**Dinamismos a la Web: Conceptos básicos**

**👋 Introducción**

**¡Hola! 👋Bienvenida/o a este nuevo paso!**

En los pasos anteriores,  te has familiarizado con los conceptos básicos para introducirte en el mundo de JavaScript. Esas prácticas iniciales te permitieron de manera independiente poner a prueba cada uno de los conocimientos adquiridos.

Desde ahora te proponemos continuar avanzando en el proyecto que anteriormente realizaste en tu curso de HTML y lograr  darle dinamismo a tu página web.

**📖 Material Descargable**

A continuación te proporcionamos un material descargable que te permitirá repasar algunos conceptos e incorporar otros básicos necesarios para tu siguiente práctica.

👉[**Teoría Introducción a Javascript : Conceptos básicos**](https://drive.google.com/file/d/1c-Swl97dRnnCNSAF0yeJssbtzUxwxdgb/view?usp=drive_link)

**✅ Checklist**

Asegúrate de:

* Leer el material teórico de Teoría Javascript. Conceptos básicos.
* Tener disponible . Proyecto final del curso HTML.. En caso de no disponer de él, es posible obtenerlo desde [**aquí**](https://drive.google.com/drive/folders/1E8or3lCODa6x8rhwpa5d3f_1ibs9O6us?usp=drive_link).
* Descargar el siguiente prototipo para el paso siguiente: [**Prototipo del carrito**](https://drive.google.com/file/d/17UKSAY4r1IvETlbn48lK9R68iEguQeRs/view?usp=drive_link)

*💡 Revisar estos puntos es muy importante dado que ayudarán a que todos puedan tener la mejor experiencia posible y logren resolver los desafíos correspondiente*s.

# Maquetado básico de un carrito de compras

En este paso, nos enfocaremos en establecer el maquetado básico del carrito de compras.

## ✏️Actividad Carrito de compras

En los próximos pasos, maquetarás 'Tu Carrito de Compras'. Es fundamental utilizar las estructuras proporcionadas por el proyecto Tienda para mantener la coherencia de estilos y seguir utilizando las mismas imágenes.

### 1. Creando el archivo HTML

Crea un archivo llamado cart.html en la carpeta donde tienes los archivos**index.html**y **product.html.**

En tu archivo **cart.html**, comienza escribiendo la estructura básica de un documento HTML. Esto incluye la declaración de tipo de documento (DOCTYPE), la etiqueta**<html>**con los atributos de lenguaje correspondientes, la sección**<head>**que contiene la información del meta y el título de la página. Finalmente, la sección**<body>,** donde irá todo el contenido visible de tu página.

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>Tienda Virtual</title>

  </head>

  <body>

    Aquí desarrollaremos la estructura de nuestro carrito.

  </body>

</html>

En esta etapa, no te preocupes por añadir estilos detallados o contenido específico. La idea es establecer una estructura básica que te permita visualizar la disposición general de la página según el prototipo provisto. Los detalles más finos y estilos específicos se añadirán en pasos posteriores.

Una vez que tu archivo cart.html esté configurado con esta estructura básica, habrás completado este paso.

**📢 Recuerda siempre guardar tus cambios a medida que avances en tu proyecto.**

### 2. Creando la estructura principal

**Definir la Cabecera (Header):** La etiqueta <header> de tu documento HTML será el área donde colocarás elementos como el logotipo de tu tienda, la barra de navegación o cualquier otro contenido que consideres pertinente para la cabecera de tu sitio. Este mismo se encuentra en el archivo index.html Luce algo como:

<header>...</header>

**Agregar Secciones Auxiliares (Sections):** Utiliza la etiqueta <section> para crear distintas secciones auxiliares dentro de tu página. Estas pueden incluir banners promocionales, información adicional sobre productos, o cualquier otro elemento que complemente la experiencia de compra en tu tienda. En nuestro caso, la utilizaremos para crear un área publicitaria destacada con el nombre de la categoría de productos visibles. Este mismo se encuentra en el archivo index.html Luce algo como:

    <section id="hero">...</section>

**Establecer la Sección Principal (Main):**La etiqueta <main> es el corazón de tu carrito. Aquí se ubicará tanto el contenido principal del mismo, como la exhibición de productos y el resumen del pedido.

  <main>

        <div class="cart-container">

            <div class="card resume">

                <div>

                    <h2>Resumen del pedido</h2>

                    <div class="subtotal">

                    <h3>Subtotal</h3>

                    <strong class="price"></strong>

                    </div>

                    <small>Incluye impuesto PAIS y percepción AFIP.</small>

                </div>

            <button class="btn-primary">Finaliza tu compra</button>

            </div>

        </div>

    </main>

**Incluir el Pie de Página (Footer):**Finalmente, la etiqueta <footer> es donde incluirás información como enlaces a tus redes sociales, información de contacto, términos y condiciones, entre otros. El pie de página es un elemento importante para brindar confianza y opciones de contacto adicionales a tus usuarios. Este mismo se encuentra en el archivo index.html.

Luce algo como:

<footer>...</footer>

Una vez hayas añadido estas etiquetas a tu documento HTML, tendrás una **estructura principal sólida**para tu tienda virtual. Asegúrate de guardar los cambios en tu archivo cart.html. En los próximos pasos, nos enfocaremos en manipular dinámicamente los productos como así también calcular el total de nuestro pedido.

El código de tu sitio debería verse así:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

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

    <title>Tienda Virtual</title>

  </head>

  <body>

    <header>...</header>

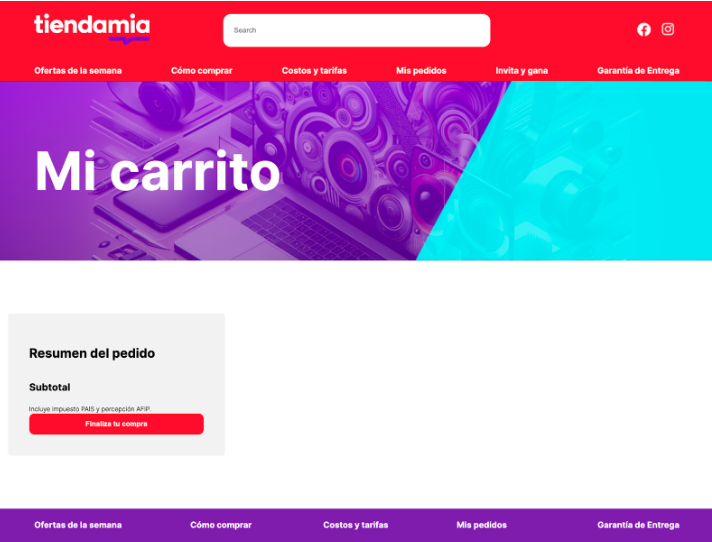
    <section>...</section>

    <main>...</main>

    <footer>...</footer>

  </body>

</html>



# Agregar dinamismo al carrito

En este paso, nos enfocaremos en rellenar dinámicamente el carrito de compras en la página, como así también realizar el resumen del pedido.

## ✏️Actividad resumen del pedido

### 1. Creando la etiqueta <script>

Después del footer y antes del cierre del body, escriba una etiqueta **<script></script>**. La etiqueta**<script>**en un archivo html se utiliza para incorporar o referenciar código JavaScript, que es esencial para añadir interactividad, funcionalidad y comportamientos dinámicos en una página web.

<body>

    <footer>

        ...

    </footer>

    <script></script>

</body>

### 2. JavaScript

#### a- Creación del carrito

A continuación, dentro de la etiqueta <script></script>, crearemos el carrito (por ahora estático):

const cart = [

    {

        title: "MacBook Pro 13'4",

        description:

        "MacBook Pro 13'4 con 8GB de RAM y 512GB de almacenamiento",

        price: 750000,

        quantity: 1,

        image: "../assets/mock1.jpg",

    },

    {

        title: "MacBook Pro 15'4",

        description:

        "MacBook Pro 15'4 con 8GB de RAM y 512GB de almacenamiento",

        price: 800000,

        quantity: 2,

        image: " ../assets/mock1.jpg",

    },

    {

        title: "Lenovo",

        description: "Lenovo 13'4 con 8GB de RAM y 256GB de almacenamiento",

        price: 75000,

        quantity: 1,

        image: " ../assets/mock1.jpg",

    },

];

* cart es un array de objetos, los cuales representan cada producto.
* Cada producto tiene 5 atributos:
  + **title**: Título.
  + **description**: Descripción.
  + **price**: Precio.
  + **quantity**: Cantidad en el carrito.
  + **image**: URL de la imagen.

#### b- Mostrar el carrito a la página

Abajo del array cart, crearemos una función para mostrar cada producto en la página.

**Como vimos en la teoría, hay una propiedad de los elementos del DOM llamada innerHTML, la cual modifica el HTML y no es una buena práctica usarla. Por lo que tendremos que crear cada elemento y agregarle sus propiedades.**

La estructura HTML que necesitamos por cada producto es:

<article class="product-cart">

    <img src="URL de la imagen" alt="">

    <div class="middle">

        <div class="info">

            <strong class="title">

              Título

            </strong>

            <span class="description">

                Descripción

            </span>

        </div>

        <input type="number" min="1">

    </div>

    <strong class="price">Precio</strong>

</article>

La función luce algo como lo siguiente:

function populate() {

    const cartContainer = document.querySelector(".cart-container");

    if (!cartContainer) return console.log("Error");

    const productContainer = document.createElement("div");

    productContainer.classList.add("card", "products");

    cart.forEach((product) => {

        const productElement = document.createElement("article");

        productElement.classList.add("product-cart");

        const productImage = document.createElement("img");

        productImage.src = product.image;

        productImage.alt = product.title;

        const productMiddle = document.createElement("div");

        productMiddle.classList.add("middle");

        const productInfo = document.createElement("div");

        productInfo.classList.add("info");

        const productTitle = document.createElement("strong");

        productTitle.classList.add("title");

        productTitle.textContent = product.title;

        const productDescription = document.createElement("span");

        productDescription.classList.add("description");

        productDescription.textContent = product.description;

        const productInput = document.createElement("input");

        productInput.type = "number";

        productInput.value = product.quantity;

        productInput.min = 1;

        const productPrice = document.createElement("strong");

        productPrice.classList.add("price");

        productPrice.textContent = `AR$ ${formatPrice(product.price)}`;

        productInfo.appendChild(productTitle);

        productInfo.appendChild(productDescription);

        productMiddle.appendChild(productInfo);

        productMiddle.appendChild(productInput);

        productElement.appendChild(productImage);

        productElement.appendChild(productMiddle);

        productElement.appendChild(productPrice);

        productContainer.appendChild(productElement);

    });

    cartContainer.insertAdjacentElement("afterbegin", productContainer);

    const priceElement = document.querySelector(".resume .price");

    priceElement.textContent = formatPrice(

        cart.reduce(

            (total, { price, quantity }) => total + price \* quantity,

            0

        )

    );

}

function formatPrice(price) {

    return new Intl.NumberFormat("es-AR", {

        style: "currency",

        currency: "ARS",

    }).format(price);

}

document.addEventListener("DOMContentLoaded", populate);

## Explicación

### 1. Función populate

Esta función crea y muestra los elementos del carrito.

**Seleccionar el Contenedor:**

* Inicialmente, la función busca y selecciona el contenedor del carrito en el DOM, utilizando **document.querySelector(".cart-container").**
* Este paso es crucial para asegurarse de que tenemos el lugar correcto en el DOM donde se añadirán los elementos del carrito.

**Crear Elementos del Producto:**

* Por cada objeto de producto en el array **cart**, la función genera un nuevo elemento article.
* Este article incluirá varios sub-elementos, como una imagen del producto (img), que representa visualmente el artículo.
* Además, se incluyen elementos para mostrar información detallada del producto, como el título y la descripción.

**Mostrar Precio y Descripción:**

* Cada producto en el carrito tiene su título y descripción claramente mostrados, facilitando la identificación por parte del usuario.
* Se utiliza una función de formato para mostrar el precio de cada producto de una manera legible y localizada, lo que mejora la experiencia del usuario al proporcionar claridad en los costos. (Más info al final del paso)

**Añadir al DOM:**

* Una vez que cada producto ha sido creado y configurado con la información necesaria, se agrega al contenedor del carrito en el DOM.
* Este paso es esencial para que los productos sean visibles en la página web, permitiendo a los usuarios ver los artículos que han agregado a su carrito.

### 2. Función formatPrice

Esta función formatea el precio en el formato de moneda local, en este caso, el peso argentino (ARS).

### 3. Cálculo del Precio Total

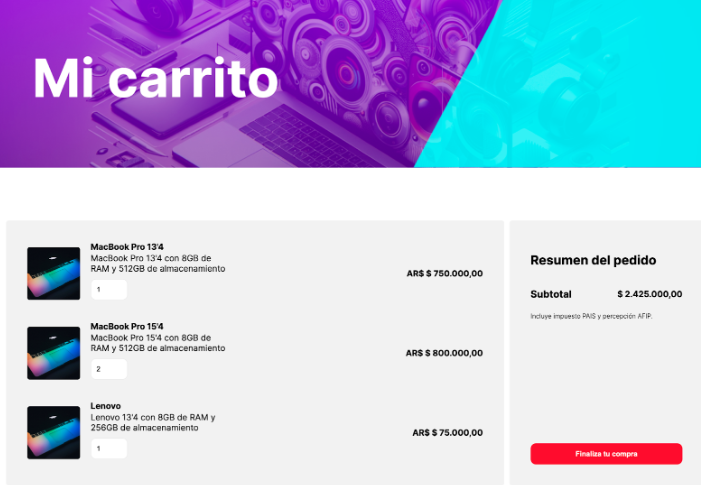
Calcula el precio total de los productos en el carrito, sumando el precio de cada producto multiplicado por su cantidad. Se utiliza el método *reduce* de los arrays, el cual habrá más información al final de este paso.

### 4. document.addEventListener

document.addEventListener("DOMContentLoaded", populate); asegura que el script se ejecute después de que el contenido del DOM esté completamente cargado. Este enfoque es útil para seleccionar elementos del DOM y verificar que estén definidos correctamente.

La función addEventListener la estudiaremos en profundidad más adelante.

#### Tu carrito debería verse así:



### 🧰Material Complementario

Te proporcionamos accesos a recursos de aprendizaje opcionales que te brindarán la oportunidad de profundizar en diversos aspectos:

* [**Format Price**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Intl/NumberFormat)
* [**Método reduce**](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/reduce)

**⏭️ En el paso siguiente** conocerás cómo manejar eventos en JS.