



MANEJO DEL DOM

# POR AMOR AL CÓDIGO

Gerónimo Riesco

# MANIPULACIÓN DEL DOM EN JAVASCRIPT

## TEMAS QUE SE ABORDARÁN

¿Qué es el DOM?

Acceso a elementos del DOM

Acceder a hijos/padres de un elemento

Crear nuevos elementos en el DOM

Añadir elementos al DOM

Añadir/eliminar/modificar Clases

# ¿Qué es el DOM?

**El DOM da una representación del documento como un grupo de nodos y objetos estructurados que tienen propiedades y métodos. En resumen, es la representación de la página web en la memoria del navegador, a la que podemos acceder a través de JavaScript. El DOM es un árbol donde cada nodo es un objeto con todas sus propiedades y métodos que nos permiten modificarlo. Estas son algunas funciones que nos permiten acceder y modificar los elementos del DOM:**



# ACCESO A ELEMENTOS DEL DOM

---


```
// Obtiene un elemento por id
document.getElementById('someid');

// Obtinee una lista con los elementos que tienen esa clase
document.getElementsByClassName('someclass');

// Obtiene una HTMLCollection con los todos los elementos 'li'
document.getElementsByTagName('LI');

// Devuelve el primer elemento del documento que cumpla la selección
// (la notación es como en CSS)
document.querySelector('.someclass');

// Devuelve una lista de elementos que cumplen con la selección (notación como en CSS)
document.querySelectorAll('div.note, div.alert');
```



# ACCEDER A HIJOS/PADRES DE UN ELEMENTO

```
// Obtener los hijos de un elemento
var elem = document.getElementById('someid');
var hijos = elem.childNodes;

// Su nodo padre
var padre = elem.parentNode;
```

# CREAR NUEVOS ELEMENTOS EN EL DOM

```
// Para crear elementos llamamos a createElement con el nombre del elemento
var nuevoH1 = document.createElement('h1');
var nuevoParrafo = document.createElement('p');

// Crear nodos de texto para un elemento
var textoH1 = document.createTextNode('Hola mundo!');
var textoParrafo = document.createTextNode('lorem ipsum...');

// Añadir el texto a los elementos
nuevoH1.appendChild(textoH1);
nuevoParrafo.appendChild(textoParrafo);

// también podemos asignar directamente el valor a la propiedad innerHTML
nuevoH1.innerHTML = textoH1
nuevoParrafo.innerHTML = textoParrafo

// los elementos estarían listos para añadirlos al DOM, ahora mismo solo existen en memoria, pero no serán visibles hasta que no los añadamos a un elemento del DOM
```

# AÑADIR ELEMENTOS AL DOM

```
// seleccionamos un elemento
var cabecera = document.getElementById('cabecera');

// Añadir elementos hijos a un elemento
cabecera.appendChild(nuevoH1);
cabecera.appendChild(nuevoParrafo);

// También podemos añadir elementos ANTES del elemento seleccionado

// Tomamos el padre
var padre = cabecera.parentNode;

// Insertamos el h1 antes de la cabecera
padre.insertBefore(nuevoH1, cabecera);
```

También podemos añadir directamente un trozo de HTML antes o después de un elemento del DOM, supongamos que tenemos estos elementos en la página:

```
<div id='box1'>
  <p>aquí algo de texto</p>
</div>
<div id='box2'>
  <p>otro parrafo bla bla bla</p>
</div>
```

Podemos hacer:

```
var box2 = document.getElementById('box2');
box2.insertAdjacentHTML('beforebegin', '<div><p>un parrafo nuevo.</p>
</div>');

// beforebegin - El nuevo HTML es insertado justo antes del elemento,
a la misma altura (hermano).
// afterbegin - El nuevo HTML se inserta dentro del elemento, antes de
l primer hijo.
// beforeend - El nuevo HTML se inserta dentro del elemento, después d
el último hijo.
// afterend - El nuevo HTML es insertado justo después del elemento, a
la misma altura (hermano).
```



# Añadir/eliminar/modificar Clases

```
// Tomamos un elemento
var cabecera = document.getElementById('cabecera');

// elimina una clase del elemento
cabecera.classList.remove('foo');

// Añade una clase si no existe
cabecera.classList.add('otra');

// añade o elimina varias clases a la vez
cabecera.classList.add('foo', 'bar');
cabecera.classList.remove('foo', 'bar');

// Si la clase existe la elimina, si no existe, la crea
cabecera.classList.toggle('visible');

// Devuelve true si el elemento contiene esa clase
cabecera.classList.contains('foo');
```



# GITHUB: GERONIMO-RIESCO



Si quieres comentar, corregir algo o contribuir, puedes hacerlo en Github.