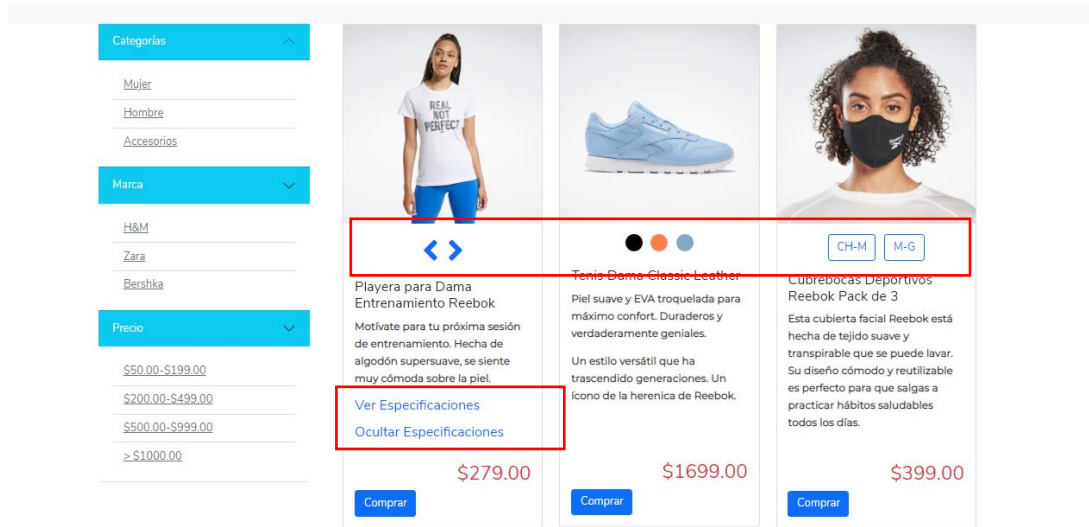


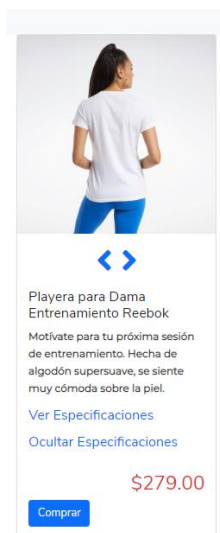
Práctica 7. Introducción a Javascript

Objetivo: Familiarizarse con el lenguaje de Javascript Se trabajará con la página shop.html desarrollada en la práctica 5, se harán ligeras modificaciones al código HTML para que se muestre como sigue (lo enmarcado en rojo se añade):

NOTA: El contenido de los productos de tu página, debes ser coherentes con el proyecto que te fue asignado, en el desarrollo de la práctica se presenta el prototipo de un sitio web de ropa.

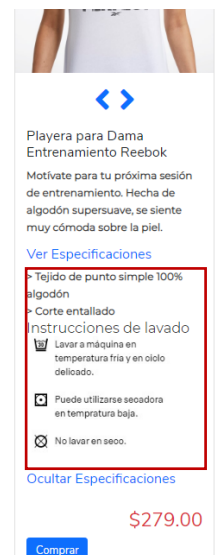


En el producto Playera para Dama Entrenamiento Reebok, el usuario puede dar clic en los iconos < o > e irán cambiando las imágenes, como se muestra en la siguiente imagen.





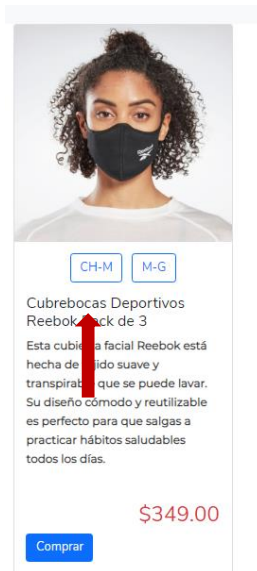
Si el usuario da clic en Ver Especificaciones (imagen anterior señalada con la flecha roja), se muestra una sección con las especificaciones del producto. Si da clic en Ocultar Especificaciones, se ocultan las especificaciones mostradas previamente.



En el producto Tenis Dama Classic Leather, cuando el usuario da clic en los iconos de color debajo de la imagen, la imagen debe cambiar de acuerdo al color seleccionado.



En el producto Cubrebocas Deportivos Reebok Pack de 3, cuando el usuario da clic en el botón CH-M el precio cambia a \$349.00 y cuando da clic en M-G el precio cambia a \$399



Parte 1. Modificación al producto 1 de la Página de Tienda

1. Abre tu proyecto donde has trabajado las prácticas anteriores (5 y 6) y abre la página shop.html, verifica que tengas en el <head> las referencias del FontAwesome y a tu archivo style.css, que las necesitaremos más adelante.
2. Agrega la referencia del archivo de javascript siguiente antes del </body> y abajo del la liga del framework de Bootstrap:

<!--Referencia a Javascript-->

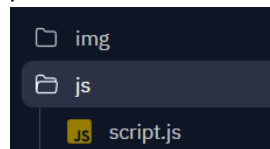
<script src="js/script.js"></script>

```

103 </footer><!--Termina pie de página-->
104 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
w76AqPfdkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwhtBTKf7CXvN" crossorigin="anonymous">
</script>
105 <!--Referencia a Javascript-->
106 <script src="js/script.js"></script>
107
108 </body>
109 </html>

```

- Ahora en la estructura de archivos, agrega la carpeta js y dentro crea el archivo script.js como lo indica la ruta relativa del paso 2.



- Ubica en tu código la estructura del primer producto, recuerda que se estructuró con el componente card, así que ubica el elemento `<div class="card">`. Identifica la imagen asociada al primer producto y agrega el atributo `id="producto"`, es importante este atributo para poder manipular las imágenes por código javascript.

```

<div class="col-lg-4">
  <!--Inicia producto 1-->
  <div class="card">
    
    <div class="card-body">

```

- Identifica debajo de ``, el elemento `<div class="card-body">` y agrega el siguiente código que estructura los iconos `< y >`

```

<div class="col-lg-4">
  <!--Inicia producto 1-->
  <div class="card">
    
    <div class="card-body">
      <!--Iconos adelante y atrás-->
      <ul class="list-inline text-center text-primary">
        <li class="list-inline-item">
          <i id="atras" class="fas fa-angle-left fa-3x"></i>
        </li>
        <li class="list-inline-item">
          <i id="adelante" class="fas fa-angle-right fa-3x"></i>
        </li>
      </ul>
      <!--Fin de Iconos adelante y atrás-->
    </div>
    <h5 class="card-title">Playera para Dama Entrenamiento Reebok</h5>
  </div>
</div>

```

Nota: Para que sean visibles los iconos en tu página shop.html debes tener la referencia la Framework de FontAwesome. Y no olvides agregar los atributos de id porque serán necesarios para trabajar con javascript.

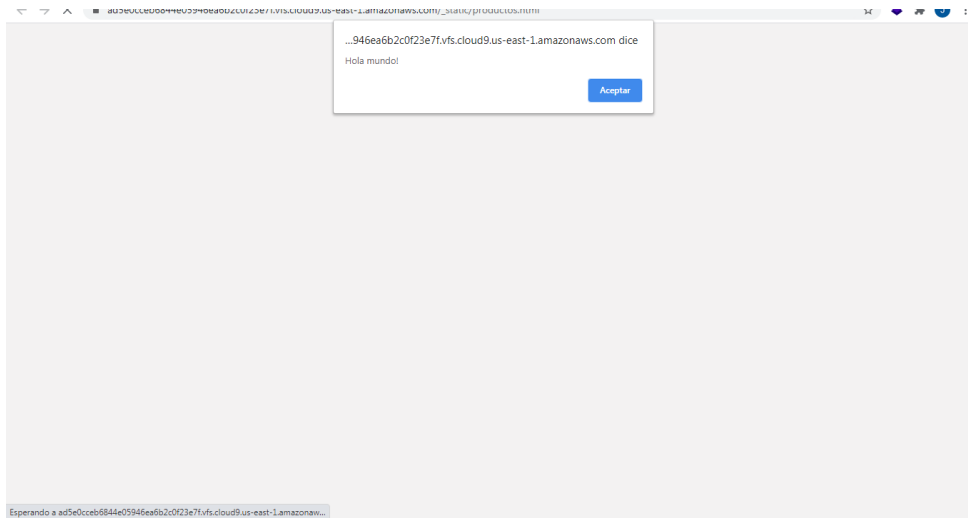
Parte 2. Validación del archivo javascript

- Ahora abre tu archivo script.js agrega el siguiente código:

```
alert("¡Hola mundo!");
```

NOTA: La instrucción anterior mostrará el mensaje ¡Hola mundo! en un recuadro antes de que inicie la carga de la página.

- Ejecuta tu proyecto y visualiza la página shop.html y debes ver el mensaje como sigue:



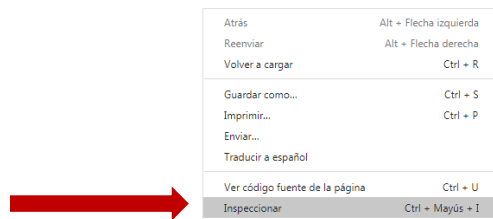
3. Si aparece el mensaje, sólo da aceptar. Eso significa que tu archivo javascript se está cargando correctamente, en caso contrario, verifica tu ruta, nombre de archivo.

Parte 3. Uso de la consola para visualizar errores.

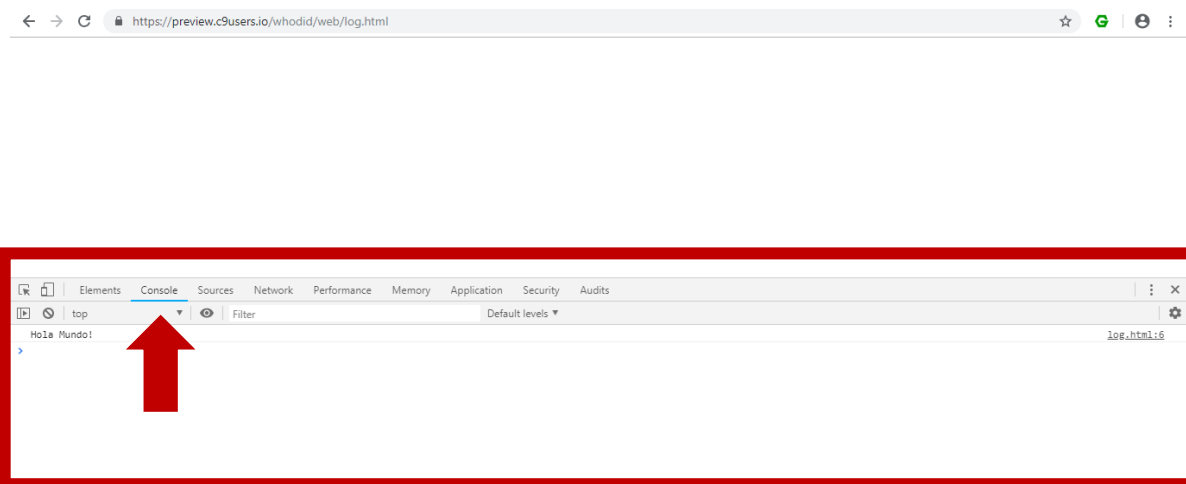
1. Dado que javascript es un lenguaje interpretado, si nuestro código tiene errores simplemente no se ejecutará el código. Una forma de identificar si tenemos algún error es enviando mensajes a la consola de desarrollador dentro de un bloque de ejecución.

Para enviar mensajes a consola lo haremos con la función **console.log()**, esta función no tiene efecto para el usuario, solo muestra en consola de desarrollador los mensajes.

2. Para acceder a la consola del navegador, da clic derecho en la vista de usuario de shop.html y selecciona el menú Inspeccionar



Se abrirá en tu página una sección como se muestra, dirígete al menú Console

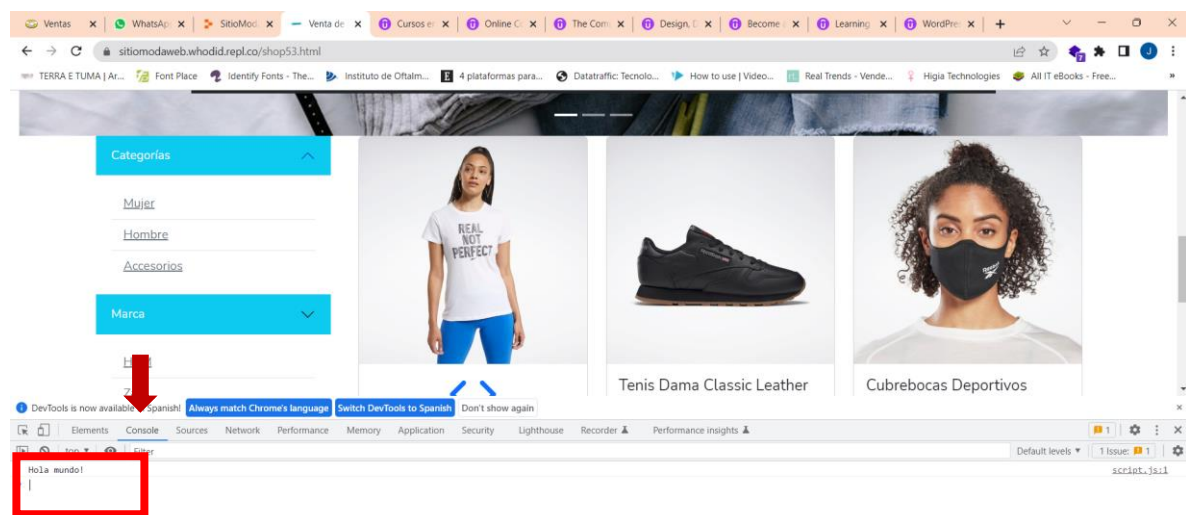


En esta sección podemos identificar algunos errores o bien, recibir los mensajes que se envían como símbolo de que nuestro código se está ejecutando.

3. Para hacer la prueba, en tu archivo script.js sustituye la instrucción `alert("Hola mundo!");` por

`console.log("Hola mundo!");`

Abre tu página shop.html con la consola de desarrollador abierta. En la sección de Console se visualizará el mensaje Hola mundo!



4. Cuando tu código no haga lo esperado, revisa esta consola para determinar los indicios de error o para mandar mensajes antes o después de una línea de código de javascript, los

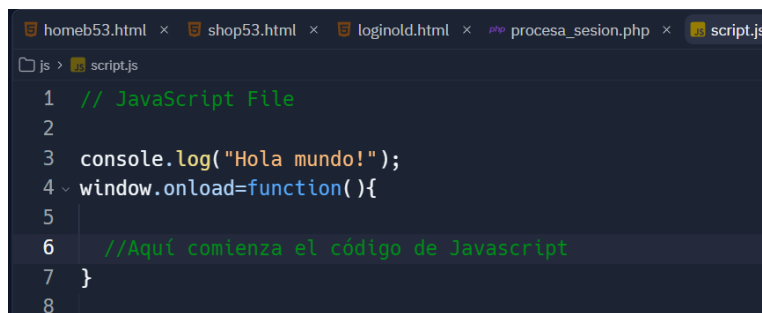
mensajes que se muestran indican que el código anterior se ejecutó, si no se muestra un mensaje en particular significa que antes de imprimir el mensaje hubo una falla y te dará idea de que línea de código atender.

Para más información del uso y depuración de tecnologías front-end, revisa a detalle las funcionalidades de las herramientas para desarrolladores: <https://support.google.com/dcm/answer/2828688?hl=es>

Parte 5. Añadiendo código javascript funcional

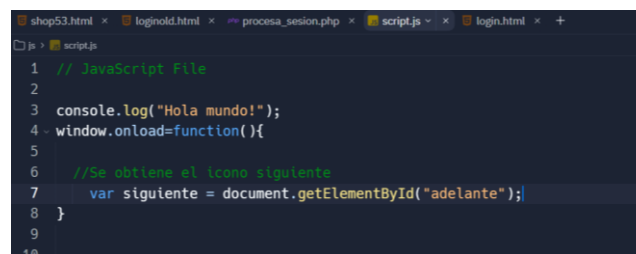
1. A continuación, se agregará el código javascript para que una vez que está cargada la imagen inicial, al dar clic en el icono siguiente > se cambie la imagen.
2. En tu archivo script.js captura el nombre de la función `window.onload=function(){ }`

Entre las llaves irá todo nuestro código javascript y esta función sirve para que se ejecute nuestro código javascript hasta que se haya cargado toda nuestra página html y así evitar variables nulas.



```
1 // JavaScript File
2
3 console.log("Hola mundo!");
4 window.onload=function(){
5
6 //Aquí comienza el código de Javascript
7 }
8
```

3. En tu archivo script.js dentro de las llaves de la función `window.onload=function()` se va a obtener en un variable el elemento del DOM del icono >



```
1 // JavaScript File
2
3 console.log("Hola mundo!");
4 window.onload=function(){
5
6 //Se obtiene el icono siguiente
7 var siguiente = document.getElementById("adelante");
8 }
9
10
```

Recuerda que “adelante” es el id de un elemento que representa el icono > en tu código html y lo que estamos haciendo es obtenerlo en una variable para manipularlo.

4. Dado que el icono se tiene guardado en la variable siguiente, ahora se le va a registrar un evento, para que cuando se dé clic en el icono con id “adelante”, se va a ejecutar la función `verSiguiente` (la función se va a encargar de cambiar la imagen)


```
shop53.html x loginold.html x procesa_sesion.php x script.js x login.html x +
js > script.js
1 // JavaScript File
2
3 console.log("Hola mundo!");
4 window.onload=function(){
5
6 //Se obtiene el icono siguiente
7 var siguiente = document.getElementById("adelante");
8 //Se registra el evento
9 siguiente.addEventListener("click", verSiguiente);
10 }
11
12
```

Con la función `addEventListener` se registra un evento a un elemento del DOM, cuyo parámetro es el evento a registrar (en este caso "click") y el otro parámetro el nombre de la función a ejecutarse cuando se le dé clic al elemento DOM

- Ahora resta crear la función `verSiguiente` y las instrucciones para que cambie la imagen, para crear una función es con la instrucción `function nombrefuncion() { }`

```
shop53.html x loginold.html x procesa_sesion.php x script.js x login.html x +
js > script.js
1 // JavaScript File
2
3 console.log("Hola mundo!");
4 window.onload=function(){
5
6 //Se obtiene el icono siguiente
7 var siguiente = document.getElementById("adelante");
8 //Se registra el evento
9 siguiente.addEventListener("click", verSiguiente);
10
11 //Función que maneja el cambio de imágenes
12 //cuando el usuario da clic en el icono siguiente
13 function verSiguiente() {
14
15 //Aqui va el código de la función
16
17 }
18 }
19
20
```

- En el cuerpo de la función `verSiguiente` captura el siguiente código

```
shop53.html x loginold.html x procesa_sesion.php x script.js x login.html x +
js > script.js
//Se registra el evento
siguiente.addEventListener("click", verSiguiente);

//Función que maneja el cambio de imágenes
//cuando el usuario da clic en el icono siguiente
function verSiguiente() {

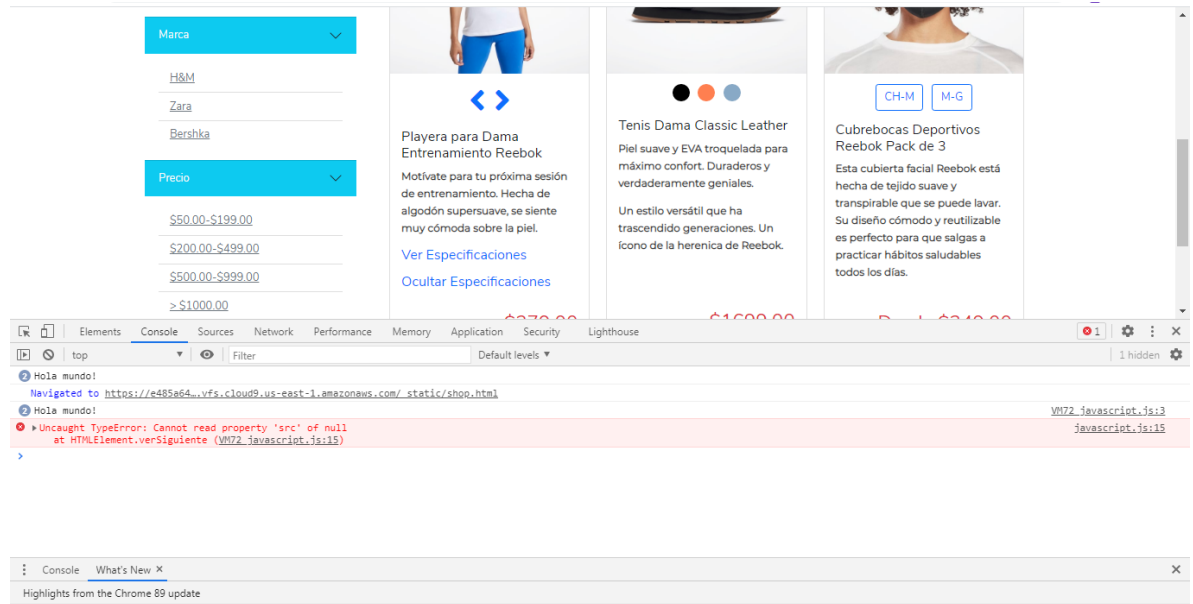
//Aqui va el código de la función
var aux = document.getElementById("producto")
var ruta= aux.src;
if (ruta.includes('img/playera01.jpg'))
    aux.src="img/playera02.jpg";
else if(ruta.includes('img/playera02.jpg'))
    aux.src="img/playera03.jpg";
else if(ruta.includes('img/playera03.jpg'))
    aux.src="img/playera04.jpg";
else
    aux.src="img/playera01.jpg";

}
}
```

35 : Col 39 (424 chars)

En la variable aux, se obtiene el elemento del DOM de la imagen y se verifica la ruta de la imagen para saber si contiene el nombre de una de las imágenes esperadas para asignarle una diferente de las disponibles 4 disponibles a través de sentencias de control if-else.

7. Ejecuta tu código y prueba el icono >, da clic para verificar que las imágenes cambian.
8. Si no hace lo esperado, verifica tu consola para obtener evidencia de los errores, puede verse algo similar a:



Los mensajes en rojo en consola significan que hay errores en el código de Javascript y dicho mensaje te puede dar evidencia de tu error.

Puedes apoyarte de la instrucción `console.log ()` para escribir mensajes a lo largo de tu código y verificar si se está ejecutando, por ejemplo en la función `verSiguiente()` puedes agregar un mensaje al inicio, si al dar clic icono se muestra el mensaje en consola quiere decir que si se está ejecutando tu evento y de ahí partir para verificar los errores.

9. Si tu código funciona correctamente, ahora integra el código de forma similar para que al dar clic en el icono atrás <, la secuencia de fotos vaya en sentido contrario.

Consideraciones:

- El código va justo abajo del código que ya tienes, pero dentro de la llave de cierre de la función `window.onload=function() {}`
- No olvides que tu icono atrás (<) tiene `id=atrás`, puedes confirmarlo en tu código HTML
- Apóyate de la siguiente estructura para completar el código

```

22     else
23         aux.src="img/playera01.jpg";
24     }
25
26     //Se obtiene el icono atras
27     var vista_atras = document.getElementById("atras");
28     //Se registra el evento
29     vista_atras.addEventListener("click", verAtras);
30
31     //Función que maneja el cambio de imágenes
32     //cuando el usuario da clic en el icono atras
33     function verAtras() {
34
35         //Aquí va tu código
36     }
37

```

10. Prueba tu código y corrige los errores si son necesarios.

Parte 5. Agregar contenido dinámico con HTML

Se van a agregar funcionalidad a los encabezados Ver Especificaciones y Ocultar Especificaciones de tal forma que cuando se dé clic en Ver Especificaciones, dinámicamente se genere el contenido de las especificaciones y al dar clic en Ocultar Especificaciones se eliminen

1. En tu código ubica la descripción de tu producto que está estructurado por un párrafo `<p class="card-text"></p>` Debajo de esta estructura agrega el siguiente código, que incluye un h5, un div anidado con un párrafo y otro h5:

```

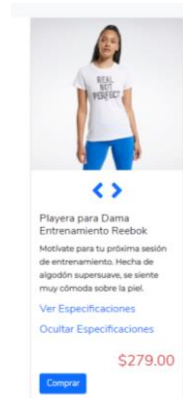
<p class="card-text">Motivate para tu próxima sesión de entrenamiento.
Hecha de algodón supersuave, se siente muy cómoda sobre la piel.</p>
<h5 class="text-primary" id="ver">Ver Especificaciones</h5>
<div>
  <p id="datos"></p>
</div>
<h5 class="text-primary" id="ocultar">Ocultar Especificaciones</h5>
<br>
<h3 class="text-danger text-end pr-5">$279.00</h3>
<a href="#" class="btn btn-primary text-light">Comprar</a>
</div>
</div>

```

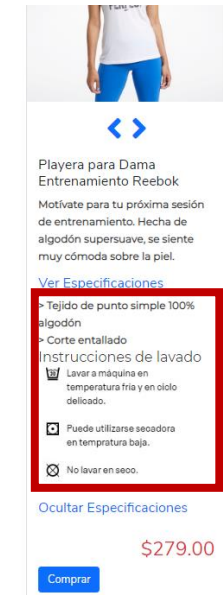
Especificaciones:

- El Encabezado “Ver Especificaciones” debe tener id=”ver”
- El Encabezado “Ocultar Especificaciones” debe tener id=”ocultar”
- Dentro de un <div></div> tener un párrafo con id=”datos”

Estado Inicial de la Página



Al dar clic Ver Especificaciones se debe generar dinámicamente el siguiente contenido enmarcado en rojo:



Se deben mostrar las especificaciones del producto, que se conforma por un texto y una imagen, elige una imagen de acuerdo a tu producto y cárgala a tu proyecto.

La generación de este código debe hacerse solamente con Javascript

1. En tu archivo script.js, dentro de la función `window.onload=function() {}` y justo debajo de tu código generado en la Parte 4, obtén en una variable llamada `btn` el elemento DOM cuyo id es `ver`, utiliza la función `document.getElementById()`
`var btn= document.getElementById("ver");`

2. Registra a la variable btn el evento click que llame a la función mostrarEspecificaciones, utiliza la función addEventListener
 btn.addEventListener("click", mostrarEspecificaciones)
3. Crea la función mostrarEspecificaciones, pero antes revisa la teoría necesaria que se presenta

```

53
54 //Se obtiene el elemento h5 Ver Especificaciones
55 var btn= document.getElementById("ver");
56 btn.addEventListener("click", mostrarEspecificaciones)
57
58 //Función que crea y añade elementos html al DOM
59 function mostrarEspecificaciones()
60 {
61
62 //Aquí va el código
63
64 }
65

```

TEORÍA NECESARIA: Para crear contenido dinámico, necesitamos dos métodos:

document.createElement(*elementHTML*): Permite crear etiquetas html dinámicamente, donde, *elementHTML* es la etiqueta html que se desea crear; ejemplo de uso:

```
var y= document.createElement("p");
```

En la variable **y** se crea un elemento html tipo p (párrafo).

Para más información de uso:

https://www.w3schools.com/jsref/met_document_createelement.asp

Element1.appendChild(*element2*): Permite añadir un elemento html a otro elemento html, donde, *element2* se añade a *element1*; ejemplo de uso:

```
var y= document.createElement("h1"); //Crea elemento de encabezado
```

```
var w= document.createElement("p"); //Crea element de párrafo
```

```
y.appendChild(w); //Al encabezado le añade un párrafo.
```

Para más información: https://www.w3schools.com/jsref/met_node_appendchild.asp

Se hará uso de la propiedad innerHTML:

htmlElement.innerHTML="Texto dinámico"

Esta propiedad permite cambiar o agregar contenido a un elemento html específico, ejemplo de uso:

```
var z= document.createElement("p"); //Se crea un parrafo
```

```
z.innerHTML="Algun texto visible para usuario"; //Se agrega contenido al párrafo
```

Para mas información: https://www.w3schools.com/jsref/prop_html_innerhtml.asp

4. Se van a utilizar las funciones y propiedad anteriores para la generación dinámica de contenido. Dentro de la función mostrarEspecificaciones(), obtén en una variable llamada text con el elemento DOM párrafo con id datos, usa la función document.getElementById()

```
var text= document.getElementById("datos");
```

5. Agrega la primera característica del producto con la propiedad innerHTML como sigue, personalízalo de acuerdo a tu producto:

```
text.innerHTML="> Tejido de punto simple 100% algodón <br> > Corte entallado";  
Ahora crear otra variable llamada instrucciones, en la que se debe crear un elemento  
DOM tipo h5 (de encabezado 5) usa la función document.createElement("h5")
```

```
var instrucciones= document.createElement("h5");
```

6. Con la propiedad innerHTML agrega el texto "Instrucciones de lavado" a la variable instrucciones;

```
instrucciones.innerHTML="Instrucciones de lavado";
```

7. Ahora a la variable text añade el elemento recién creado llamado instrucciones con la función appendChild.

```
text.appendChild(instrucciones);
```

8. Ahora crea una variable llamada img_instrucciones, en la que se debe crear un elemento DOM tipo img (de una imagen) usa la función document.createElement()

```
var img_instrucciones= document.createElement("img");
```

9. Asigna a la variable img_instrucciones la propiedad src con la imagen de cuidados.jpg como sigue: img_instrucciones.src="img/cuidados.jpg", esta imagen debe ser personalizada de acuerdo a tu producto y asegúrate de cargarla en la carpeta img.

10. Ahora a la variable instrucciones añade el elemento img_instrucciones con ayuda de la función appendChild

```
instrucciones.appendChild(img_instrucciones);
```

11. Con la anterior secuencia de instrucciones lo que se debe lograr es crear elementos html dinámicamente.

El código debe quedar similar a:

```
54 //Se obtiene el elemento h5 Ver Especificaciones
55 var btn= document.getElementById("ver");
56 btn.addEventListener("click", mostrarEspecificaciones)
57
58 //Función que crea y añade elementos html al DOM
59 function mostrarEspecificaciones()
60 {
61
62     var text= document.getElementById("datos");
63     text.innerHTML="> Tejido de punto simple 100% algodón <br> > Corte entallado";
64     var instrucciones= document.createElement("h5");
65     instrucciones.innerHTML="Instrucciones de lavado";
66     text.appendChild(instrucciones);
67     var img_instrucciones= document.createElement("img");
68     img_instrucciones.src="img/cuidados.jpg"
69     instrucciones.appendChild(img_instrucciones);
70
71 }
72
```

Line 69 : Col 54 (453 chars) Histor

12. Ejecuta tu código, abriendo la página shop.html y prueba la funcionalidad del encabezado Ver Especificaciones.

Parte 6. Borrar especificaciones

Implementa el borrado de los elementos que se crearon en la parte 5. Cuando el usuario de clic en el encabezado Ocultar Especificaciones deben desaparecer los datos de la especificación del producto

- Implementa este código justo abajo del código que ya tienes

```
76
77 //Función elimina las especificaciones
78 function ocultarEspecificaciones()
79 {
80
81     var elemento= document.getElementById("datos");
82     while (elemento.firstChild) {
83         elemento.removeChild(elemento.firstChild);
84     }
85
86 }
87
```

Ejecuta tu código, abriendo la página shop.html y prueba la funcionalidad del encabezado Ocultar Especificaciones.

Parte 7. Modifications a Producto 2

1. Se ve a modificar la estructura del segundo producto:



2. Ubica el código de tu segundo producto, a la imagen agrega el id="producto2"

```

228 -         <div class="col-lg-4">
229 -             <div class="card">
230 -                 
231 -             <div class="card-body">

```

3. Después del <div class="card-body"> agrega el siguiente código

```

228 -         <div class="col-lg-4">
229 -             <div class="card">
230 -                 
231 -             <div class="card-body">
232 -                 <!-- Inician iconos-->
233 -                 <ul class="list-inline text-center">
234 -                     <li class="list-inline-item">
235 -                         <a id="negro" style="background:black; " class="color"></a>
236 -                     </li>
237 -                     <li class="list-inline-item">
238 -                         <a id="coral" style="background:coral; " class="color"></a>
239 -                     </li>
240 -                     <li class="list-inline-item">
241 -                         <a id="azul" style="background:#88a9c6; " class="color"></a>
242 -                     </li>
243 -                 </ul>
244 -                 <!-- Inician iconos-->
245 -                 <h5 class="card-title">Tenis Dama Classic Leather</h5>
246 -                 <p class="card-text">Piel suave y EVA troquelada para máximo confort.

```

Nota: Los id nuevamente son importantes, puedes personalizar el color de la propiedad background para personalizar

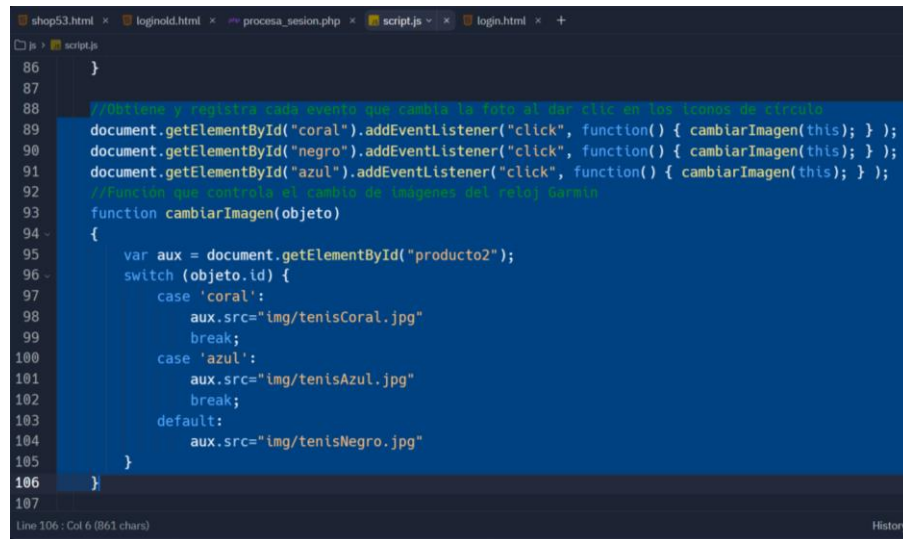
4. En el código se hace referencia a una clase llamada color, agrega el siguiente código css al final de tu archivo style.css, estas reglas permiten que la liga se muestre como un círculo de un color definido

```

14 .color
15 {
16     height: 25px;
17     width: 25px;
18     border-radius: 50%;
19     display: inline-block;
20 }
21 }
22

```


5. En tu archivo script.js agrega el siguiente código justo debajo del código agregado en la parte 6 de la práctica y antes del cierre de la llave de la instrucción `window.onload=function(){ }`



```
86 }
87
88 //Obtiene y registra cada evento que cambia la foto al dar clic en los iconos de círculo
89 document.getElementById("coral").addEventListener("click", function() { cambiarImagen(this); });
90 document.getElementById("negro").addEventListener("click", function() { cambiarImagen(this); });
91 document.getElementById("azul").addEventListener("click", function() { cambiarImagen(this); });
92 //Función que controla el cambio de imágenes del reloj deportivo
93 function cambiarImagen(objeto)
94 {
95     var aux = document.getElementById("producto2");
96     switch (objeto.id) {
97         case 'coral':
98             aux.src="img/tenisCoral.jpg"
99             break;
100         case 'azul':
101             aux.src="img/tenisAzul.jpg"
102             break;
103         default:
104             aux.src="img/tenisNegro.jpg"
105     }
106 }
107
```

La primera instrucción `document.getElementById("coral").addEventListener("click", function() { cambiarImagen(this); });` es equivalente a declarar una variable para obtener el elemento del DOM y luego a esa variable registrar el evento (como se realizó en la parte 5 de la práctica), se puede hacer en una sola instrucción como se muestra en la primeras tres líneas, en cada instrucción se obtiene el icono de cada color y se registra el evento `cambiarImagen(this)`.

A diferencia de lo realizado, la función `cambiarImagen` recibe como parámetro el objeto del DOM que la invoca para saber con base en el id que objeto es y así seleccionar la imagen a través de una sentencia de control `switch`

La variable `aux`, sirve para obtener la imagen que se va ir cambiando de acuerdo al icono de color seleccionado.

6. Ejecuta tu código, abriendo la página `shop.html`, cuando el usuario da clic en el icono coral la imagen cambia a un tenis de color coral, cuando se da clic en el icono azul la imagen cambia a un tenis de color azul y así para el icono negro, la imagen cambia a un tenis de color negro. Personaliza tu producto.
7. Corrige tus errores de ser necesario.

Parte 8. Modificaciones a Producto 3

1. Se modificará la estructura del producto 3, agregando 2 botones:



Ubica el código de tu tercer producto, dentro del `<div class="card-body">`, agrega el siguiente código

```

258 - <div class="card">
259 -   
260 -   <div class="card-body">
261 -     <!-- Botones -->
262 -     <ul class="list-inline text-center text-primary">
263 -       <li class="list-inline-item">
264 -         <button id="chico" class="btn btn-outline-primary" type="button">CH-M</button>
265 -       </li>
266 -       <li class="list-inline-item">
267 -         <button id="mediano" class="btn btn-outline-primary" type="button">M-G</button>
268 -       </li>
269 -     </ul>
270 -     <!-- Botones -->
271 -     <h5 class="card-title">Cubrebocas Deportivos Reebok Pack de 3</h5>
272 -     <p class="card-text">Esta cubierta facial Reebok está hecha de tejido suave
273 -       y transpirable que se puede lavar. Su diseño cómodo y reutilizable es perfecto
274 -       para que salgas a practicar hábitos saludables todos los días.</p>
275 -   </div>

```

En el párrafo que contiene el precio agrega el `id="precio"`

```

272 - <p class="card-text">Esta cubierta facial Reebok está hecha de tejido suave
273 -   y transpirable que se puede lavar. Su diseño cómodo y reutilizable es perfecto
274 -   para que salgas a practicar hábitos saludables todos los días.</p>
275 -   <!--<div style="height:70px;"></div-->
276 -   <h3 id="precio" class="text-danger text-end pr-5">Desde $349.00</h3>
277 -   <a href="#" class="btn btn-primary text-light">Comprar</a>
278 - </div>
279 - </div>
280 - </div>
281 -

```

- En tu archivo `script.js` agrega el siguiente código justo debajo del código agregado en la parte 8 de la práctica y antes del cierre de la llave de la instrucción `window.onload=function(){ }`

```
104     aux.src = "img/centronegro.jpg";
105   }
106 }
107
108 //Obtiene y registra cada evento que cambia el precio de la pantalla
109 document.getElementById("chico").addEventListener("click", function() { cambioPrecio(this); });
110 document.getElementById("mediano").addEventListener("click", function() { cambioPrecio(this); });
111 function cambioPrecio(objeto) {
112
113     var precio= document.getElementById("precio");
114     switch (objeto.id) {
115         case 'mediano':
116             precio.innerHTML="$399.00"
117             break;
118
119         default:
120             precio.innerHTML="$349.00"
121     }
122 }
123 } //Fin de window.onload
124
```

El código anterior permite que cuando se da clic en alguno de los botones, se evalúa cual fue el botón al que se le dio clic para cambiar el precio dinámicamente.

3. Ejecuta tu proyecto, abre shop.html y prueba el tercer producto, asegúrate que al seleccionar un botón cambia el precio. En caso de errores corrige.
4. Asegúrate que los tres productos funcionen de acuerdo a las instrucciones de la práctica y puedes considerar tu práctica como terminada.