Thuy-Vy Nguyen
HW 6B

Links
Live site: https://tvnn1231.github.io/PUI/Homework_6B/
Github repo: https://github.com/tvnn1231/PUI/tree/master/Homework_6B

Reflection

One of the biggest issues I continually encountered throughout this assignment was when I was working with numbers. Figuring out how to update the number in the cart based on how many items were added, as well as how to update the total prices in the cart dropdown and on the cart page, was difficult mostly because I would be working with the numbers as if they were numbers when they were actually strings. I would often get "NaN" errors if I tried to multiply a number (that was actually a string) by another number, or I would see "01" because the code would be concatenating "0" and "1" instead of adding 0+1. I worked through these issues by using console.log and the typeof operator to understand at which point in my code a number turned into a string. To resolve this issue, I had to look up how to turn a string into a number, and found that I could use parseInt() to turn them back into numbers.

Once I turned them into numbers, though, I would have issues with the numbers rounding or not showing enough decimal places. For example, 2.75 would round to 3, so the math for calculating the total price would end up wrong and I couldn't figure out why. When I eventually realized what was going on, I looked up how to make numbers have a certain number of decimal places and discovered that I could use the toFixed() method to round the number to a certain number of decimal places, which then fixed the issue. Now that I'm aware of these potential issues when working with numbers, in the future, I will know to be cognizant of when numbers are turning to strings (or vice versa) and to specify the number of decimal places I want to include.

I also ran into some styling issues. When I was adding elements to the cart dropdown, they were appending side-by-side instead of adding below, even though the container was a flex container. After looking through my code and playing with it for a while, I realized the flex was set on the parent div, but not on the div that actually contained the image and details that were being appended. When I corrected that, though, they were still not appending how I wanted them to. After researching a bit online, I learned I should set "flex-wrap: wrap" on that container, which fixed the issue.

Another bug I encountered was when I was trying to append text to a div, it wasn't appending unless I already had text in it. I didn't understand why that would be the case, so I asked around to see if the append method had a rule I didn't know about, and no one seemed to understand why it wouldn't append to a div without placeholder text. After looking over my code some more and asking others to look at it as well, I found that the issue was that the new elements I was creating and appending had the same IDs, which were also the same as those in the existing <p> tags that were in the div I was trying to append to. Since I was referencing elements by ID in my script, and since an ID should only be associated with one element, this was breaking the code. Once I completely emptied out the div so it had no existing elements in it (since I didn't need them in there anyways, as those were the elements I was appending) and changed the JavaScript so that the new elements I was creating did not have IDs, the code worked correctly, and I was able to append the new elements to the div. In the future, I will be sure to be aware of any time I may be using the same ID on multiple elements, especially being cognizant of it when I create new elements, and how doing so may be what's causing issues with my code.

Programming Concepts

As I was using JavaScript to render the product detail page, I learned and applied these five JavaScript concepts: objects and object constructors, arrays, local storage, JSON, and navigating multiple data types. I had initially created a separate HTML file for each product detail page, but for this assignment, I wanted to have one HTML file that set up a generic product detail page that would change and render according to which item was clicked on the product browsing page. In doing so, I applied my new knowledge each of these concepts.

I used an **object constructor** to easily create an **object** for each product. This allowed me to set the ID, name, price, images, and description of each product, which would then be used to render that specific product detail page. I could set the attributes or innerHTML of the corresponding HTML elements on the generic product detail page using the object properties of the specific product I wanted.

I saved each of those objects in **local storage**, which allows me to store data in the browser so that I can access it whenever I need to. As part of this concept, I learned how to put an item into local storage and retrieve it. By saving the objects that represent the products into local storage, I could dynamically render the product detail pages by retrieving them whenever I needed. I also used local storage to save items that were added to the cart and then retrieved that data to show those items on the separate cart page.

As part of learning about local storage, I also had to learn the concept of using **JSON** to easily store and retrieve object data. I learned how to format the data in JSON and use methods to use it with local storage. For example, because local storage can only save strings, I used JSON.stringify to convert the objects that represented the various products into strings first. Then, when I wanted to retrieve those items, I used JSON.parse to convert the JSON string back into a JavaScript object. Learning JSON gave me a better understanding of how storing and retrieving data works, and made it easier for me to understand and use local storage.

I also learned about **arrays**. I created an array that contained the objects of the products that were added to the cart, and saved that array as an item in local storage. I learned how to push objects to the array whenever a product was added to the cart, and then to access those objects, I learned how to index through the array. I also learned how to remove items from an array using the splice method. When I wanted to remove an item from the cart, I used splice to remove that object from the array, and then I could reset the local storage to contain the updated array without the item I had removed, effectively using array methods to delete an object from local storage.

Lastly, for this assignment, I used many data types, including objects, arrays, numbers, and strings, and I had to learn about **how to navigate these various data types**. For example, I learned that I could use multiple data types together, such as when I created an array of objects, and learned how to navigate that nesting by, for example, first indexing into the array to reach the object and then being able to get the property of that object. I also learned how to convert numbers to strings and vice versa. To display the number of items in the cart on the cart icon and to calculate the total price based on the prices of the products and the quantity, I had to convert from string to number and back multiple times.

References

Photos:

- Photo by Klara Avsenik on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Kjartan Einarsson on Unsplash - https://unsplash.com/s/photos/bakery?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Joseph Gonzalez on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Lauren Gray on Unsplash - https://unsplash.com/s/photos/cinnamon-rolls?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Fallon Michael on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Fallon Michael on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Otto Norin on Unsplash - https://unsplash.com/s/photos/bakery?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Mark Olsen on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Rob Sarmieneto on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Sarah Shaffer on Unsplash - https://unsplash.com/s/photos/cinnamon-rolls?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Image by Rudy and Peter Skitterians from Pixabay - https://pixabay.com/?utm_source=link-attribution&amp;utm_medium=referral&amp;utm_campaign=image&amp;utm_content=4719023
- Photo by Maranda Vandergriff on Unsplash - https://unsplash.com/s/photos/bakery?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText
- Photo by Karolina Wv on Unsplash - https://unsplash.com/s/photos/cinnamon-rolls?utm_source=unsplash&amp;utm_medium=referral&amp;utm_content=creditCopyText

- Photo by Nathan Dumlao on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText
- Photo by Sergio Arze on Unsplash - https://unsplash.com/s/photos/cinnamon-roll?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText
- Photo by Brina Blum on Unsplash - https://unsplash.com/s/photos/chocolate-pastry?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText

Icons from Font Awesome (license: https://fontawesome.com/license)

- Map marker - https://fontawesome.com/icons/map-marker-alt?style=solid
- Phone - https://fontawesome.com/icons/phone?style=solid
- At symbol - https://fontawesome.com/icons/at?style=solid

Icons from Flat Icon (license: https://file000.flaticon.com/downloads/license/license.pdf)

- Oven by photo3idea_studio - https://www.flaticon.com/authors/photo3idea-studio
- Shipping truck by monkik - https://www.flaticon.com/authors/monkik
- Home by Kiranshastry - https://www.flaticon.com/authors/kiranshastry