Challenges & Bugs

1. I have been struggling with understanding the concept of the localStorage. Since it's not cleared every time I open or refresh the webpage, it caused confusion during my debugging process. I always thought that the localStorage should be cleared whenever I refreshed. To examine if my localStorage was working well, I had o put "localStorage.clear()" in the code to clear it up and delete this line of code.

2. Before I implementedhe localStorage, I was using the container to store the items. After I tried to use localStorage for storing items, **the whole storing structure of the shopping cart had to be redesigned.** In my final design, the page reloads the items in localStorage and adds the codes into container after opening the shopping cart webpage.

3. I'd been struggling with the **implementation of different functions**, such as ".remove(), localStorage.key()", how to iterate my localStorage and so on. I tried to search for tutorials and finally made it.

4. Another problem I met is that **there is no container for storing in the webpages other than shopping cart webpage.** Every time I added something to the localStorage, there will be an error that there is no such a container to add the item into. To solve this, I separated my JS file into two files. On the shopping cart webpage, every time an item is added to the local storage, it should be added both to the localStorage for storing and the container for display. On the other webpages, the item is only added to the localStorage and update the counter on the badge.

Concepts

1. Modular Programming

I tried to implement modular programming while writing functions. To make the code clean and easy to debug, the functions should be written in separated modules. So that I can test and use each function easily without copying a big amount of code everywhere.

2. Functional programming

The code is working under a concept of functional programming. There will be a series of functions working after clicking a button. For example, when the "add to shopping cart" button is clicked, there will be a function getting the name, price and quantity of the item, then a function will update the counter on the badge and another function will store the item into localStorage.

```
//Function for purchasing items.
function purchaseClicked() {
  var cartItems = document.getElementsByClassName('containerCart')[0];
  while (cartItems.hasChildNodes()) {
      cartItems.removeChild(cartItems.firstChild);
  }
  for (var i = 0; i < localStorage.length; i++) {
    localStorage.removeItem(localStorage.key(i));
  }
  updateCartTotal();
  updateBadgeItem();
  alert('Thank you for your purchase');
}
```

3. Object Oriented Programming

To store an item into the localStorage, each item should be constructed as an object. In this

assignment, the constructor is “function Item(name, price, imageSrc)”. The item is stored as a string in the localStorage. It is much easier to get the corresponding properties of an item.

```
//Constructor of shopping items.
function Item(name, price, imageSrc) {
  this.name = name;
  this.price = price;
  this.imageSrc = imageSrc;
}
```

4. Adding html code in JS

To add the item into the shopping cart and display it, I need to add HTML code in JavaScript files. There is an important concept that the HTML code is editable in JS which helped a lot with making changes on the webpage.

```
var cartRowContents = `
  <br/>
  <div class="p-img">
    <img width="110" alt="${title}" src="${imageSrc}">
  </div>
  <div class="cart-item">
    <div class="productDetailCart">
      <span class="shop-item-title">${title}</span>
      <span class="shop-item-price">${price}</span>
    </div>
    <div>
      <label for="quantity">QUANTITY </label>
      <input class="quantity" type="number" name="quantity" value="1" style="width:60px">
      <button class="btn-remove" type="button">REMOVE</button>
    </div>
  </div>
  <hr>`;
cartRow.innerHTML = cartRowContents;
cartItems.append(cartRow);
```

5. LocalStorage won’t be cleared when the webpage is closed or freshed. The data in localStorage will never be out of date until the user clears it. For example, when I changed the code and refreshed the webpage, the items were still the same in the shopping cart.