

Antes de adentrarte en frameworks o librerías como React, Vue, Angular, Svelte, etc. Debes saber que es el DOM. DOM Significa Document Object Model, y es un concepto fundamental en la programación web, tan fundamental que si no lo entiendes bien, vas a andar por ahí usando muy mal tecnologías complejas y dejando sufrimiento y lagrimas a tu paso, no solo para ti si no para los que te rodean, en fin.

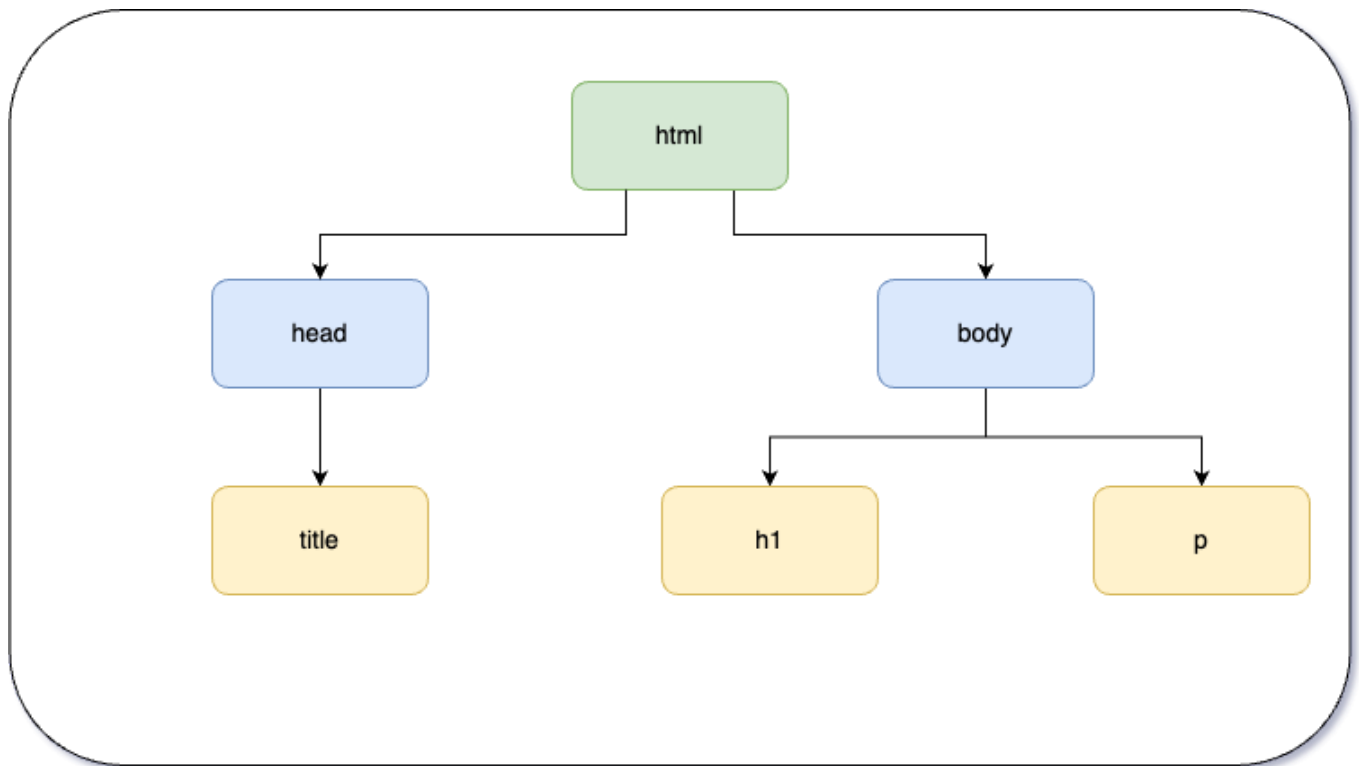
Para entender bien el concepto primero vamos construir un documento HTML muy simple.

```
<!DOCTYPE html>
<html>
<head>
  <title>Mi Página</title>
</head>
<body>
  <h1>iHola Mundo!</h1>
  <p>Bienvenidos a mi página web.</p>
</body>
</html>
```

Como podrás notar ese HTML tiene una estructura definida, esa estructura nos da la posibilidad de manipular su contenido porque ya sabemos que podemos encontrarnos en cada una de sus partes, la representación de ese concepto es el DOM.

En palabras más técnicas el DOM es una representación programática de un documento HTML o XML y es gracias a el que podemos manipular y gestionar lo que sucede dentro de ese documento.

Adicionalmente hay que entender que el DOM se organiza a través de una jerarquía que es dictada por la estructura HTML o XML del documento, por ejemplo el HTML anterior se podría traducir a nivel del DOM a algo como lo siguiente:



Documentos HTML o XML complejos.

Es necesario aclarar que esta estructura aplica de la misma manera para documentos HTML o XML más complejos ya que podemos definir atributos a cada etiqueta, incluso podemos añadir atributos personalizados y tratarlos de una manera diferente, sin embargo esto no afecta el concepto antes explicado, al final el documento HTML o XML que estemos creando puede ser procesado y gestionado en el DOM.

¿Cómo interactuar con el DOM?

Si estamos trabajando con JavaScript este lenguaje de programación ya lo soporta de forma nativa, te dejo algunos ejemplos simples:

```
// Seleccionar por ID
const titulo = document.getElementById('mi-titulo');

// Seleccionar por clase
const parrafos = document.getElementsByClassName('mi-clase');
```

```
// Seleccionar por etiqueta
const divs = document.getElementsByTagName('div');

// Seleccionar usando selectores de CSS
const primerParrafo = document.querySelector('p');
const todosLosParrafos = document.querySelectorAll('p');
```

Cada uno de esos métodos está adaptado para buscar elementos a través de atributos o particularidades específicas, además de hacer búsquedas podemos también manipular lo que sucede:

```
// Cambiar el contenido de texto
titulo.textContent = 'Nuevo Título';

// Cambiar el contenido HTML
primerParrafo.innerHTML = '<strong>Texto en negrita</strong>';

// Cambiar atributos
titulo.setAttribute('class', 'nuevo-clase');
primerParrafo.removeAttribute('class');

// Cambiar estilos
titulo.style.color = 'blue';
primerParrafo.style.fontSize = '20px';
```

Y así como el ejemplo anterior podemos también agregar elementos de forma dinámica, en fin. Son muchas posibilidades y la mayoría de frameworks front-end de JavaScript hacen justamente manipular el DOM, de hecho, el reto principal de muchos framework es gestionar el DOM de forma correcta y optima.

Así de compleja puede ser la gestión del DOM, por eso es importante que sea un concepto que entiendas antes de adentrarte en conceptos más complejos.