**HTML**

<!DOCTYPE html>

<html lang="es">

<head>

</head>

<body>

</body>

</html>

Declaramos que el documento es HTML5 con <!DOCTYPE html>.

lang="es" indica que el idioma principal es español.

Dentro del <head> van los datos técnicos y los enlaces a estilos y fuentes.

Dentro del <body> va todo el contenido visible de la página.

2. Metadatos y estilos

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Productos - Kick Store</title>

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

<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@700&family=Open+Sans:wght@400&family=Bebas+Neue&display=swap" rel="stylesheet" />

charset="UTF-8" define la codificación para que se muestren bien los caracteres.

viewport permite que la página sea responsive (se adapte a móviles).

title es el texto que aparece en la pestaña del navegador.

Se carga una hoja de estilos propia styleproducto.css.

Se importan fuentes de Google Fonts (Montserrat, Open Sans, Bebas Neue).

3. Header (encabezado con navegación)

<header>

<div class="logo"><a href="pagina.html">KickStore</a></div>

<nav>

<a href="pagina.html">Inicio</a>

<a href="productos.html" class="active">Productos</a>

<a href="info.html">Información</a>

<a href="contactos.html">Contacto</a>

<span class="cart-icon" onclick="toggleCart()">

🛒<span class="cart-count" id="cart-count">0</span>

</span>

</nav>

</header>

El logo es un texto que enlaza a la página principal.

El menú tiene enlaces a diferentes secciones del sitio.

El enlace de "Productos" tiene clase active, para indicar que estamos en esa página.

Hay un icono de carrito 🛒 con el contador de productos (cart-count), que al hacer clic llama a la función toggleCart() para mostrar u ocultar el carrito.

4. Sección principal de productos

<section class="products">

<h2>Compra tus Tenis</h2>

<div class="product-list">

<div class="product">

<img src="hola1.jpg" alt="Tenis Urbanos" />

<h3>Tenis Urbanos</h3>

<p>$1,200 MXN</p>

<button onclick="addToCart('Tenis Urbanos', 1200)">Agregar al carrito</button>

</div>

<!-- Dos productos más similares -->

</div>

</section>

El título indica la sección.

La lista tiene varios productos, cada uno con:

Imagen y texto alternativo para accesibilidad.

Nombre del producto.

Precio en pesos mexicanos.

Botón que, al hacer clic, llama a la función addToCart pasando el nombre y precio para añadir ese producto al carrito.

5. Resumen del carrito

<div class="cart-summary" id="cart-summary">

<h3>Carrito de Compras</h3>

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

<p class="total-line"><strong>Total:</strong> $<span id="cart-total">0</span> MXN</p>

<a href="pago.html">

<button>Pagar</button>

</a>

</div>

Este bloque muestra los productos que el usuario agregó al carrito.

ul con id cart-items será llenado dinámicamente con los productos agregados.

Se muestra el total con el id cart-total.

Botón que lleva a la página de pago.

6. Footer (pie de página)

<footer class="footer">

<p>&copy; 2025 Kick Store - Todos los derechos reservados</p>

</footer>

Texto simple que indica derechos reservados y el año.

7. Carga del archivo JavaScript

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

Se incluye el archivo JavaScript que manejará la lógica para agregar productos, mostrar el carrito, etc.

**CSS**

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

Elimina márgenes y rellenos predeterminados de todos los elementos.

box-sizing: border-box hace que el tamaño de los elementos incluya el padding y el borde.

body {

font-family: 'Open Sans', sans-serif;

background: linear-gradient(to right, #340016, #020001);

color: #FE6C2B;

min-height: 100vh;

display: flex;

flex-direction: column;

}

Fuente general: Open Sans.

Fondo degradado horizontal de color vino a negro.

Color principal del texto: naranja (#FE6C2B).

min-height: 100vh: garantiza que el cuerpo ocupe toda la altura de la ventana.

display: flex y flex-direction: column: para organizar los bloques (header, sección y footer) en columna.

🧭 2. Header y navegación

header {

background-color: #020001;

padding: 20px 50px;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.3);

display: flex;

justify-content: space-between;

align-items: center;

}

Fondo negro profundo.

Espaciado interno y sombra inferior.

Distribuye el logo a la izquierda y la navegación a la derecha.

.logo a {

font-family: 'Montserrat', sans-serif;

font-weight: 700;

color: #FE6C2B;

text-decoration: none;

font-size: 1.8rem;

}

El logo usa la fuente Montserrat, se ve grande y destacado en naranja.

nav a {

margin-left: 30px;

font-family: 'Montserrat', sans-serif;

font-weight: 700;

text-decoration: none;

color: #FE6C2B;

transition: color 0.3s;

}

nav a.active,

nav a:hover {

color: #D43B2D;

}

Los enlaces del menú están separados, usan Montserrat y cambian a rojo oscuro al pasar el cursor o si están activos.

🛒 3. Icono del carrito y contador

.cart-icon {

position: relative;

cursor: pointer;

margin-left: 20px;

font-size: 1.6rem;

color: #FE6C2B;

user-select: none;

transition: color 0.3s;

}

.cart-icon:hover {

color: #D43B2D;

}

El carrito cambia de color al pasar el mouse.

.cart-count {

position: absolute;

top: -10px;

right: -10px;

background: #D43B2D;

color: white;

border-radius: 50%;

padding: 2px 6px;

font-size: 0.75rem;

font-weight: 700;

font-family: 'Montserrat', sans-serif;

}

El número del carrito aparece sobre el icono, en un círculo rojo con texto blanco.

👟 4. Sección de productos

section.products {

padding: 50px;

flex-grow: 1;

background: linear-gradient(to bottom, #020001, #340016);

}

Relleno generoso, fondo degradado vertical.

flex-grow: 1 permite que esta sección crezca para ocupar espacio disponible.

section.products h2 {

font-family: 'Montserrat', sans-serif;

font-size: 2.5rem;

text-align: center;

margin-bottom: 40px;

color: #FE6C2B;

text-shadow: 1px 1px 3px #000;

}

Título llamativo centrado con sombra para destacar.

.product-list {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));

gap: 30px;

}

Usa CSS Grid para mostrar productos adaptables a distintas pantallas.

.product {

background: linear-gradient(145deg, #9F102C, #D43B2D);

border-radius: 20px;

padding: 20px;

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.6);

text-align: center;

transition: transform 0.3s;

}

.product:hover {

transform: scale(1.05);

box-shadow: 0 10px 30px #D43B2D;

}

Cada tarjeta de producto tiene un fondo en degradado rojo, esquinas redondeadas, sombra y animación al pasar el mouse.

.product img {

width: 100%;

border-radius: 12px;

box-shadow: 0 5px 15px #020001;

}

Las imágenes ocupan todo el ancho del contenedor y tienen borde redondeado con sombra.

.product h3 {

margin: 15px 0 10px;

font-family: 'Montserrat', sans-serif;

font-size: 1.5rem;

text-shadow: 1px 1px 2px #340016;

}

Nombre del producto con sombra para realce.

.product p {

font-size: 1.2rem;

color: #FFB86C;

font-weight: 600;

text-shadow: 0 0 2px #020001;

}

Precio en un tono más dorado con sombra para que destaque.

🧷 5. Botón de agregar al carrito

button {

font-family: 'Bebas Neue', cursive;

background-color: #FE6C2B;

color: #020001;

border-radius: 12px;

padding: 10px 25px;

font-size: 1.2rem;

box-shadow: 0 4px 8px #9F102C;

cursor: pointer;

transition: background-color 0.3s, color 0.3s;

}

button:hover {

background-color: #D43B2D;

color: #fff;

box-shadow: 0 6px 12px #FE6C2B;

}

Botón llamativo con sombra y cambio de color al pasar el mouse.

🛍️ 6. Carrito flotante (resumen)

.cart-summary {

display: none;

position: fixed;

top: 80px;

right: 20px;

width: 320px;

background: #340016;

border: 1px solid #9F102C;

padding: 20px;

border-radius: 15px;

z-index: 100;

}

Carrito oculto que aparece al hacer clic en el icono.

Posición fija, flotando en la esquina superior derecha.

.cart-summary ul {

max-height: 200px;

overflow-y: auto;

}

.cart-summary ul li {

display: flex;

justify-content: space-between;

font-weight: 600;

color: #FFB86C;

}

.cart-summary ul li button {

background: #D43B2D;

border-radius: 6px;

}

.cart-summary ul li button:hover {

background: #FE6C2B;

}

Lista de productos en el carrito, con botones para eliminar cada uno.

.total-line {

font-weight: 700;

font-size: 1.3rem;

text-align: right;

}

Muestra el total de la compra.

.cart-summary a button {

width: 100%;

font-size: 1.4rem;

margin-top: 10px;

}

Botón de "Pagar" ocupa todo el ancho del carrito.

📌 7. Footer

footer.footer {

background-color: #020001;

color: #FE6C2B;

text-align: center;

padding: 20px 10px;

font-family: 'Montserrat', sans-serif;

font-weight: 700;

}

Pie de página sencillo, con estilo uniforme al resto de la web.

**SCRIPT**

**1. Inicialización del carrito**

let cart = JSON.parse(localStorage.getItem('cart') || '[]');

* Esto intenta obtener el carrito guardado en localStorage.
* Si no hay nada guardado (null), se usa un arreglo vacío ([]).
* JSON.parse convierte el texto guardado en un arreglo de objetos.

**2. Función para agregar productos**

function addToCart(name, price) {

cart.push({ name, price });

localStorage.setItem('cart', JSON.stringify(cart));

updateCartUI();

}

* cart.push({ name, price }): agrega un nuevo objeto con nombre y precio al carrito.
* Luego, localStorage.setItem(...): guarda el carrito actualizado como texto.
* Finalmente, actualiza la interfaz con updateCartUI().

**3. Función para eliminar productos**

function removeFromCart(index) {

cart.splice(index, 1);

localStorage.setItem('cart', JSON.stringify(cart));

updateCartUI();

}

* cart.splice(index, 1): elimina el producto en la posición index del arreglo.
* Se actualiza el carrito en localStorage y también la interfaz.

**4. Actualizar el contenido del carrito en pantalla**

function updateCartUI() {

const cartCount = document.getElementById('cart-count');

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

const cartTotal = document.getElementById('cart-total');

cartCount.textContent = cart.length;

cartItems.innerHTML = '';

let total = 0;

cart.forEach((item, index) => {

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

li.innerHTML = `

${item.name} - $${item.price} MXN

<button onclick="removeFromCart(${index})">X</button>

`;

cartItems.appendChild(li);

total += item.price;

});

cartTotal.textContent = total;

}

* Busca los elementos HTML del contador del carrito (cart-count), la lista de productos (cart-items) y el total (cart-total).
* Limpia la lista anterior con innerHTML = ''.
* Recorre cada producto en el carrito y:
  + Crea un <li> con el nombre, precio y botón para eliminar.
  + Suma el precio al total.
* Finalmente, actualiza el total y el número de productos.

**5. Mostrar u ocultar el carrito**

function toggleCart() {

const summary = document.getElementById('cart-summary');

summary.style.display = summary.style.display === 'block' ? 'none' : 'block';

}

* Esta función se usa cuando haces clic en el ícono del carrito.
* Alterna entre mostrar (block) y ocultar (none) el resumen del carrito.

**6. Actualizar el carrito al cargar la página**

window.addEventListener('load', updateCartUI);

* Cuando la página termine de cargar, se actualiza automáticamente el carrito en pantalla con los datos guardados en localStorage.