Programming Concepts

Local Storage
Local storage is a type of HTML web storage which allows me to store data that won't be deleted when the browser is closed. There is a "key" to access the stored "value". "setItem()" is used to save data, "getItem()" is used to read data, and "removeItem()" is used to remove data. Here I'm getting the objects that are stored in the cart with local storage, using it to check whether there are items already inside.

```
function removeItem(obj) {
    var cartItemsString = localStorage.getItem("cartItems");
    if (cartItemsString !== null) {
      var cartItems = JSON.parse(cartItemsString);
```

JS Objects
JavaScript objects have properties and methods. Properties can have different values and are accessed through "propertyName". Methods are actions that objects can perform and are accessed through "methodName()". Here I'm creating a cinnamon roll object with properties such as its name, glaze, quantity, and price, so I can put this item and information into the cart.

```
var itemObject = {
  name: productName,
  glaze: glazeValue,
  quantity: quantityValue,
  price: productPrice
};
```

JS Arrays
JavaScript arrays can contain many values as well as objects. They start from an index of 0 instead of 1. Each value/object can be called using the index number. Declare arrays with square brackets and commas like […, …, …]. Here I'm creating an array for the cart to store multiple cinnamon roll objects and then display their information in the shopping cart page.

```
function addItem() {
  var cartItemsString = localStorage.getItem("cartItems");
  if (cartItemsString === null) {
    var cartItems = [];
  } else {
    cartItems = JSON.parse(cartItemsString);
  }
```

JSON

JSON (JavaScript Object Notation) is text that helps store and transport data so it's easier to parse and translate. Objects can be converted into JSON with "stringify()" and vice versa with "parse()". Here I'm using the JSON method "stringify()" to turn cinnamon roll objects into the JSON format, so that it can be stored in local storage and accessed later.

```javascript
if (ind !== -1) {
   cartItems.splice(ind, 1);
   localStorage.setItem("cartItems", JSON.stringify(cartItems));
   updatePage();
}
```

On Change

"onchange" is a HTML event attribute that executes a JavaScript when a user changes the selected option of a <select> element. Instead of using it in HTML, I used it in JavaScript to build a function that changes the item price. Depending on the value selected from the quantity and glaze drop-down menus, the price of the product is updated accordingly.

```javascript
quantity.onchange = function () {
   numOfProds = document.getElementById("quantity-number").value;
   document.getElementById("product-price").innerHTML =
     "$" + prodPrice * numOfProds;
};
```