

DOM - Document Object Model

Para que JavaScript pueda interpretar y manipular el HTML necesita convertir cada etiqueta a una serie de **NODOS** que él pueda entender y gestionar, ese grupo de **NODOS** son objetos y el conjunto de todos los objetos de nuestro documento forman el **DOM** (Modelo de Objetos del documento)

```
<h1 id="main-title-title" class="title--big">Soy un título</h1>
```

Esto es un fragmento de cómo interpreta JavaScript nuestra etiqueta <h1>

```
const title = {  
  children: [],  
  classList: ['title', 'title--big'],  
  id: 'main-title',  
  tagName: 'h1',  
  textContent: 'Soy un título'  
};
```

DOM - Document Object Model

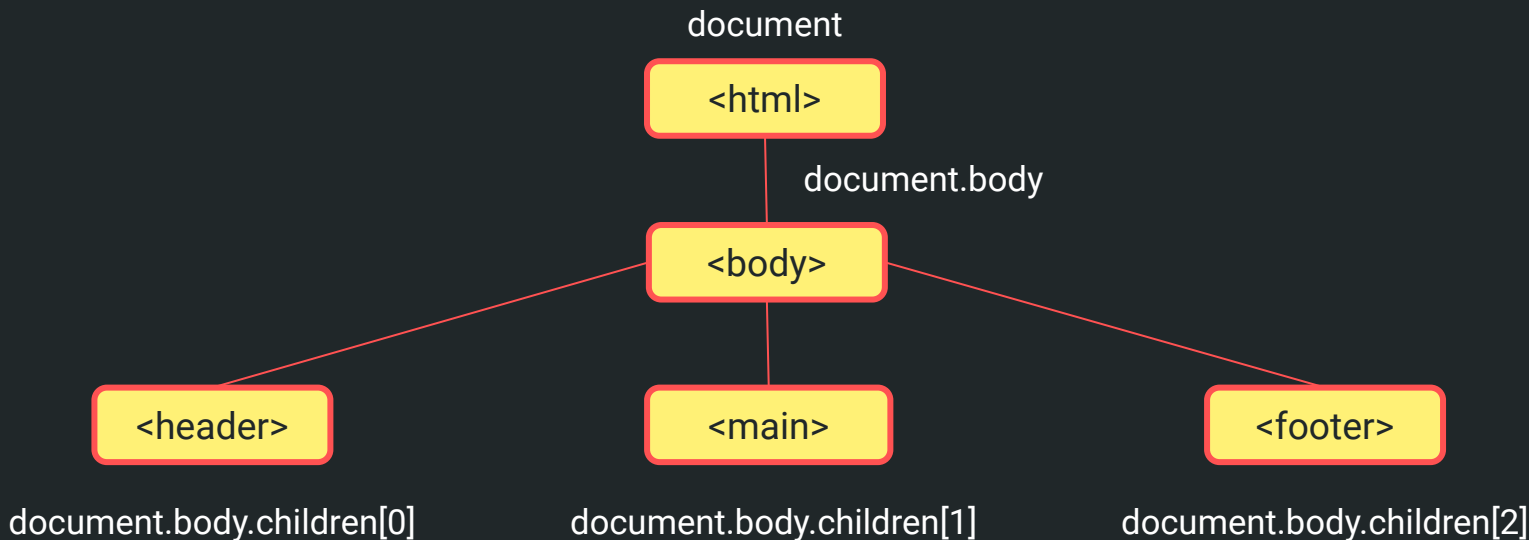
Si tenemos un conjunto de etiquetas más grande el grupo de objetos aumenta, pero sigue siendo la misma estructura.

```
<header class="header">
  <h1 id="main-title" class="title title--big">Soy un título</h1>
  <div class="banner">
    
  </div>
</header>
```

```
const header = {
  tagName: 'header',
  classList: ['header'],
  children: [
    {
      children: [],
      classList: ['title', 'title--big'],
      id: 'main-title',
      tagName: 'h1',
      textContent: 'Soy un título'
    },
    {
      tagName: 'div',
      classList: ['banner'],
      children: [
        {
          tagName: 'img',
          src: 'assets/images/banner.jpg',
          alt: 'Imagen de banner'
        }
      ]
    }
  ]
};
```

DOM - Document Object Model

El esquema que genera JavaScript es una estructura de árbol.





DOM - Selección de elementos



Para acceder a los elementos de forma directa hay varios métodos que nos da el objeto document, todos estos métodos se escriben con `document.método()` los que más se utilizan son:

getElementById(id): este método recibe un string que deberá coincidir con **UN SOLO** elemento del DOM. Devuelve ese elemento.

querySelector(selectorCSS): este método recibe un string y devolverá el **PRIMER ELEMENTO** que coincida con el selector CSS que le pasemos.

querySelectorAll(selectorCSS): este método recibe un string y devolverá **TODOS** los elementos que coincidan con el selector CSS que le pasemos.

Para poder movernos por esa estructura y editar el contenido de las etiquetas, el navegador nos da una serie de herramientas para facilitarnos la vida. Todas estas propiedades pertenecen a los elementos del DOM por lo que siempre será `element.propiedad`

`parentElement`: Nos permite seleccionar al padre del elemento desde el que partimos.

`nextElementSibling`: Nos permite seleccionar al hermano siguiente del elemento

`previousElementSibling`: Nos permite seleccionar al hermano anterior del elemento

`children`: Es una propiedad, por lo que no lleva paréntesis. En esta propiedad tenemos un “array” con los hijos del elemento.

`textContent`: Es una propiedad que tiene el contenido del elemento seleccionado.