

# DOM III

## Nuevos Selectores

Los siguientes métodos están disponibles para **referenciar elementos HTML** desde Javascript:

- **getElementsByTagName** referencia un elemento por su nombre o palabra clave.
- **getElementByName** referencia un elemento por el valor de su atributo name.
- **getElementById** referencia un elemento por el valor de su atributo id.
- **getElementsByName** permite referenciar un elemento por el valor de su atributo class.

Los elementos HTML tienen que ser referenciados desde Javascript para ser afectados por el código. Además de estos métodos, **CSS3** ofrece un poderoso sistema de referencia y selección que no tiene comparación con los pocos métodos proveídos por Javascript para este propósito.

Para elevar Javascript al nivel que las circunstancias, se incorporaron nuevas alternativas. Desde ahora se puede seleccionar elementos HTML aplicando cualquier clase de selectores CSS por medio de los nuevos métodos **querySelector()** y **querySelectorAll()**.

### querySelector()

Este método devuelve **el primer elemento que concuerda** con el grupo de selectores **especificados** entre paréntesis. Los selectores son declarados usando comillas y la misma **sintaxis CSS**, como en **el ejemplo** siguiente:

Fichero **HTML**:

```
<!DOCTYPE html>
<html>
<head>
<title>DAW HTML5 & Javascript</title>
<script src="script.js"></script>
</head>
<body>
```

```
<div id="principal">
<p>Hacer Clic</p>
<p>No puede hacer Clic</p>
</div>
</body>
</html>
```

Fichero **script.js**:

```
function ferclic(){
    document.querySelector('#principal p:first-child').onclick=mostraralerta;
}
```

```
function mostraralerta(){
    alert('ha hecho clic!');
}
```

```
window.onload=fercllic;
```

Los selectores para esta consulta en particular están referenciando al **primer elemento <p>** que es **hijo del elemento identificado con el atributo id con el valor principal**.

En realidad, como este **método sólo devuelve el primer elemento encontrado**, la **pseudoclase first-child es redundante**. El método **querySelector()** en nuestro ejemplo devolverá el primer elemento <p> dentro de <div> que es su primer hijo. Se comprueba con este ejemplo que **querySelector()** acepta **cualquier clase de selectores válidos CSS** y ahora, del mismo modo que en CSS, Javascript también provee herramientas importantes para referenciar cada elemento en el documento.

**Varios grupos de selectores pueden ser declarados separados por comas**. El método **querySelector()** devolverá el primer elemento que concuerde con cualquiera de ellos.

## querySelectorAll()

En vez de solamente uno, el método **querySelectorAll()** devuelve **todos los elementos que concuerdan con el grupo de selectores** declarados entre paréntesis. El valor devuelto es un **array que contiene cada elemento encontrado en la orden en que**

aparecen en el documento.

```
function ferclic(){
    var lista=document.querySelectorAll("#principal p");
    lista[0].onclick=mostraralerta;
}
function mostraralerta(){
    alert('has hecho clic!');
}
window.onload=ferclic;
```

El grupo de selectores especificados en el método **querySelectorAll()** en el listado anterior encontrará **cada elemento <p> en el documento HTML** previo que es **hijo del elemento <div> con valor id igual a principal**. Después de la ejecución de esta primera línea, la array lista tendrá dos valores: una referencia al **primer elemento <p>** y una referencia al **segundo elemento <p>**. Por el hecho que el índice de cada array empieza por 0, en la próxima línea el primer elemento encontrado es referenciado usando corchetes y el valor 0 (**lista[0]**).

Normalmente **querySelectorAll()** será utilizado para **afectar varios elementos y no sólo uno**, como en este caso. Para interactuar con una lista de elementos devueltos por este método, podemos utilizar un **bucle for**:

```
function ferclic(){
    var lista=document.querySelectorAll("#principal p");
    for(var f=0; f<lista.length; f++){
        lista[f].onclick=mostraralerta;
    }
}
function mostraralerta(){
    alert('ha hecho clic!');
}
window.onload=ferclic;
```

Ahora en vez de seleccionar sólo el primer elemento encontrado, registramos el gestor de acontecimientos **onclick para cada uno de ellos** usando un **bucle for**. **Todos los elementos <p> dentro de <div>** mostrarán una pequeña ventana cuando el usuario haga clic sobre ellos.

El método **querySelectorAll()**, igual que **querySelector()**, **puede contener uno o más grupos de selectores separados por comas**. Estos y otros métodos estudiados **pueden ser combinados** para referenciar elementos a los que resulta difícil llegar. Por ejemplo, en el siguiente listado, el mismo resultado del código del listado anterior es conseguido utilizando **querySelectorAll()** y **getElementById()** juntos:

```
function ferclic(){
    var lista =
    document.getElementById('principal').querySelectorAll("p");
    lista[0].onclick=mostraralerta;
}
function mostraralerta(){
    alert('has hecho clic!');
}
window.onload=ferclic;
```

Usando esta técnica se puede observar como de precisos pueden ser estos **métodos**. Se pueden combinar en una misma línea y después realizar una segunda selección con otro método para conseguir elementos dentro de los primeros.