Claro, aquí tienes un ejemplo de cómo puedes combinar todos los scripts necesarios para el carrito de compras y el botón de alternancia de modo claro/oscuro en una sola clase JavaScript. Este enfoque encapsula la funcionalidad en una clase, lo que puede ayudar a mantener el código organizado y modular.

### Código Completo

#### HTML

```html

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Carrito de Compras y Modo Oscuro</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div id="cart">

<h2>Carrito de Compras</h2>

<ul id="cart-items"></ul>

<p>Total: <span id="total-price">0</span> €</p>

<button id="checkout">Finalizar Compra</button>

</div>

<button class="add-to-cart" data-name="Producto 1" data-price="10">Agregar Producto 1</button>

<button class="add-to-cart" data-name="Producto 2" data-price="15">Agregar Producto 2</button>

<button id="dark-mode-toggle">Cambiar a Modo Oscuro</button>

<script src="script.js"></script>

</body>

</html>

```

#### CSS (styles.css)

```css

#cart {

border: 1px solid #ccc;

padding: 10px;

width: 300px;

}

.dark-mode {

background-color: #333;

color: white;

}

```

#### JavaScript (script.js)

```javascript

class ShoppingCart {

constructor() {

this.cart = [];

this.totalPrice = 0;

this.init();

}

init() {

this.bindEvents();

this.loadDarkModePreference();

}

bindEvents() {

document.querySelectorAll('.add-to-cart').forEach(button => {

button.addEventListener('click', () => {

const name = button.getAttribute('data-name');

const price = parseFloat(button.getAttribute('data-price'));

this.addToCart(name, price);

});

});

document.getElementById('checkout').addEventListener('click', () => {

alert('Compra finalizada');

});

document.getElementById('dark-mode-toggle').addEventListener('click', () => {

this.toggleDarkMode();

});

}

addToCart(name, price) {

this.cart.push({ name, price });

this.totalPrice += price;

this.updateCart();

}

updateCart() {

const cartItems = document.getElementById('cart-items');

cartItems.innerHTML = '';

this.cart.forEach(item => {

const li = document.createElement('li');

li.textContent = `${item.name} - ${item.price} €`;

cartItems.appendChild(li);

});

document.getElementById('total-price').textContent = this.totalPrice.toFixed(2);

}

toggleDarkMode() {

document.body.classList.toggle('dark-mode');

const isDarkMode = document.body.classList.contains('dark-mode');

localStorage.setItem('dark-mode', isDarkMode);

}

loadDarkModePreference() {

const isDarkMode = localStorage.getItem('dark-mode') === 'true';

if (isDarkMode) {

document.body.classList.add('dark-mode');

}

}

// Inicializar el carrito de compras

const shoppingCart = new ShoppingCart();

```

### Explicación del Código

1. \*\*Clase `ShoppingCart`\*\*:

- Esta clase encapsula toda la funcionalidad relacionada con el carrito de compras y el modo oscuro.

- Tiene un constructor que inicializa el carrito y llama a los métodos necesarios para configurar los eventos y cargar la preferencia del modo oscuro.

2. \*\*Métodos\*\*:

- `init()`: Configura los eventos y carga la preferencia del modo oscuro.

- `bindEvents()`: Asocia los eventos de clic a los botones de agregar al carrito, finalizar compra y alternar el modo oscuro.

- `addToCart(name, price)`: Agrega un producto al carrito y actualiza el total.

- `updateCart()`: Actualiza la visualización del carrito en la interfaz.

- `toggleDarkMode()`: Alterna la clase de modo oscuro y guarda la preferencia en `localStorage`.

- `loadDarkModePreference()`: Carga la preferencia del modo oscuro al cargar la página.

##

# Cómo Usar

1. Crea un archivo HTML y copia el código HTML proporcionado.

2. Crea un archivo CSS llamado `styles.css` y copia el código CSS

<div id="cart">

<h2>Carrito de Compras</h2>

<ul id="cart-items"></ul>

<p>Total: <span id="total-price">0</span> €</p>

<button id="checkout">Finalizar Compra</button>

</div>

<button class="add-to-cart" data-name="Producto 1" data-price="10">Agregar Producto 1</button>

<button class="add-to-cart" data-name="Producto 2" data-price="15">Agregar Producto 2</button>

#cart {

border: 1px solid #ccc;

padding: 10px;

width: 300px;

}

.dark-mode {

background-color: #333;

color: white;

}

let cart = [];

let totalPrice = 0;

document.querySelectorAll('.add-to-cart').forEach(button => {

button.addEventListener('click', () => {

const name = button.getAttribute('data-name');

const price = parseFloat(button.getAttribute('data-price'));

cart.push({ name, price });

totalPrice += price;

updateCart();

});

});

function updateCart() {

const cartItems = document.getElementById('cart-items');

cartItems.innerHTML = '';

cart.forEach(item => {

const li = document.createElement('li');

li.textContent = `${item.name} - ${item.price} €`;

cartItems.appendChild(li);

});

document.getElementById('total-price').textContent = totalPrice.toFixed(2);

}

document.getElementById('checkout').addEventListener('click', () => {

alert('Compra finalizada');

});

<button id="dark-mode-toggle">Cambiar a Modo Oscuro</button>

const toggleButton = document.getElementById('dark-mode-toggle');

toggleButton.addEventListener('click', () => {

document.body.classList.toggle('dark-mode');

const isDarkMode = document.body.classList.contains('dark-mode');

localStorage.setItem('dark-mode', isDarkMode);

});

// Cargar la preferencia del modo oscuro al cargar la página

window.onload = () => {

const isDarkMode = localStorage.getItem('dark-mode') === 'true';

if (isDarkMode) {

document.body.classList.add('dark-mode');

}

};