
SENPAI
ACADEMY

DOM 

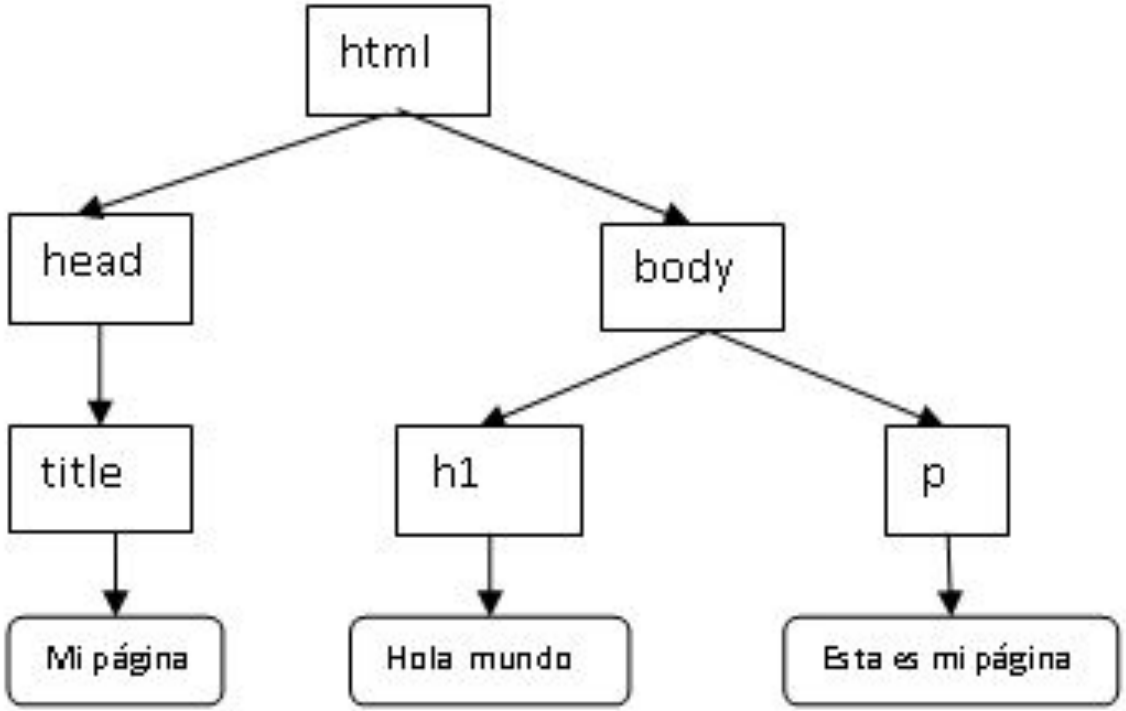
SENPAI
ACADEMY

DOM - Intro

Las siglas **DOM** significan **Document Object Model**, o lo que es lo mismo, la estructura del documento HTML. Una página HTML está formada por **múltiples etiquetas HTML**, anidadas una dentro de otra, formando un árbol de etiquetas relacionadas entre sí, que se denomina **árbol DOM** (o simplemente DOM).

En Javascript, cuando nos referimos al **DOM** nos referimos a esta estructura, que podemos **modificar de forma dinámica** desde Javascript, añadiendo nuevas etiquetas, modificando o eliminando otras, cambiando sus atributos HTML, añadiendo clases, cambiando el contenido de texto, etc...

DOM - Intro



SENPAI ACADEMY

DOM - El objeto document

En Javascript, la forma de acceder al DOM es a través de un objeto llamado **document**, que r**epresenta el árbol DOM** de la página o pestaña del navegador donde nos encontramos.

Veremos que Javascript nos proporciona un conjunto de herramientas y **métodos** para trabajar de forma nativa con el **DOM** de la página.

SENPAI
ACADEMY

Seleccionar elementos 🔍

(Tradicional)

SENPAI
ACADEMY

DOM - Seleccionar elementos (Tradicional)

Si nos encontramos en nuestro código Javascript y queremos hacer modificaciones en un elemento de la página HTML, lo primero que debemos hacer es **buscar dicho elemento**. Para ello, se suele intentar identificar el elemento a través de alguno de sus atributos más utilizados, generalmente el **id** o la **clase**.

Existen varios métodos, los más clásicos y tradicionales para realizar búsquedas de elementos en el documento. Observa que si lo que buscas es un elemento específico, lo mejor sería utilizar **getElementById()**, en caso contrario, los 3 siguientes métodos, nos devolverá un Array donde tendremos que elegir el elemento en cuestión posteriormente.

SENPAI
ACADEMY

DOM - Seleccionar elementos (Tradicional)

```
//Busca el elemento HTML con el id id. Si no, devuelve null.
const form = document.getElementById('contacto');

//Busca elementos con la clase tarea. Si no, devuelve []
const tareas = document.getElementsByClassName('tarea');

//Busca elementos con atributo name 'email'. Si no, devuelve [].
const emails =  document.getElementsByName('email');

//Busca elementos <p>. Si no encuentra ninguno, devuelve [].
const parrafos = document.getElementsByTagName('p');
```

SENPAI ACADEMY

DOM - Seleccionar elementos (Tradicional)

Estos son los **4 métodos tradicionales** de Javascript para manipular el DOM.

Se denominan tradicionales porque son los que existen en Javascript desde versiones más antiguas.

Dichos métodos te permiten buscar elementos en la página dependiendo de los atributos **id**, **class**, **name** o de la **propia etiqueta,** respectivamente.

SENPAI ACADEMY

Seleccionar elementos 🔍

(Moderno)

SENPAI
ACADEMY

DOM - Seleccionar elementos (Moderno)

Aunque podemos utilizar los **métodos tradicionales** que acabamos de ver, actualmente tenemos a nuestra disposición dos nuevos métodos de búsqueda de elementos que son mucho más cómodos y prácticos si conocemos y dominamos los **selectores CSS**.

Es el caso de los métodos **.querySelector()** y **.querySelectorAll()**.

Con estos dos métodos podemos realizar todo lo que hacíamos con los métodos tradicionales mencionados anteriormente e incluso muchas más cosas (en menos código), ya que son **muy flexibles** y **potentes** gracias a los **selectores CSS**.

SENPAI
ACADEMY

DOM - Seleccionar elementos (Moderno)

```
//Busca el primer elemento que coincide
//con el selector CSS #tareas. Si no, null.
const tareasPadre = document.querySelector('#tareas');

//Busca todos los elementos que coinciden
//con el selector CSS li.tarea. Si no, [].
const tareas = document.querySelectorAll('li.tarea');
```

SENPAI ACADEMY

Manipular clases 

SENPAI
ACADEMY

DOM - manipular clases

En CSS es muy común utilizar múltiples clases CSS para asignar estilos relacionados dependiendo de lo que queramos.

Javascript tiene a nuestra disposición una propiedad **.className** en todos los elementos HTML. Dicha propiedad contiene el valor del **atributo HTML class.**

Si accedemos a **.classList**, nos devolverá un (lista) de clases CSS de dicho elemento. Pero además, incorpora una **serie de métodos ayudantes** que nos harán muy sencillo trabajar con clases CSS.

SENPAI
ACADEMY

DOM - manipular clases

```
const div = document.querySelector("#page");

div.classList; // ["info", "data", "dark"]

div.classList.add("uno", "dos"); // No devuelve nada.
div.classList; // ["info", "data", "dark", "uno", "dos"]

div.classList.remove("uno", "dos"); // No devuelve nada.
div.classList; // ["info", "data", "dark"]
```

SENPAI ACADEMY

Reemplazar contenido 

SENPAI
ACADEMY

DOM - Reemplazar contenido

Utilizando las siguientes propiedades, se trata de una vía rápida con la cuál podemos añadir (o más bien, reemplazar) el **contenido de una etiqueta HTML.**

La propiedad .**textContent** nos devuelve el contenido de **texto** de un elemento HTML. Es útil para obtener (o modificar) sólo el texto dentro de un elemento, obviando el etiquetado HTML.

Por otro lado, la propiedad **.innerHTML** nos permite hacer lo mismo, pero interpretando el código HTML indicado y renderizando sus elementos

SENPAI
ACADEMY

DOM - Reemplazar contenido

```
const div = document.querySelector(".info"); // <div class="info"></div>

div.innerHTML = "<strong>Importante</strong>"; // Interpreta el HTML
div.innerHTML;  // "<strong>Importante</strong>"
div.textContent;  // "Importante"

div.textContent = "<strong>Importante</strong>";  // No interpreta el HTML
```

SENPAI
ACADEMY

LINKS 🔗

SENPAI
ACADEMY

LINKS

- DOM
  - https://developer.mozilla.org/en-US/docs/Web/API/HTML_DOM_API
  - https://lenguajejs.com/javascript/dom/que-es/
  - https://es.wikipedia.org/wiki/Document_Object_Model
  - https://lenguajejs.com/javascript/dom/seleccionar-elementos-dom/
  - https://lenguajejs.com/javascript/dom/manipular-clases-css/
  - https://lenguajejs.com/javascript/dom/insertar-elementos-dom/

SENPAI
ACADEMY

SENPAI
ACADEMY

gustavguez

gustavguez

gustavguez

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE