

PROGRAMACIÓN BÁSICA EN JAVASCRIPT



Módulo 2





CLASE 7

Introducción

DOM

Manipular los elementos

Manipular los estilos

Texto y contenido

Manipular los atributos

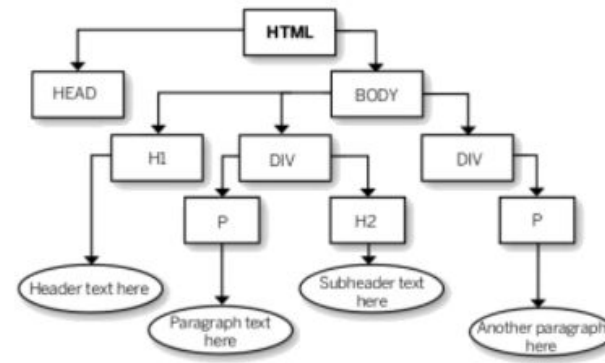
Manipulación de varios elementos

Crear, agregar y remover elementos

Introducción

En la segunda clase de este Bootcamp, hace tan solo 14 clases atrás... Le hemos mostrado la siguiente diapositiva:

El browser, entonces, construye una representación de una página en memoria que se llama Document Object Model (DOM). El DOM tiene un esquema de estructura representada aquí al lado.



Ahora, profundizaremos todo esto, ya que prometimos volver a ver este tema... Pero antes, una advertencia...



Es absolutamente importante que tenga bien presente todos los conceptos fundamentales de HTML y CSS. Ya que los volveremos a utilizar de ahora en más.



¿Qué es el *DOM*?

Para saber que es el DOM, debemos recordar que el browser construye una representación del documento HTML que se llama Document Object Model (DOM).

En este sentido, el DOM lo organiza en forma de **Nodos** y **Objetos** que tienen **Propiedades** y **Métodos**.

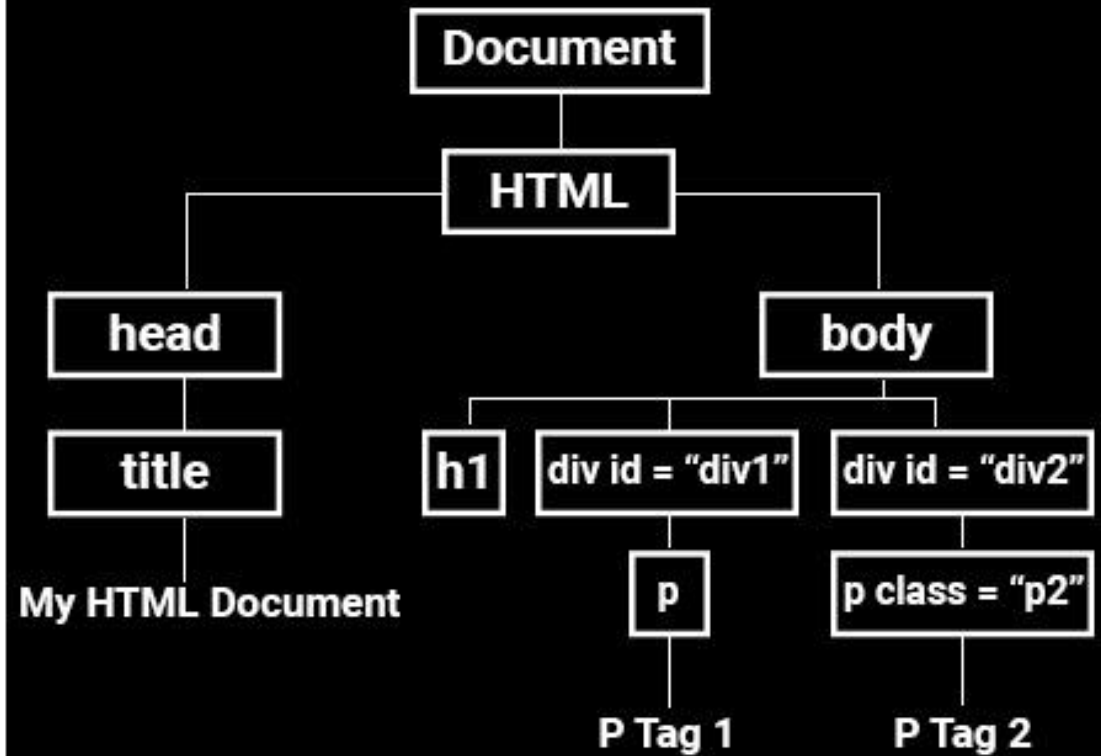
Esencialmente conecta las páginas web a los scripts o lenguajes de programación. Es decir, **el DOM representa todo el contenido del documento HTML y permite que el script de JavaScript interactúe con él.**

Cuando cargamos un documento HTML+CSS en el browser, el DOM convierte cada tag en un Objeto de JavaScript. Esto permite que el script manipule los elementos HTML+CSS como si fueran Objetos.

HTML Document

```
index.html x
1 <html>
2   <head>
3     <title>My HTML Document</title>
4   </head>
5
6   <body>
7     <h1>Heading</h1>
8     <div id="div1">
9       <p>P Tag 1</p>
10    </div>
11    <div id="div2">
12      <p class="p2">P Tag 2</p>
13    </div>
14  </body>
15 </html>
```

Document Object Model (DOM)



+

- El DOM tiene muchísimas Propiedades. **IGNORE LA COMPLEJIDAD** y busque solo lo que necesita.
-

+

-
-

Manipular los elementos



Los `document.querySelector()`: Es uno de los métodos para seleccionar un elemento. Devuelve solo el primer elemento que coincida con el selector CSS dado. El término “Query Selector” viene del inglés “consulta por selector”.

Estos métodos pueden ser asignados a variables. Por ejemplo:

```
let parrafo= document.querySelector("p");
```

Luego le cambiamos su color:

```
parrafo.style.color="red";
```

```
document.body;  
document.querySelector("p");  
document.querySelector("#primary");  
document.querySelector(".important");  
document.querySelector("p.important");  
document.querySelector("#nav ol li");  
document.querySelector("h1, p");
```


RECUERDE

- En vez de cambiar todas las Propiedades del estilo desde JavaScript, la forma más escalable de hacerlo es a través de la Clase en el documento CSS.

Manipular el estilo

Si quisiéramos modificar las Propiedades de un documento, podríamos seleccionar un elemento con un **querySelector** y, luego, modificar su estilo. Sin embargo, esto es **poco escalable**.

La alternativa más práctica es definir una **Clase** con todas las propiedades CSS y aplicarla usando JavaScript. Para eso, usaremos un método llamado **classList**, en el que seleccionamos un elemento y le agregamos una Clase determinada pasándola como valor.

La **Propiedad classList** nos permite acceder a las Clases presentes en un elemento. A su vez, podemos combinarlo con ciertos métodos para manipular el estilo.

Manipular el estilo

Veamos un ejemplo:

```
h2.classList.add("importante");
```

En este caso, le agregamos la Clase importante a un `<h2>` utilizando el método `add` (del inglés "agregar").

```
h2.classList.remove("importante");
```

De la misma manera, podríamos suprimir la Clase importante utilizando el método `remove` (del inglés "quitar"):

```
h2.classList.toggle("importante");
```

También hay un método llamado `toggle` (del inglés "alternar") que **cambia entre sacar y agregar** la Clase. En otras palabras, si la Clase está presente, `toggle` la quita; si está ausente, la agrega.

¿RECUERDA QUÉ ES UN ATRIBUTO?

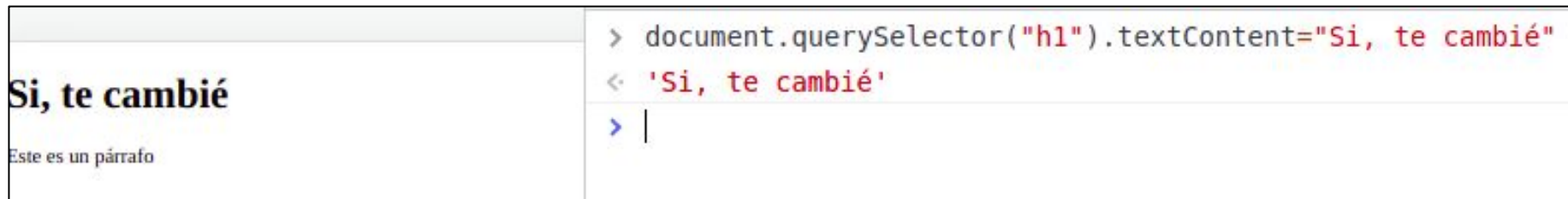
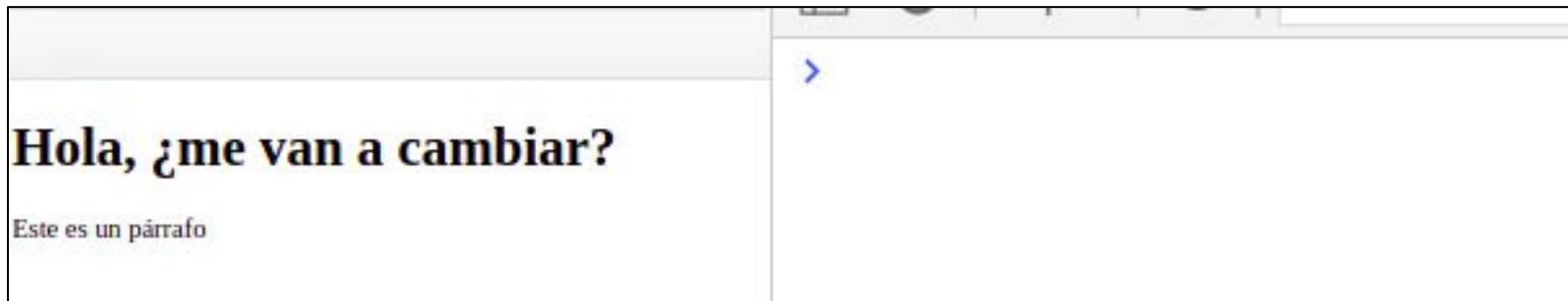
- - Los elementos en HTML tienen atributos; estos son valores adicionales que configuran los elementos o ajustan su comportamiento de diversas formas para cumplir los criterios de los usuarios.

Manipular el texto y el contenido

Para manipular los textos del HTML usaremos la propiedad `textContent` e `innerHTML`.

Con el `textContent` solo vamos a poder manipular los textos de los Tags, en otras palabras, el contenido que trae ese Tag.

Pero en caso de necesitar transformar una Etiqueta, utilizaremos el `innerHTML`, ya que nos trae toda la declaración del HTML (nos devuelve el Tag con el contenido dentro).



Manipular los atributos de un elemento

El método `getAttribute()` nos devuelve el valor del atributo especificado. Con el método `setAttribute()` definiremos un valor para el atributo.

Veamos un ejemplo. Supongamos que tenemos el siguiente tag:

```
<a href="http://www.google.com" id="btnGoogle">Ir a Google</a>
```

Por ahora el link nos lleva a www.google.com, pero queremos modificarlo por <http://plataforma5.iicap.cl/> Para hacerlo, primero debemos usar el método `querySelector` desde JavaScript:

```
let botonGoogle = document.querySelector("#btnGoogle");
```

Luego, usaremos `getAttribute()` y `setAttribute()`:

```
botonGoogle.getAttribute("href") // Nos muestra cuál es el valor del atributo href.
```

```
botonGoogle.setAttribute("href","http://plataforma5.iicap.cl/") // Cambia el valor del atributo href por el valor que se pasa como segundo Parámetro.
```

De esta manera, reemplazamos un atributo href por el otro.

IMPORTANTE

- El atributo `src` es una Propiedad del Objeto `img` que tiene como valor una dirección en donde está alojada la imagen.

Manipular varios elementos

Bien sabemos que si utilizamos `querySelector("a")`, solamente nos trae el primer link.

En cambio, `querySelectorAll("a")` nos devuelve una **lista de nodos** con todos los elementos de este tipo.

Ahora bien, si quisiéramos modificar **todos los links** de nuestra página, no podremos hacerlo usando `setAttribute()`, ya que solo funciona sobre Objetos (no listas). Por eso, deberás recorrerlo usando un loop.

Para eso, sigue estos pasos:

- Guarda la lista de nodos en una Variable.

```
let todosLosLinks = document.querySelectorAll("a");
```

- Recorre la Variable con un loop, de modo tal que, en cada vuelta, se modifique con `setAttribute`.

```
for (var i = 0; i < todosLosLinks.length; i++) {  
  todosLosLinks[i].setAttribute("href", "http://plataforma5.iicap.cl/")  
};
```


IMPORTANTE

Es una convención usar las letras "i" o "j" para los contadores de los Loops.



Crear, agregar y remover elementos

Con `createElement` se crea un Objeto que luego agregaremos al DOM.

```
let elemento = document.createElement("elemento");
```

Pasamos un String como parámetro, que indica qué etiqueta queremos crear.

Por ejemplo, primero debemos guardar su referencia en una Variable:

```
let h1 = document.createElement("h1");
```

Luego, le podemos agregar texto usando `textContent`:

```
h1.textContent = "Plataforma 5 Chile";
```

Por último, podemos crear un elemento `div`:

```
let div = document.createElement("div");
```

Crear, agregar y remover elementos

Para agregar un elemento dentro de otro usaremos el método `appendChild`. De esta forma, podemos incluir en un nodo un nuevo hijo.

Veamos su sintaxis:

```
document.elementoPadre.appendChild(elementoHijo);
```

Importante: También puede usar este método para mover un elemento, de un elemento a otro.

- La Variable `div` debe llamar al método `appendChild`.
- Para ello, el argumento del método debe ser el elemento a agregar (en este caso, `h1`).

```
div.appendChild(h1); // En este caso, h1 es el Objeto  
creado anteriormente. Por eso no es un String (como en  
el caso de document.createElement).
```

- Luego, agregá el `div` al `body`, pasándolo como argumento:

```
document.body.appendChild(div);
```

Si quisiéramos eliminar el `div` anterior deberíamos usar:

```
document.body.removeChild(div);
```

MÓDULO 2

+

o

.

GRACIAS

Aquí finaliza la clase n°7 del módulo 2

OTEC PLATAFORMA 5 CHILE