



## 6.A. Modelo de objetos del documento en JavaScript

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

#### 1.4.- Acceso a los nodos de tipo atributo.

Una vez que ya hemos visto cómo acceder a los nodos (elementos XHTML) en un documento, vamos a ver cómo podemos acceder a los nodos de tipo atributo. Para referenciar un atributo, como por ejemplo el atributo `type="text"` del campo "apellidos", emplearemos la colección `attributes`. Dependiendo del navegador, esta colección se podrá cubrir de diferentes maneras y podrán existir muchos pares en la colección, tantos como atributos tenga el elemento. Para buscar el par correcto emplearemos la propiedad `nodeName`, que nos devolverá el nombre del atributo (en minúsculas cuando trabajamos con XHTML), y para acceder a su valor usaremos `nodeValue`.

En el ejemplo:

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



Para imprimir todos los atributos del elemento "apellidos", podríamos hacer un bucle que recorriera todos esos atributos imprimiendo su valor:

```
document.write("<br/>El elemento <b>apellidos</b> contiene los pares atributo -> valor: <br/>");

for( var x = 0; x < document.getElementById("apellidos").attributes.length; x++ ) {

    var atributo = document.getElementById("apellidos").attributes[x];

    document.write(atributo.nodeName+ " -> "+atributo.nodeValue+"<br/>");

}
```

También podemos **modificar los valores de un atributo** de un nodo manualmente, por ejemplo:

```
document.getElementById("apellidos").attributes[0].nodeValue="password";

// En este caso hemos modificado el type del campo apellidos y lo hemos puesto de tipo "password".
```

O también:

```
document.getElementById("apellidos").attributes["type"].nodeValue="password";

// hemos puesto el nombre del atributo como referencia en el array de atributos.
```

O también:

```
document.getElementById("apellidos").type="password";

// hemos puesto el atributo como una propiedad del objeto apellidos y lo hemos modificado.
```

El método `setAttribute()` nos permitirá **crear o modificar atributos** de un elemento. Por ejemplo, para ponerle de nuevo al campo "apellidos" `type='text'` y un `value='Cid Blanco'`, haríamos:

```
document.getElementById("apellidos").setAttribute('type', 'text');

document.getElementById("apellidos").setAttribute('value', 'Cid Blanco');
```

Si lo que quieres realmente es chequear el valor del atributo y no modificarlo, se puede utilizar `getAttribute()`:

```
var valor = document.getElementById("apellidos").getAttribute('type');

// o también
```

```
var valor= document.getElementById("apellidos").type;
```

Y si lo que quieres es eliminar un atributo, lo podemos hacer con `removeAttribute()`:

```
// <div id="contenedor" align="left" width="200px">
```

```
document.getElementById("contenedor").removeAttribute("align");
```

```
// Obtendremos como resultado: <div id="contenedor" width="200px">
```

Créditos de la imagen

Autoría: gapysphoto.

Licencia: CC BY 2.0.

[◀ Solución a la tarea para DWEC06](#)

Ir a...



[6.B. HTML DOM en W3Schools ▶](#)