

2. DOM I

El DOM conecta el HTML y CSS del documento con JavaScript representando el HTML mediante un sistema de nodos en árbol con una serie de propiedades y métodos que podemos usar para modificar la estructura, estilo y contenido.

Para acceder a todas estas propiedades y métodos usaremos el objeto **document**.

Seleccionando un nodo

Podemos usar los métodos **document.getElementById()** y **document.querySelector()** para seleccionar elementos singulares en el DOM.

```
// <h1 id="title">Título de la página</h1>
const title = document.getElementById('title');

const sameTitle = document.querySelector('#title');

// Esto modifica el estilo del elemento estableciendo la propiedad de CSS color.
title.style.color = 'red';
```

Este método para seleccionar elementos nos devuelve un objeto de tipo **Node** que representa una parte del árbol del DOM.

Seleccionando varios nodos

Podemos usar los métodos **getElementsByClassName**, **getElementsByName**, y **querySelectorAll** para seleccionar varios elementos al mismo tiempo.

```
// Selecciona todos los elementos con la class ".important".
const importantElements = document.getElementsByClassName('important');

// Selecciona todos los encabezados de segundo nivel.
const h2s = document.getElementsByTagName('h2');

// Selecciona todos los párrafos hijos directos de un section.
const ps = document.querySelectorAll('section > p');
```

Este método de seleccionar elementos nos devuelve un objeto de tipo **NodeList** que se parece a un Array y podremos manipular.

```
const ps = document.querySelectorAll('section > p');

// Recorre todos los párrafos y establece la propiedad de CSS "background-color" a
"red".
for (const p of ps) {
  p.style.backgroundColor = 'red';
}
```

Moviéndose por el DOM

El árbol del DOM está formada por nodos. Estos pueden ser de varios tipos y todos comparten una serie de características. El más importante es el nodo de tipo **Element** que representa un elemento HTML (un elemento con tag). Cuando seleccionamos un único nodo o una lista de nodos podemos acceder a varias propiedades que se refieren a los nodos hijos, padres, y a los nodos hermanos.

Aquí (https://developer.mozilla.org/en-US/docs/Web/API/Node/nodeType#Node_type_constants) podéis ver los diferentes tipos de nodos. Además, podemos acceder a su tipo usando la propiedad **_nodeType_** de cualquier nodo que tengamos seleccionado.

Todos los nodos tienen una serie de propiedades que permiten acceder a nodos padres, hijos y hermanos:

```
const intro = document.querySelector('p.intro');

intro.childNodes; // Todos los nodos hijos.
intro.firstChild; // Primer nodo hijo.
intro.lastChild; // Último nodo hijo.
intro.previousSibling; // Nodo hermano anterior.
intro.nextSibling; // Nodo hermano siguiente.
intro.parentNode; // Nodo padre.
```

Si queremos evitar seleccionar todos los tipos de nodos y queremos sólo considerar los nodos de tipo **Element**:

```
const intro = document.querySelector('p.intro');

intro.children; // Elementos hijos.
intro.firstElementChild; // Primer elemento hijo.
intro.lastElementChild; // Último elemento hijo.
intro.previousElementSibling; // Elemento hermano previo.
intro.nextElementSibling; // Elemento hermano siguiente.
intro.parentElement; // Elemento padre.
```

Modificando el texto de los elementos

Podemos modificar el texto de un elemento usando la propiedad **textContent**:

```
// Imprime en la consola el contenido textual del elemento incluyendo el texto de los elementos hijos.  
const el = document.querySelector('p.intro');  
  
// Modifica el contenido de texto del elemento, reemplazando todo el contenido.  
el.textContent = 'lorem ipsum';  
  
console.log(el.textContent);
```

Esta propiedad vale tanto para nodos de tipo genérico **Node** como de tipo específico **Element**. Para los de tipo **Element** también existe la propiedad **innerHTML** pero hay una serie de diferencias con **textContent** que puedes ver en [este enlace](https://developer.mozilla.org/en-US/docs/Web/API/Node/textContent#Differences_from_innerText) (https://developer.mozilla.org/en-US/docs/Web/API/Node/textContent#Differences_from_innerText).

Modificando el HTML de los elementos

La propiedad **textContent** establece el texto puro de los elementos y no puede establecer nuevo HTML. Para ver y establecer el contenido HTML usamos la propiedad **innerHTML** que obviamente sólo se aplica a nodos de tipo **Element**:

```
const el = document.querySelector('section#principal');  
  
// Imprime en la consola el contenido HTML del elemento.  
console.log(el.innerHTML);  
  
// Modifica el HTML del elemento.  
el.innerHTML = '<p>lorem ipsum</p>';
```

La propiedad **innerHTML** nos muestra el HTML del propio elemento incluido el elemento mismo.

Completado. Continuamos.

