

Deliverables

Cart Page direct link: <https://danafrostig.github.io/assign6/6B/checkout2.html?>

Github repository:

<https://github.com/danafrostig/danafrostig.github.io/tree/master/assign6/6B>

Product details page with JavaScript functionality:

<https://danafrostig.github.io/assign6/6B/original.html>

or

<https://danafrostig.github.io/assign6/6B/blackberry.html>

Reflection:

One of the biggest challenges for me for this assignment was simply getting started. I have used a little bit of HTML and CSS many years ago, but I had zero experience coding using JavaScript. I started by reading some Mozilla Developer Network (MDN) documentation on JavaScript and tried to practice using some of the examples from there. However, each time I tried to apply it the assignment, I still didn't know where to begin. I ended up writing out a "to-do" list, in plain English, of the steps I need to take in order to implement the JavaScript functionality within my Bun Bun Bake Shop website. I then added more steps to my to-do list, as guides for functions that will help me create the main tasks of my to-do list. This made it a lot easier to figure out what kinds of functions I needed to create for my website to implement properly. Although I had to re-order or change some of my functions later on, having the to-do list as a guide was very helpful.

Another challenge for me was creating a div element with JavaScript to display my cart items (stored in local storage) on the cart webpage. I first tried to use the HTML I hard-coded when working on assignment 6A and cloning that div with JavaScript while updating the values with the values stored in local storage, however, I kept running into issues where id's were being duplicated (which I know is an incorrect usage of id's). After googling around, I realized I could use the .innerHTML property, which I had already been using to dynamically change text using JavaScript, to create an entire div with child HTML elements as one long string. I also changed all the id's in the div to classes, so that each order could use the formatting of the div on the cart page without creating any errors due to reusing id's.

Overall, I feel as though this assignment was a good exercise that allowed me to explore many different JavaScript concepts that were unfamiliar to me.

Programming concepts learned through this assignment:

Through this assignment, I learned how to code using JavaScript, a programming language I was completely unfamiliar with. While listing all of the concepts I learned would basically look like the MDN website, I've included several concepts that were

especially interesting for me to learn. Whenever I feel stumped while programming, I feel that much more accomplished when I'm able to figure it out in the end, so the following concepts are ones I'm proud to say I learned how to implement.

- `.innerHTML`

Probably the property I used the most for this project was `.innerHTML`.

`.innerHTML` allowed me to change the text and HTML that was coded for my webpages and use JavaScript to update that text dynamically on the webpage. I also learned I include an entire block of HTML code as a string using `.innerHTML`, to add entire HTML divs with children elements (which is what I used to display my cart object on the cart page).

- Example:

Here is how I used `.innerHTML` to display a block of HTML code using JavaScript. Every element has a class, that I can update it depending on the stored values from local storage.

```
//function for creating styling for displaying orders to the cart page
function makeDivForOrder(order, i){
  ...var cartDiv = document.createElement('div');
  ...cartDiv.innerHTML = ''
  ...+ '<br><p class="infoTitle">original cinnamon roll</p>'
  ...+ '<p class="info">rolls: <b><span class="rlls">12</b><br>'
  ...+ '<glaze: <b><span class="glz">vanilla-milk</span></b><br>'
  ...+ '<b><span class="bxs">2</span></b><br>'
  ...+ '<p>price: <b><span class="prc">48</span></b></p>'
  ...+ '<p class="remove"><button onclick="removeOrderFromCart(' + i + ')">remove</button></p><br><br>';
  ...var flavorTitle = cartDiv.querySelector('infoTitle');
}
```

- Local Storage

Being able to use local storage was crucial for creating the cart webpage for my Bun Bun website. Without local storage, there was no way I could record the user input from the product details page with the details of the order and display that information on a different page.

- Example:

Here is an example of how I used local storage to store each order added to the cart. I learned that local storage stores data in strings, so I had to use JSON to parse data from local storage, or to stringify any data I was putting into storage.

```
//function for adding each order to local storage
function appendOrderToStorageList(){
  ...if (!localStorage.getItem("orderList")){
  ...var orders = [];
  ...localStorage.setItem("orderList", JSON.stringify(orders));
  ...}

  ...var orders = JSON.parse(localStorage.getItem("orderList"));
  ...orders.push(createOrder());
  ...console.log(orders)
  ...localStorage.setItem("orderList", JSON.stringify(orders));
};
```

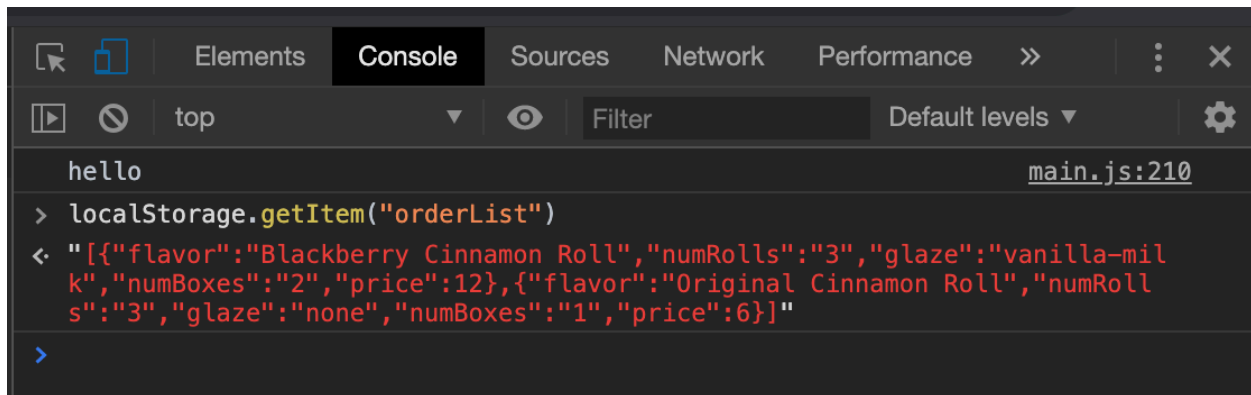
- Using Inspect

Element and Console in the browser

Before I had coded the functions for displaying the cart items on the cart webpage, I first coded the functions for putting orders into local storage. To make sure those functions were working properly, I used the console in the browser often to check the contents of the array I had created to store the objects/orders.

- Example:

Here is an example of how I would check the contents of my local storage



- querySelector()

I used the querySelector() method to update the values in each div created on the cart webpage. querySelector() allowed me to change the inner HTML of an element by calling the that element's class.

- Example:

```
...if (order.flavor=="Blackberry Cinnamon Roll"){
...    var cartImg = cartDiv.querySelector('.cartimg');
...    cartImg.src = './imgs/blackberry_roll.png'
...};

...var rollsSpan = cartDiv.querySelector('.rolls');
...rollsSpan.innerHTML = order.numRolls;
...var glazeSpan = cartDiv.querySelector('.glz');
...glazeSpan.innerHTML = order.glaze;
...var boxesSpan = cartDiv.querySelector('.bxs');
...boxesSpan.innerHTML = order.numBoxes;
...var priceSpan = cartDiv.querySelector('.prc');
...priceSpan.innerHTML = order.price;

...return cartDiv;

};
```

- .splice(index, number)

I used the .splice() method to remove an object (or bun order) from my array of objects (bun orders) for my cart webpage. This function was called when the user clicked on the "remove" button under the order they wanted to remove from their cart.

- Example:

Here is how I used the `.splice()` method to remove the cart item. I passed in the index of the item and indicated that I only wanted one item to be removed each time the method was called.

```
//function for allowing users to remove orders from their cart
function removeOrderFromCart(i) {
  ...console.log(i);

  ...var orders = JSON.parse(localStorage.getItem("orderList"));
  ...orders.splice(i, 1);
  ...localStorage.setItem("orderList", JSON.stringify(orders));

  ...displayAllOrders();
};
```