

JavaScript Shopping Cart Code Breakdown

1. Add Item to Cart Function

```
const addItemToCart = (productId, productName, quantity, price) => {  
  
  if (!productId || !productName || !quantity || !price) {  
  
    console.error('Invalid product data');  
  
    return;  
  
  }  
  
  const product = { productId, productName, quantity, price };  
  
  cart.push(product);  
  
  console.log(`Added: ${productName} (Quantity: ${quantity}, Price: ${price})`);  
  
};
```

// Output: Adds items to the cart and validates the input data.

2. Remove Item from Cart Function

```
const removeItemFromCart = (productId) => {  
  
  const initialLength = cart.length;  
  
  cart = cart.filter(item => item.productId !== productId);  
  
  if (cart.length === initialLength) {  
  
    console.error(`Product ID ${productId} not found`);  
  
  } else {  
  
    console.log(`Removed product with ID: ${productId}`);  
  
  }  
  
};
```

// Output: Removes item based on productId, handles the case where product doesn't exist.

3. Update Item Quantity Function

```
const updateItemQuantity = (productId, newQuantity) => {  
  
  let itemFound = false;  
  
  cart = cart.map(item => {  
  
    if (item.productId === productId) {  
  
      itemFound = true;  
  
      return { ...item, quantity: newQuantity };  
  
    }  
  
    return item;  
  
  });  
  
  if (!itemFound) {  
  
    console.error(`Product ID ${productId} not found`);  
  
  } else {  
  
    console.log(`Updated product with ID: ${productId}, new quantity: ${newQuantity}`);  
  
  }  
};
```

// Output: Updates the quantity of an item if it exists, logs error if not found.

4. Calculate Total Cost Function

```
const calculateTotalCost = () => {  
  
  if (cart.length === 0) {  
  
    console.warn('Cart is empty');  
  
    return 0;  
  
  }  
};
```

```
}

const total = cart.reduce((total, item) => total + (item.quantity * item.price), 0);

return total;

};
```

// Output: Calculates the total cost of all items in the cart, warns if cart is empty.

5. Display Cart Summary Function

```
const displayCartSummary = () => {

  if (cart.length === 0) {

    console.warn('Cart is empty');

    return;

  }

  console.log('Cart Summary:');

  cart.forEach(item => {

    console.log(`Product: ${item.productName}, Quantity: ${item.quantity}, Price: $$${item.price}`);

  });

  const totalCost = calculateTotalCost();

  console.log('Total Cost: $' + totalCost);

};
```

// Output: Displays the items in the cart and their total cost. Warns if the cart is empty.

Conclusions

1. **Handling Errors:** I learned how to validate inputs to avoid unexpected errors. Proper handling of

empty values and invalid product data prevents issues in the cart.

2. **Modular Code:** Breaking the cart management into separate functions makes the code easier to understand and maintain.

3. **Array Manipulation:** The use of array methods like `map()`, `filter()`, and `reduce()` simplifies operations like updating quantities, removing items, and calculating totals.

4. **Logging and Debugging:** Proper logging using `console.log()` helps to track actions and spot errors during runtime.