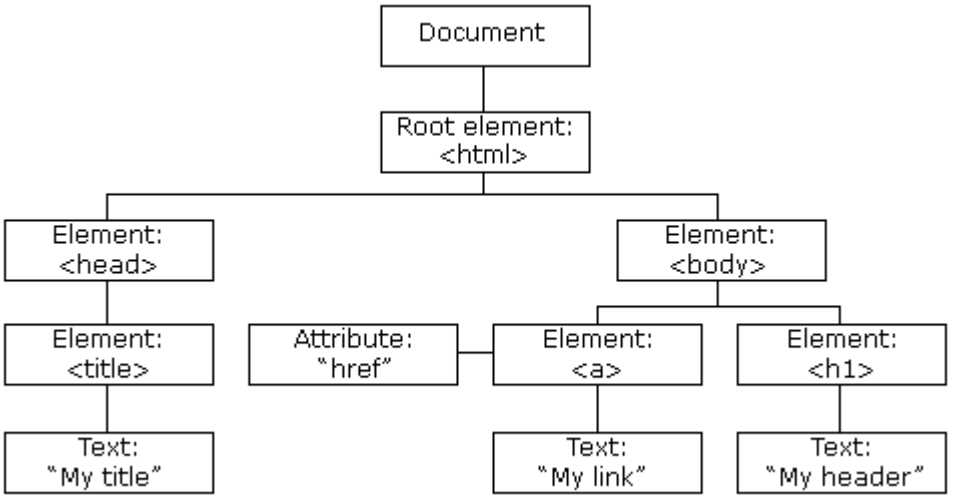
JavaScript

DOM

El árbol de objetos HTML DOM



El Modelo de Objetos de Documento (DOM) del W3C es una plataforma y una interfaz neutral en cuanto al idioma que permite a los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de un documento. En definitiva, HTML DOM es un estándar sobre cómo obtener, cambiar, agregar o eliminar elementos HTML.

El estándar W3C DOM se divide en 3 partes diferentes:

Core DOM: modelo estándar para todo tipo de documentos
XML DOM: modelo estándar para documentos XML
HTML DOM: modelo estándar para documentos HTML

La interfaz de programación son las propiedades y métodos de cada objeto.

Una propiedad es un valor que puede obtener o establecer (como cambiar el contenido de un elemento HTML).

Un método es una acción que puedes realizar (como agregar o eliminar un elemento HTML).

El siguiente ejemplo cambia el contenido (el innerHTML) del <p>elemento con id="demo":

```
<html>
    <body>
        <p id="demo"></p>
        <script>
            document.getElementById("demo").innerHTML = "Hello World!";
        </script>
    </body>
</html>
```

En el ejemplo anterior, getElementByIdes un **método** , mientras que innerHTML es una **propiedad**.

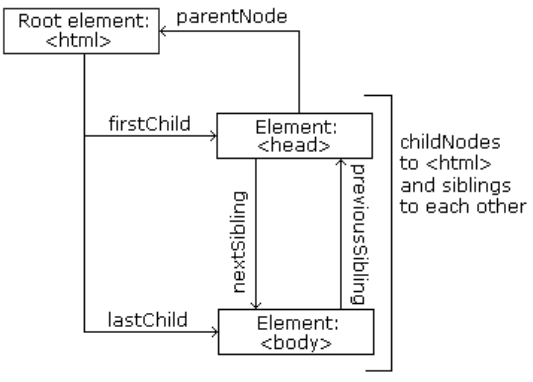
Volvamos a la navegabilidad. Tenemos el documento:

```
<html>

  <head>
    <title>DOM Tutorial</title>
  </head>

  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>

</html>
```



En el HTML anterior puedes leer:

<html> es el nodo raíz
<html> no tiene padres
<html> es el padre de <head> y <body>
<head> es el primer hijo de <html>
<body> es el último hijo de <html>
y:

<head> tiene un hijo: <title>
<title> tiene un hijo (un nodo de texto): "DOM Tutorial"
<body> tiene dos hijos: <h1> y <p>
<h1> tiene un hijo: "DOM Lesson one"
<p> tiene un hijo: "¡Hello world!"
<h1> y <p> son hermanos

Puede utilizar las siguientes propiedades de nodo para navegar entre nodos con JavaScript:

- parentNode
- childNodes[nodenumber]
- firstChild
- lastChild
- nextSibling
- previousSibling

Por ejemplo el título puede ser accedido de esta manera:

myTitle = document.getElementById("demo").**firstChild**.nodeValue;

o de esta;

myTitle = document.getElementById("demo")**.innerHTML**;

o esta:

myTitle = document.getElementById("demo").**childNodes[0].nodeValue**;

Buscar elementos

Las formas de acceder a un elemento son:

- document.**getElementById(id)** → Busca elemento por id
- document.getElementsByTagName(name) → Busca elemento por tag
- document.getElementsByClassName(name) → Busca elemento por clase

Ejemplo: dado el párrafo:

Hola

- document.**getElementById(**main_section**)**
- document.**getElementsByTagName(**p**)**
- document.**getElementsByClassName(**main_paragraph**)**

Otras formas de acceder

Modificar elementos

Propiedades

element.innerHTML = nuevo valor → cambiar el html interno de un elemento.
element.attribute = nuevo valor → cambia el valor del atributo
element.style.property = nuevo estilo →cambia el estilo

Método

element.setAttribute(attribute, value) Cambia el valor del atributo de un elemento HTML

Agregar y eliminar elementos

document.createElement(element) → **Crea** un elemento en un documento HTML
document.removeChild(element) → **Borra** un elemento en un documento HTML
document.appendChild(element) → **Añadir** un elemento en un documento HTML
document.replaceChild(new, old) → **Reemplaza** un elemento en un doc. HTML
document.write(text) → **Escribir** en el flujo de salida HTML