



6.A. Modelo de objetos del documento en JavaScript

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

1.6.- Creación y borrado de nodos.

La creación y borrado de nodos fue uno de los objetivos para los que se creó el DOM. Podremos crear elementos y luego insertarlos en el DOM, y la actualización quedará reflejada automáticamente por el navegador. También podremos mover nodos ya existentes (como el párrafo del punto 1.4) simplemente insertándolo en cualquier otro lugar del árbol del DOM.

Ten en cuenta que cuando estemos creando nodos de elementos, el elemento debe estar en minúsculas. Aunque en HTML ésto daría igual, el XHTML sí que es sensible a mayúsculas y minúsculas y tendrá que ir, por lo tanto, en minúsculas.



Usaremos los métodos `createElement()`, `createTextNode()` y `appendChild()`, que nos permitirán crear un elemento, crear un nodo de texto y añadir un nuevo nodo hijo.

Ejemplo de creación de un nuevo párrafo, suponiendo que partimos del siguiente código HTML:

```
<p title="Texto de un párrafo" id="parrafito">Esto es un ejemplo de <b>texto HTML<br />
que puedes tener</b> en tu documento.</p>
```

Para crear el nuevo párrafo haremos:

```
var nuevoParrafo = document.createElement('p');

var nuevoTexto = document.createTextNode('Contenido añadido al párrafo.');
```

```
nuevoParrafo.appendChild(nuevoTexto);

document.getElementById('parrafito').appendChild(nuevoParrafo);
```

Y obtendremos como resultado HTML:

```
<p id="parrafito" title="Texto de un párrafo">
Esto es un ejemplo de <b>texto HTML<br>que puedes tener</b>en tu documento.

<p>Contenido añadido al párrafo.</p> </p>
```

Podríamos haber utilizado `insertBefore` en lugar de `appendChild` o, incluso, añadir manualmente el nuevo elemento al final de la colección de nodos `childNodes`. Si usamos `replaceChild`, incluso podríamos sobrescribir nodos ya existentes. También es posible copiar un nodo usando `cloneNode(true)`. Ésto devolverá una copia del nodo, pero no lo añade automáticamente a la colección `childNodes`.

Para eliminar un nodo existente, lo podremos hacer con `element.removeChild(referencia al nodo hijo)`.

Ejemplo de creación de elementos e inserción en el documento:

```
//Creamos tres elementos nuevos: p, b, br

var elementoP = document.createElement('p');

var elementoB = document.createElement('b');

var elementoBR = document.createElement('br');

//Le asignamos un nuevo atributo title al elementoP que hemos creado.

elementoP.setAttribute('title','Parrafo creado desde JavaScript');
```

```
//Preparamos los nodos de texto

var texto1 = document.createTextNode('Con JavaScript se ');

var texto2 = document.createTextNode('pueden realizar ');

var texto3 = document.createTextNode('un monton');

var texto4 = document.createTextNode(' de cosas sobre el documento.');
```



```
//Añadimos al elemento B los nodos de texto2, elemento BR y texto3.

elementoB.appendChild(texto2);

elementoB.appendChild(elementoBR);

elementoB.appendChild(texto3);
```



```
//Añadimos al elemento P los nodos de texto1, elemento B y texto 4.

elementoP.appendChild(texto1);

elementoP.appendChild(elementoB);

elementoP.appendChild(texto4);
```



```
//insertamos el nuevo párrafo como un nuevo hijo de nuestro parrafo

document.getElementById('parrafito').appendChild(elementoP);
```

Créditos de la imagen

Autoría: Kanko.

Licencia: CC BY 2.0.

◀ Solución a la tarea para DWEC06

Ir a...



6.B. HTML DOM en W3Schools ►