Tamara Zilovic

PUI Fall 2021 / **Assignment 6b –**
Adding Functionality to a Website with JavaScript

REFLECTION + CONCEPTS

REFLECTION

Coding Bug # 1 - Adding & Styling Drop-Down Menus

Within my product detail page, I employed drop-down menus that would allow the user to select the glazing option and roll quantity, which would subsequently update the displayed price. I spent quite a bit of time styling these drop-down menus; while they were not extremely important to the functionality, I was very insistent on not having a gradient effect within the menu. Even when I tried to make the menus transparent, there would still be some sort of inner gradient fill. After searching for an answer to this issue for a while, I came across a CSS example on the website codepen.io that included a background image - with formatting - that ended up resolving this issue! I did modify the code a bit, but the reference can be found here:
codepen.io https://codepen.io/vkjgr/pen/VYMeXp

Coding Bug # 2 - Adding a pop-up circle/number to the shopping cart

Upon adding an item to the cart, I wanted a bright pink circle - with quantity of items currently in cart - to appear on top of the shopping cart icon. To do this, I was initially encountering bugs with how to add this - when does it appear, and how? How to get this to appear **on top** of the actual cart icon? In order to add this, I basically made a "div" element with the id "cartpop"; this gets activated when the **add to cart** button is clicked, through the function "cartamt()", which switches the **display** style from 'none' to 'block'. Using Javascript to actually change the overall display of an element (whether it's displayed at all or not) was well-explained by this W3 schools page:
https://www.w3schools.com/jsref/prop_style_display.asp

Coding Bug # 3 - Having one action prompt multiple functions

When adding an item to the cart, I wanted both the cart amount to update, as well as an alert to appear, acting as immediate feedback that that item had been added to the cart. So, I wanted two different functions - "cartamt()" and "itemalert()" - to be activated when the add to cart button was clicked. This might seem pretty simple, but I initially really wasn't sure how to prompt two different functions with one single action. This page from "GeeksforGeeks" was super helpful, as I realized that you could write onClick="function1(); function2()" :
https://www.geeksforgeeks.org/call-multiple-javascript-functions-in-onclick-event/

Coding Bug # 4 - Assigning values within <select> tags

I used <select> tags to create my drop-down menus for choosing glazing option and quantity. When these were selected by the user (onChange), I wanted the price to update as well. Choosing a glazing quantity would slightly increase the price of a single cinnamon bun, while changing the quantity would also, of course, change the total price for that selection. W3 schools had a really great page about using select tags, and actually assigning values to the options in your menu. I used this to embed price values in the different options, which would then update the price using Javascript. Page from W3 schools:
https://www.w3schools.com/tags/tag_select.asp

Coding Bug # 5 - Converting option value to a number

I initially was getting some bugs when trying to use the option value(s) within my <select> tag to calculate total new price with my Javascript function. I didn't realize that Javascript does not automatically view integers as such, and had to add **Number**(cartamount) to get this to read as a number (and be able to use the numbers in equations). This W3 Schools page about The Number() Method was helpful:
https://www.w3schools.com/js/js_number_methods.asp

PROGRAMMING CONCEPTS

CONCEPT #1 - Changing CSS styles in Javascript

As mentioned in my ‘Coding Bug #2’, I didn’t realize that Javascript could actually alter particular CSS styles of an id element or class of elements. This was pretty awesome actually, as I used this technique in displaying the circle on the cart that would pop up (and update the interior number) when items were added to the cart. I did this by using a function to change the display style; in particular, I used the cartamt() function to change the display of the pink circle pop up (I literally called it pinkCircle) by writing the following code:

```
pinkCircle.style.display=”block”;
```

(previously in CSS the display style was “none”).

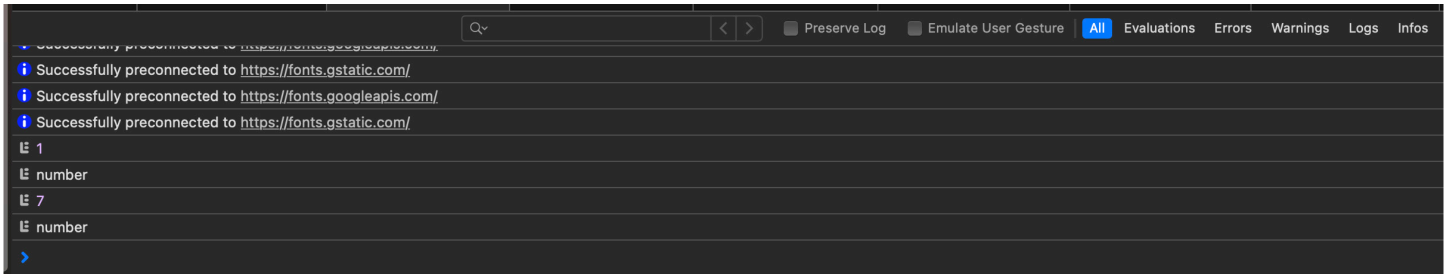
CONCEPT #2 - Replacing text/quantities using Javascript .replace

Within my cartamt() function, I ended up making a function that would constantly add to whatever quantity was already there (initialized at 0). To make this new number replace whatever number was written in the middle, I used the .replace functionality, which would then replace the ‘baseline’ amount in cart to the new cart amount, so that I didn’t really need to make any sort of a loop, or even use local storage initially. This site had a good explanation of this: https://www.javascripttutorial.net/javascript-string-replace/

CONCEPT #3 - Employing console.log(typeof ____)

Though I didn’t really use much personally from this tutorial, I found the concept of checking on your Javascript code through console.log - specifically, console.log(typeof ____) - to be extremely beneficial in going through the process of embedding the shopping cart total to the local storage. I kept getting “NaN” errors for a while, so being able to check the type of the variable through this was great, and let me know when I had finally converted the variable to a number.
Tutorial:
https://www.youtube.com/watch?v=PoTGs38DR9E

Example in my workflow:



```
Successfully preconnected to https://fonts.googleapis.com/
Successfully preconnected to https://fonts.gstatic.com/
Successfully preconnected to https://fonts.googleapis.com/
Successfully preconnected to https://fonts.gstatic.com/
1
number
7
number
>
```

CONCEPT #4 - Using Local Storage

Local storage was mentioned in class as a way to store some (not sensitive) data on the client-side. I tried to use this concept since I used various HTML files for my different product pages, so this was a way to keep the amount of items added to the cart consistent across pages, and so that the data wouldn’t disappear when pages were reloaded.

CONCEPT #5 - Getting value/text of selected option in Javascript drop-down
Something I had a lot of challenges with was getting the inner text of an option within a <select> tag from one of the drop-down menus I made (for quantity and glazing). I didn't realize you could actually specify this by employing square brackets, but this article on stackoverflow was really helpful:
https://stackoverflow.com/questions/1085801/get-selected-value-in-dropdown-list-using-javascript

OVERALL REFLECTIONS ….
I found this assignment to be really, really challenging … I feel really frustrated that I could not complete the last parts of having the details of the items I added to the carts page, and the ability to remove them. But, I am pretty happy that I managed to figure out how to use local storage across my pages, and also had this number appear/update on the cart page.

I am hopeful I'll be able to receive at least partial points on the actual coding aspect of the assignment? I tried to really add a lot to my reflections page to make up for this.

Please Note: I also read through the book ***HTML&CSS design and build websites*** by Jon Duckett, which was very informative.

All images are from Unplash - Unsplash license and were freely downloaded

GITHUB SITE: https://tzilovic.github.io/homework_6b/