

Andy Qiu

PUI Fall 2020

Github link: https://github.com/byfar1/byfar1.github.io/tree/master/homework_6B

Website link: https://byfar1.github.io/homework_6B/product_description.html

During my coding endeavors, I encountered a number of issues that were difficult to solve, but with time and patience I managed to overcome them. The first issue I had was the shopping cart remove button removing the entire div class. I had a portion of this coded out from 6A, but it was buggy to say the least. I found out that it was because my `buttonClicked.parentElement.parentElement.parentElement.remove()` had one too many `.parentElements`, so it ended up removing the entire parent DIV, thus deleting my entire page. This wasn't exactly a challenging error to fix, but it was more of an importance of stressing the details--any small coding error will multiplicatively cause issues in the future.

```
function setItems(product) {
  console.log(product);
  let cartItems = localStorage.getItem('productsInCart');
  cartItems = JSON.parse(cartItems);

  if (cartItems != null) {
    cartItems[product.tag].inCart += 1
  } else {
    product.inCart = 1;
    cartItems = {
      [product.tag]: product
    }
  }
  localStorage.setItem("productsInCart", JSON.stringify(cartItems));
}

function addToCartClicked(event) {
  var title = document.getElementsByClassName('item-name')[0].innerText;
  var price = document.getElementsByClassName('item-price')[0].innerText;
  var color = document.getElementsByClassName('item-color')[0].innerText;
  var size = document.getElementsByClassName('item-size')[0].innerText;
  var image = document.getElementsByClassName('shop-item-image')[0].src;
  localStorage.setItem("name", title)
  localStorage.setItem("price", price)
  localStorage.setItem("img", image)
  localStorage.setItem("color", color)
  localStorage.setItem("size", size)
  // addItemToCart(title, price, image)
}
```

Another really challenging issue that I encountered was trying to make cross-html changes through JavaScript. I discovered a solution through the utilization of `localStorage`. When the 'add to cart' button was clicked, it would store the selected details onto the `localStorage`, which I can then call from the shopping cart page and apply to the cart details.

There were also 5 specific programming concepts that I learned as a part of the assignment, and found to be extremely helpful to know.

```
function onLoadCartNumbers() {  
  let productNumbers = localStorage.getItem('cartNumbers');  
  if (productNumbers) {  
    document.querySelector('.add-cart span').textContent = '(' + productNumbers + ')';  
    document.querySelector('.navbar span').textContent = '(' + productNumbers + ')';  
  }  
}
```

- 1) document.querySelector was a very useful substitute for .getElementById as it allowed for finer selections within the class

```
function setItems(product) {  
  console.log(product);  
  let cartItems = localStorage.getItem('productsInCart');  
  cartItems = JSON.parse(cartItems);  
  
  if (cartItems !== null) {  
    cartItems[product.tag].inCart += 1  
  } else {  
    product.inCart = 1;  
    cartItems = {  
      [product.tag]: product  
    }  
  }  
  localStorage.setItem("productsInCart", JSON.stringify(cartItems));  
}
```

- 2) I learned that you can immediately access the dictionary by using `[product.tag]`. It really helped speed up the process in my coding

```
let products = [
  {
    name: 'Cat Backpack',
    id: 'catbackpack',
    tag: 'catbackpack',
    color: 'crazyberry',
    size: 'tiny',
    img: 'img/cbcrazyberry.jpg',
    price: 25,
    inCart: 0
  }
]
```

3) **Dictionary:** Using dictionaries allows me to create a more organized and detailed categorization of an item. It also allows for easy calling and editing of the details.

4) **List Selection:** I learned how to select an item in a list using an index number

```
var ClearCartButton = document.getElementsByClassName('shop-item-button2');
for (var i = 0; i < ClearCartButton.length; i++) {
  var button = ClearCartButton[i];
  button.addEventListener('click', clearCart);
}
```

5) **Event Listeners:** Learning how to utilize event listeners was crucial in creating interactive interfaces. I learned how to use event listeners to react to button clicks and subsequently make changes in html.