

DESARROLLO WEB FULL STACK

Introducción al DOM

Document Object Model

DWFS COR



DOM

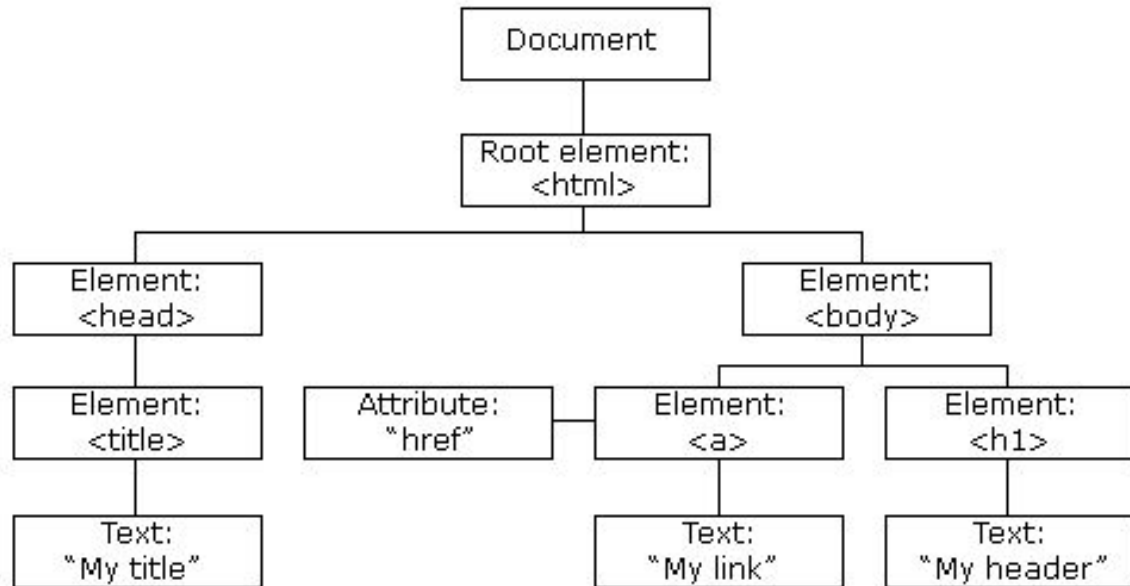
El modelo de objeto de documento (en inglés DOM) es una interfaz de programación (API) para los documentos HTML.

Facilita una representación estructurada del documento y define de qué manera los programas pueden acceder, al fin de modificar, tanto su estructura, estilo y contenido.

El DOM da una representación del documento como un grupo de nodos y objetos estructurados que tienen propiedades y métodos. Esencialmente, conecta las páginas web a scripts o lenguajes de programación.



Árbol del documento



Nodos

Llamamos **nodo** a cada elemento individual en el árbol del documento, los tipos de nodos más frecuentemente usados son:

- 1. document:** el nodo document es el nodo raíz, a partir del cual derivan el resto de nodos.
- 2. element:** son los nodos definidos por etiquetas html. Por ejemplo una etiqueta div genera un nodo. Si dentro de ese div tenemos tres etiquetas p, dichas etiquetas definen nodos hijos de la etiqueta div.



Nodos

3. text: el texto dentro de un nodo element se considera un nuevo nodo hijo de tipo text (texto). Los navegadores también crean nodos tipo texto sin contenido para representar elementos como saltos de línea o espacios vacíos.

4. attribute: los atributos de las etiquetas definen nodos, aunque trabajando con JavaScript no los veremos como nodos, sino que lo consideramos información asociada al nodo de tipo element.

5. comentarios y otros: los comentarios y otros elementos como las declaraciones doctype en cabecera de los documentos HTML generan nodos.



Propiedades de Nodo	Métodos de Nodo
childNodes	appendChild()
firstChild	cloneNode()
lastChild	compareDocumentPosition()
nextSibling	contains()
nodeName	hasChildNodes()
nodeType	insertBefore()
nodeValue	isEqualNode()
parentNode	removeChild()
previousSibling	replaceChild()



Métodos del document	Propiedades de HTML*Element
document.createElement()	innerHTML
document.createTextNode()	outerHTML
document.getElementsByTagName()	textContent
document.querySelector()	innerText
document.getElementById()	children
	firstElementChild
	lastElementChild
	nextElementChild
	previousElementChild



Window

El objeto **window** representa la ventana que contiene un documento DOM; la propiedad **document** apunta al DOM document cargado en esa ventana. El objeto window al que pertenece un documento puede ser obtenido usando la propiedad document.defaultView.

[Propiedades](#)



A photograph of two men in an office environment. The man in the foreground, on the right, is wearing a striped polo shirt and is pointing his right index finger towards a computer screen. He has dark hair and is looking intently at the screen. Behind him, another man with a beard and glasses is also looking at the screen. The background is slightly blurred, showing office furniture and plants. The entire image has a blue and purple color overlay.

¿Cómo empiezo
a utilizar DOM?



Encontrando elementos

Esto devuelve una colección de nodos p.

```
var parrafos = document.getElementsByTagName('p');  
console.log(parrafos);
```



Viendo atributos

Devuelve un objeto con los atributos del nodo.

```
var parrafos = document.getElementsByTagName('p');  
console.log(parrafos[0].attributes);
```



Modificando atributos

Modifica o agrega un atributo de estilo.

```
var parrafos = document.getElementsByTagName('p');  
parrafos[0].style.color = 'red';
```



Interfaces esenciales de DOM



Interfaces

- `document.getElementById(id)`
- `document.querySelector(selector)`
- `element.getElementsByTagName(name)`
- `document.createElement(name)`
- `parentNode.appendChild(node)`
- `element.innerHTML`
- `element.style.left`
- `element.setAttribute`
- `element.getAttribute`
- `element.addEventListener`
- `window.onload`
- `window.scrollTo`



A photograph of two men in an office setting, looking at a computer screen. The man in the foreground is pointing at the screen. The image has a blue overlay. The text "Empecemos a jugar con DOM" is written in white on the left side.

Empecemos a
jugar con DOM



Seleccionar elementos del documento
y cambiarles sus atributos.

