `addItemQty()` functions to get the entire item element in which the remove button or spin button belongs to.

Citation For Product Images

<https://unsplash.com/>

<https://www.etsy.com/listing/925132524/harness-and-lead-set-waterproof-biothane?gpla=1&gao=1&variation0=1745621729>