

**PUI Assignment 6B**  
**Programming Concepts**  
**Jee Rim**  
**11.8.20**

**1. Shifting the Product Detail Image that corresponds to the action of selecting glaze options.**

```
function chooseglazing() {  
    let glazeSelect = document.getElementById("glaze").value;  
    let imgEl = document.getElementById("product-img");
```

-To enable the action of changing the product image as the user selects the glazing option, I applied the programming concept of setting the two values including glaze select options & the image options. To update the value as the user selects a different glazing option, “document.getElementById” is used.

**2. Using If & Else If Statement along with the triple equal operators**

```
    if (glazeSelect === "none") {  
        imgEl.src = 'Images/1421348929.jpg';  
        glazeChoice = "none";  
    } else if (glazeSelect === "sugar-milk") {  
        imgEl.src = 'Images/scooters_coffee_cinnamonroll.png';  
        glazeChoice = "sugar-milk";  
    } else if (glazeSelect === "vanila-milk") {  
        imgEl.src = 'Images/6898135_preview.jpg';  
        glazeChoice = "vanila-milk";  
    } else if (glazeSelect === "double-chocolate") {  
        imgEl.src = 'Images/post_5303566.jpg';  
        glazeChoice = "double-chocolate";  
    }  
}
```

-To enable the action of changing the product image that corresponds to the drop down glazing selection options, If & else if statement is used throughout looping else if function. As the user selects the “None” glazing option, the image value remains the same and the assignment operator assigns the glazeChoice to None variable. And the other selection option is operated under the loop of else if statement, such as **else if (glazeSelect === “sugar-milk”) {**. The triple equal signs are used since it is able to strictly check the both value and type of the operands in the left and the right.

### 3. Using .innerHTML property to return the total price value that corresponds to the selection of the quantities of cinnamon rolls.

```
function priceshift() {  
    let quantitySelect = document.getElementById("quantity").value;  
    let priceTagEl = document.getElementById("price-tag");  
    if (quantitySelect === "1") {  
        priceTagEl.innerHTML= "Total Price: $6";  
    }  
}
```

-To enable the price change that matches with the number of the quantity, the value “price-tag” should be used along with .innerHTML property to return the total price value. This functionality uses similar logic with the above method used in glazing options. So if the quantitySelect value & type is equal to the value & type of “1”, then the price returns through .innerHTML as “**Total Price: \$6.**” This logic is operated through else if statements when the user selects other quantity options such as 3, 6, 12 cinnamon rolls.

### 4. Removing Cart Items through using EventTarget method, addEventListener() that sets up a function that will be called whenever the specified event is delivered to target

```
function ready() {  
    var removeCartItemButtons = document.getElementsByClassName('btn-remove')  
    for (var i = 0; i < removeCartItemButtons.length; i++) {  
        var button = removeCartItemButtons[i]  
        button.addEventListener('click', removeCartItem)  
    }  
  
    var quantityInputs = document.getElementsByClassName('cart-quantity-input')  
    for (var i = 0; i < quantityInputs.length; i++) {  
        var input = quantityInputs[i]  
        input.addEventListener('change', quantityChanged)  
    }  
}
```

-To enable the action that is specified to the targeted event, I have mainly used **input.addEventListener()** function which changes the value of quantity as the user clicks on the “**remove cart button.**” Along with this function, I used **for (statement1; statement 2; statement 3) syntax** So at First, I set a variable before the loop starts (var i = 0;) then, define the condition for the loop to run, which is **i < quantityInputs.length**. As long as the variable is less than the length of the array, the loop will continue Each time the loop executes, the variable is incremented by one “**i++**” and once the variable is no longer less than the quantity input length, the condition is false, and the loop will end.

## 5. Using `parentElement` to enable the emptying cart action

```
function removeCartItem() {  
    var buttonClicked = event.target  
    buttonClicked.parentElement.parentElement.parentElement.remove()  
    updateCartTotal()  
}
```

-To enable the removing action as the button is clicked using the `EventTarget` method, I used **`parentElement.remove()`** syntax along with the **`function removeCartItem()`**. I used `parentElement` 3 times, I was able to track “cart-items” in html, which becomes the parent of all the class elements. Using `parentElement` once only tracked on the indented element including “cart-row” elements in html, which initially did not correspond to the desired action of deleting all the items in the cart. Consequently, using parent elements enabled me to properly remove all the cart items in the cart.html.