



6.A. Modelo de objetos del documento en JavaScript

1.- Bases del Modelo de Objetos del Documento (DOM).

1.3.- Acceso a los nodos.

Cuando ya se ha construido automáticamente el árbol de nodos del DOM, ya podemos comenzar a utilizar sus funciones para acceder a cualquier nodo del árbol. El acceder a un nodo del árbol, es lo equivalente a acceder a un trozo de la página de nuestro documento. Así que, una vez que hemos accedido a esa parte del documento, ya podemos modificar valores, crear y añadir nuevos elementos, moverlos de sitio, etc.

Para acceder a un nodo específico (elemento HTML) lo podemos hacer empleando dos métodos: o bien a través de los nodos padre, o bien usando un método de acceso directo. A través de los nodos padre partiremos del nodo raíz e iremos accediendo a los nodos hijo, y así sucesivamente hasta llegar al elemento que deseemos. Y para el método de acceso directo, que por cierto es el método más utilizado, emplearemos funciones del DOM, que nos permiten ir directamente a un elemento sin tener que atravesar nodo a nodo.

Algo muy importante que tenemos que destacar es, que para que podamos acceder a todos los nodos de un árbol, el árbol tiene que estar completamente construido, es decir, cuando la página HTML haya sido cargada por completo, en ese momento es cuando podremos acceder a cualquier elemento de dicha página.

Consideremos el siguiente ejemplo y veamos las formas de acceso:

```
<input type="text" id="apellidos" name="apellidos" />
```

getElementsByTagName()

Esta función obtiene una colección, que contiene todos los elementos de la página XHTML cuyo atributo name coincida con el indicado como parámetro.

```
var elementos = document.getElementsByTagName("apellidos");
```

Una colección no es un array, aunque se le parezca mucho, ya que aunque puedas recorrerla y referenciar a sus elementos como un array, no se pueden usar métodos de array, como **push** o **pop**, en la colección.

Si sólo tenemos un elemento con **name="apellidos"** para acceder a él haremos: **var elemento = document.getElementsByTagName("apellidos")[0];**

Por ejemplo, si tuviéramos 3 elementos con el atributo **name="apellidos"** para acceder al segundo elemento haríamos: **var segundo = document.getElementsByTagName("apellidos")[1];** // recordarte que los arrays comienzan en la posición 0.

Lo que nos permiten estas colecciones de elementos, es el poder recorrerlas fácilmente empleando un bucle, por ejemplo:

```
for (var j=1; j<document.getElementsByTagName("apellidos").length; j++) {
    var elemento = document.getElementsByTagName("apellidos")[j];
    ...
}
```

getElementsByTagName()

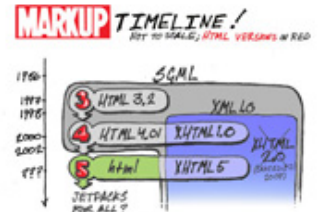
Esta función es muy similar a la anterior y también devuelve una colección de elementos cuya etiqueta XHTML coincida con la que se pasa como parámetro. Por ejemplo:

```
var elementos = document.getElementsByTagName("input");

// Este array de elementos contendrá todos los elementos input del documento.

var cuarto = document.getElementsByTagName("input")[3];
```

getElementById()



Esta función es la más utilizada, ya que nos permite acceder directamente al elemento por el ID. Entre paréntesis escribiremos la cadena de texto con el ID. Es muy importante que el ID sea único para cada elemento de una misma página. La función nos devolverá únicamente el nodo buscado. Por ejemplo:

```
var elemento= document.getElementById("apellidos");
```

Si tenemos por ejemplo una tabla con id="datos" y queremos acceder a todas las celdas de esa tabla, tendríamos que combinar `getElementById` con `getElementsByName`. Por ejemplo:

```
var miTabla= document.getElementById("datos");
```

```
var filas= miTabla.getElementsByTagName("td");
```

`getElementsByClassName()`

Muy útil para seleccionar elementos que tengan definida determinada clase CSS:

```
var elementos = document.getElementsByClassName("cabecera");
```

Créditos de la imagen

Autoría: .jssz

Licencia: CC BY-SA 2.0.

◀ Solución a la tarea para DWEC06

Ir a...



6.B. HTML DOM en W3Schools ▶