



DOCUMENT OBJECT MODEL (DOM) Y SELECTORES

DOCUMENT OBJECT MODEL (DOM)

DOCUMENT OBJECT MODEL (DOM)

El **DOM** surgió a partir de la implementación de JavaScript en los navegadores.

El **DOM** es un **modelo de documento** que **representa al documento** HTML de manera virtual como un árbol de elementos (nodos), en donde cada elemento (nodo) representa una parte del documento.

El **DOM** no es directamente **el HTML**. Es una **representación virtual** del documento HTML que recibe el navegador. Es una construcción hecha por javascript tomando como base el HTML recibido.

El **DOM** permite ejecutar código en el navegador para acceder e interactuar con cualquier elemento del documento.

Los elementos (nodos) pueden crearse, quitarse o modificarse.

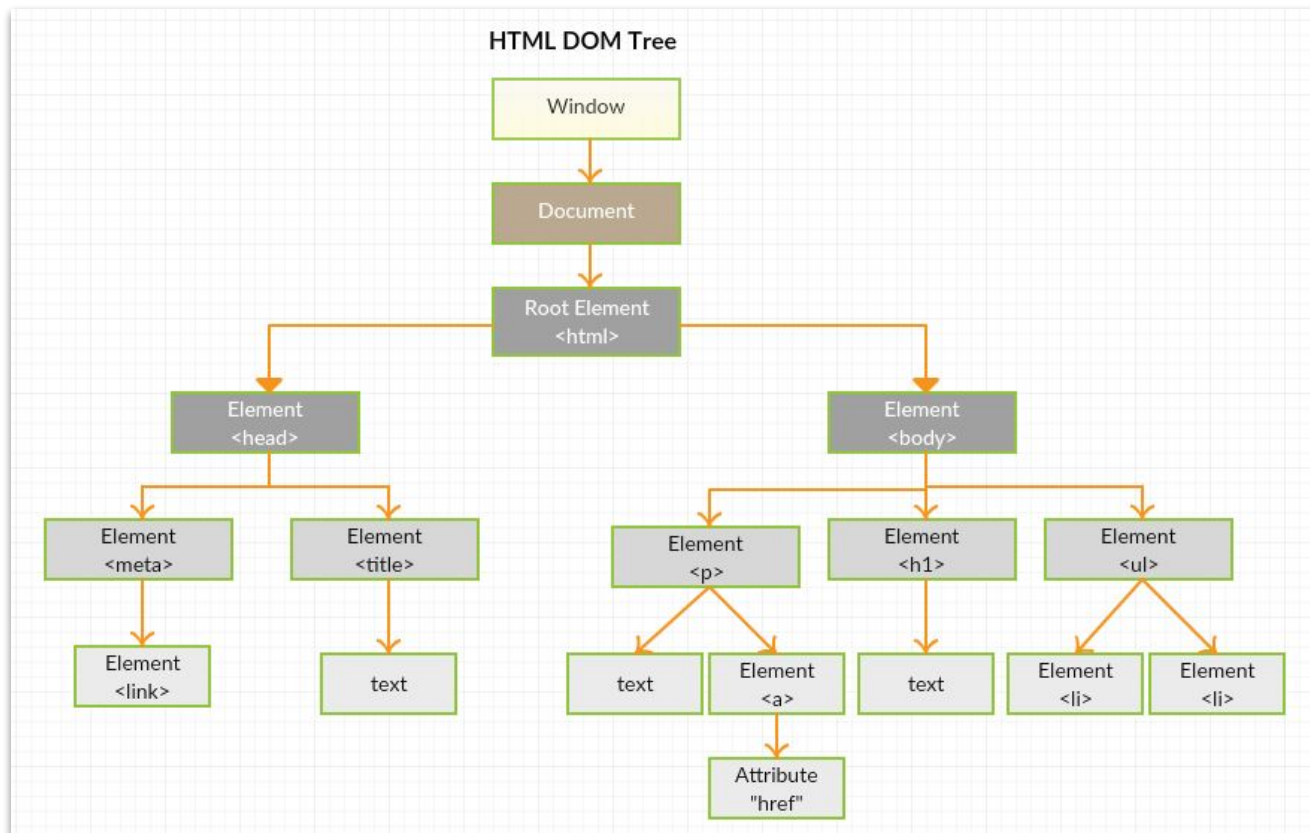
Sobre los elementos (nodos) también pueden añadirse manejadores de eventos (event listeners en inglés) que se ejecutarán cuando ocurra el evento indicado en el manejador.

¿Qué podemos hacer utilizando el DOM?

- Modificar elementos, atributos y estilos de una página.
- Borrar cualquier elemento o atributo.
- Agregar nuevos elementos o atributos.
- Reaccionar a todos los eventos HTML de la página.
- Crear nuevos eventos HTML en la página.

Recordemos que el **DOM** es una **representación virtual** del documento **HTML** por lo tanto todo lo que hagamos **afectará** la forma en que el documento se **muestra** frente al usuario pero **no modifica el documento HTML original**.

DOCUMENT OBJECT MODEL (DOM)



window

El objeto *window* es lo **primero** que se carga en el navegador.

Como todo objeto en javascript cuenta con propiedades y métodos. Podemos inspeccionarlo desde la consola del navegador escribiendo la palabra *window*.

document

El objeto *document* representa al **html** cargado en el navegador.

document es cargado dentro del objeto *window* y también podemos ver todas sus propiedades y métodos desde la consola del navegador escribiendo la palabra *document*.

SELECTORES



DigitalHouse >
Coding School

/University
/Schools
/Corporate Training
/Academy

Para capturar elementos del **DOM** usamos **selectores**.

Los selectores permiten capturar **un elemento o un grupo de elementos**.

querySelector();

querySelector() permite capturar **un solo elemento** del **DOM**. Recibe un string que indica el elemento del **DOM** a capturar.

Para identificar al elemento usamos la misma notación que en CSS.

querySelector() retorna el elemento capturado.

```
{
```

```
let titulo = document.querySelector(".mainTitle");
```



Declaramos una variable para almacenar el dato que retorna el selector para poder utilizarlo luego en otra parte de nuestro programa.

querySelectorAll();

querySelectorAll() permite capturar **un grupo de elementos** del **DOM**. Recibe un string que con el selector de los elementos a capturar.

Para identificar al elemento usamos la misma notación que en CSS.

querySelectorAll() retorna un array con todos los elementos capturados.

```
{
```

```
let textos = document.querySelectorAll("p");
```



Declaramos una variable para almacenar el dato que retorna el selector para poder utilizarlo luego en otra parte de nuestro programa.

getElementById();

getElementById() permite capturar **un solo elemento** del **DOM**. Recibe un string que con el id del elemento a capturar.

getElementById() retorna el elemento capturados.

```
{
```

```
let marca = document.getElementById("marca");
```



getElementById() fue un método de selección muy utilizado hasta la aparición de *querySelector()* y *querySelectorAll()*.

GRACIAS