**Click event**

**👋 Introducción**

¡Hola! 👋 ¡Bienvenido a tu nuevo paso! En esta ocasión nos vamos a centrar en el evento click. Pero antes vamos a aprender buenas prácticas ordenando de una mejor manera nuestro código.

**📖 Material Descargable**

Aquí es donde encontrarás el material descargable que complementará los ejercicios prácticos que vamos a ver en los siguientes pasos.

👉[**Teoría Click event**](https://drive.google.com/file/d/1sDiBrJtTTgTDv7ABE8W8-gHijkASyIR4/view?usp=drive_link)

**✅ Checklist**

Asegúrate de:

* Haber entendido y realizado los ejercicios de los pasos anteriores.
* Haber leído la teoría que te va a ayudar a comprender los ejercicios que se muestran en los pasos siguientes.

*💡 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.

**⏭️ En el paso siguiente** veremos cómo implementar el uso del evento click en JavaScript.

# Externalización de código JS

En este paso, aprenderemos a trabajar nuestro JS de una manera más ordenada.

## ✏️ Actividad: Crear archivo js

En vez de escribir todo nuestro código dentro de la etiqueta **<script></script>**, tendremos nuestro propio archivo JS ***(con extensión .js)***, para luego vincularlo con la etiqueta <script></script> en el html.

Para ello, crearemos un archivo en la misma carpeta donde tenemos nuestros html's. Le pondremos de nombre **cart.js.** A continuación, todo el código que teníamos en nuestra etiqueta **<script></script>**lo pegaremos en el nuevo archivo.

## ✏️ Actividad:  Vincular archivo JS con <script></script>

Para vincular el archivo js con la etiqueta, deberemos agregarle un atributo src con la dirección del archivo js.

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

De esta manera, tendremos separado nuestro **js** del **html**, agilizando el desarrollo.

A partir de este punto, todos los nuevos archivos **js**, los vinculamos de esta manera.

# Click event

A continuación comenzaremos con las actividades que te ayudarán a manejar eventos en javascript.

## ✏️ Actividad: Agregar una lista de imágenes al detalle de producto

Necesitamos una lista lateral con todas las imágenes, para ello pegue el siguiente código reemplazando desde la línea 60 a la 67:

<div>

    <div class="product-images-block">

        <div class="thumbnail-container">

            <img class="thumbnail" src="../assets/mock1.jpg" alt="" />

        </div>

        <div class="thumbnail-container">

            <img class="thumbnail" src="../assets/mock2.jpg" alt="" />

        </div>

    </div>

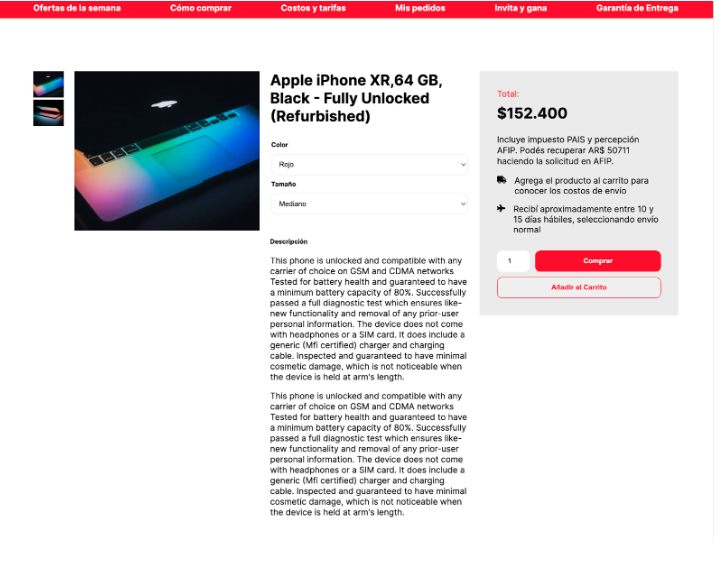
</div>

<div class="thumbnail-container">

    <img id="selected-thumbnail" src="../assets/mock1.jpg" alt="" />

</div>

Debería quedar así:



## ✏️ Actividad: Vincular un nuevo script.

Necesitaremos un script para agregar la funcionalidad del selector de imagen. Para ello, crearemos un archivo llamado product.js, para luego agregarlo al archivo **product.html** en su respectiva etiqueta**<script src="product.js"></script>**

## ✏️ Actividad: JavaScript

Pegue el siguiente código de JavaScript en **product.js:**

document.addEventListener("DOMContentLoaded", ready);

function ready() {

  const thumbnails = document.querySelectorAll(".thumbnail");

  const selected = document.querySelector("#selected-thumbnail");

  if (!thumbnails || !selected) return console.log("Error");

  thumbnails.forEach((item) => {

    item.addEventListener("click", (event) => {

      selected.src = event.target.src;

    });

  });

}

#### 1. Selección de Elementos con querySelectorAll y querySelector:

* **const thumbnails = document.querySelectorAll(".thumbnail");:**
  + querySelectorAll selecciona todos los elementos que coinciden con el selector CSS proporcionado (.thumbnail en este caso).
* **const selected = document.querySelector("#selected-thumbnail");:**
  + querySelector selecciona el primer elemento que coincide con el selector CSS proporcionado (#selected-thumbnail en este caso).

#### 2. Uso de forEach en NodeList:

* **thumbnails.forEach((item) => {...});:**
  + A pesar de que NodeList no es un array, sí tiene algunos métodos de array como forEach. Este método ejecuta la función proporcionada una vez para cada elemento.
* Dentro del forEach, se añade un controlador de eventos de clic a cada elemento thumbnail:
  + **item.addEventListener("click", (event) => {...});:**
    - En el evento de clic, la fuente src de la imagen seleccionada (selected) se actualiza para que coincida con la fuente de la imagen clickeada (event.target.src).

De esta manera, logramos agregarle un evento a cada thumbnail, cambiando el atributo src de la imagen seleccionada.

**⏭️ En el paso siguiente** verás las funcionalidades de arrays en JavaScript