DOM Avanzado 4

En la primera parte de este manual aprendimos a acceder a los diferentes nodos del DOM y a sus propiedades. Sin embargo, no se enseñó a modificar un árbol DOM existente, creando nuevos nodos o eliminando algunos de los existentes.

En este capitulo vamos a aprender a tener un control total sobre el árbol DOM, pudiendo crear, modificar, reemplazar o eliminar nodos.

Destacar que, muy posiblemente, todo lo que se explica en este tema no funcione correctamente en versiones del navegador Internet Explorer (sobre todo anteriores a la 10). Aún así, eso es algo que hoy por hoy no debe preocuparnos puesto que existen muchísimas alternativas para que los usuarios dejen de usar esos navegadores obsoletos que nunca han seguido el estándar web y a los que grandes webs como Facebook, Instagram o Twitter ya no dan soporte.

4.1 Gestión de nodos DOM

Los métodos más importantes para la gestión de nodos del árbol DOM son:

`setAtribute(nombre,valor)`	Crea un nodo de tipo atributo con el nombre y valor indicados.
`createComment(texto)`	Crea un nodo tipo comentario con el texto indicado.
`createElement(nombre_etiqueta)`	Crea un elemento del tipo indicado en el parámetro `nombre_etiqueta`.
`createTextNode(texto)`	Crea un nodo tipo texto con el valor indicado.

`appendChild(nodo)`	Cuelga un nodo a otro (es decir, añade un hijo a un nodo).
`parentNode`	Devuelve una referencia al nodo padre.
`removeChild(nodo)`	Descuelga el nodo indicado de su padre y lo elimina.
`insertBefore(nuevo,existente)`	Inserta un nuevo nodo antes del nodo existente que se indica en los argumentos.
`replaceChild(nuevo, viejo)`	Sustituye el nodo `nuevo` por el nodo `viejo`.

4.1.1 Creación de nodos

Para crear un nodo nuevo en el árbol DOM debemos seguir los siguientes pasos:

1. Crear un nodo del tipo deseado.
2. Colocar los atributos de ese nodo (si los tuviera).
3. Si el contenido de ese nodo es texto, crear un nodo texto.
4. Asociar el nodo texto al nodo creado en el paso 1.
5. Asociar el nuevo nodo a un nodo ya existente o al cuerpo de la página (`body`)

Vamos a ver detenidamente esta serie de pasos haciendo uso de la siguiente página web simple:

```
<html>
  <head>
    <title>Prueba de Nodos</title>
  </head>
  <body>
    <div id="capa"></div>
    <div id="pie"></div>
  </body>
</html>
```

Creamos el siguiente nodo tipo párrafo:

```
<p> Este párrafo no existía antes </p>
```

Lo primero es crear un nodo del tipo deseado. En nuestro caso, un párrafo:

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

A continuación añadimos los atributos que pudiera tener esa etiqueta. Obviamente, si la etiqueta no tiene atributos, este paso no se realiza.

```
p.setAttribute('align','center');
```

El siguiente paso sería, si el contenido del elemento es texto, crear un nodo tipo texto:

```
var texto = document.createTextNode("Este parrafo no estaba
existía antes");
```

Finalizamos la creación del nodo, enlazando el nodo texto del paso anterior:

```
p.appendChild(texto);
```

En este punto tenemos creado el nodo párrafo con el atributo y el texto indicados. El siguiente paso es decidir a qué nodo del árbol enganchamos este, es decir, quién será el padre.

Importante: mientras no enganchemos el nodo al árbol DOM, dicho nodo no se mostrará en la web.

Si deseamos que el nuevo párrafo esté dentro del DIV, haríamos:

```
var capa = document.getElementById('capa');
capa.appendChild(p);
```

Si por el contrario queremos que el párrafo cuelgue del BODY, cambiaríamos lo anterior por:

```
document.body.appendChild(p);
```

En el caso de querer justo antes del DIV *pie*, cambiamos lo anterior por:

```
var pie =  document.getElementById('pie');
document.body.insertBefore(p,pie);
```

El código total del ejemplo quedaría de la siguiente forma:

```
//PASO 1
var p = document.createElement('p');
//PASO 2
p.setAttribute('align','center');
//PASO 3
var texto = document.createTextNode("Este parrafo no
estaba existía antes");
//PASO 4
p.appendChild(texto);
```

Dependiendo de donde vayamos a colocar el nuevo nodo, tendríamos uno de los siguientes casos para el paso 5:

```
//PASO 5
//Enlazo al DIV capa
var capa = document.getElementById('capa');
capa.appendChild(p);
```

```
//PASO 5
//Para enlazar con BODY :
document.body.appendChild(p);
```

```
//PASO 5
//Para colocarlo justo antes del DIV pie
var pie =  document.getElementById('pie');
document.body.insertBefore(p,pie);
```

Destacar que la creación de nuevos nodos solo se puede hacer una vez que se haya cargado todo el árbol DOM de la página. Es decir, este tipo de scripts deben esperar a la carga del árbol DOM para que funcionen correctamente.

4.1.2 Eliminación de nodos

Eliminar nodos ya existentes en el árbol DOM es bastante más sencillo que crearlos. Tan solo debemos colocarnos en el padre de ese nodo y aplicar la función correspondiente para su eliminación.

Pasos para eliminar un nodo:

1. Obtener la referencia a ese nodo.
2. Obtener la referencia al padre del nodo.
3. Eliminar el nodo desde el padre.

Vamos a eliminar el nodo tipo `<p>` del siguiente ejemplo:

```
<html>
  <head>
    <title>Prueba de Nodos</title>
  </head>
  <body>
    <div id="capa"></div>
    <p id="texto">Voy a ser destruido </p>
    <div id="pie"></div>
  </body>
</html>
```

Lo primero es obtener la referencia a ese elemento:

```
var elemento = document.getElementById('texto');
```

A continuación obtenemos la referencia al padre del nodo:

```
var padre = elemento.parentNode
```

Para finalizar, borro el elemento a través del padre:

```
padre.removeChild(elemento);
```

El código completo sería:

```
//PASO 1
var elemento = document.getElementById('texto');


//PASO 2
var padre = elemento.parentNode


//PASO 3
padre.removeChild(elemento);
```

Indicar que a la hora de borrar un nodo es indiferente quien sea el padre dado que la función `parentNode` nos va a devolver una referencia a dicho padre, sea el `body`, sea un `div`, sea cualquier otra etiqueta.

4.1.3 Sustitución de nodos

Una vez que hemos aprendido a crear y eliminar nodos, el siguiente paso es aprender como sustituir un nodo existente por otro nuevo. Esto es muy interesante y puede servir para, por ejemplo, cuando queremos que el usuario decida qué se ve y qué no se ve en la página.

Los pasos que debemos seguir si queremos sustituir un nodo por otro son:

1. Crear un nuevo nodo.
2. Obtener la referencia al padre del nodo que queremos sustituir.
3. Borrar el nodo a sustituir y colocar el nuevo nodo asociado al padre o usar directamente la función `replaceChild`.

Como vemos en el esquema de pasos, todas las acciones que debemos hacer ya las hemos visto bien en la creación o bien en la eliminación de nodos. Lo único nuevo sería la función `replaceChild` que, en el caso de querer usarla, nos va a facilitar mucho la sustitución.

Vamos a usar el siguiente ejemplo para sustituir el DIV por un nuevo párrafo:

```
<html>
  <head>
    <title>Sustitucion de Nodos</title>
  </head>
  <body>
    <div id="capa"></div>
  </body>
</html>
```

Lo primero es crear un nuevo nodo tal y como vimos en los cuatro primeros pasos del apartado 3.1.1 de este capítulo:

```
var p = document.createElement('p');
var texto = document.createTextNode("Este parrafo es
nuevo");
p.appendChild(texto);
```

A continuación obtenemos la referencia al padre del nodo que vamos a sustituir:

```
var viejo = document.getElementById('capa');
var padre = viejo.parentNode
```

Para finalizar podemos hacer dos cosas: eliminar el nodo DIV como se vio en el apartado 3.1.2 y a continuación, enlazar el nuevo nodo al padre. o mejor, usar la función `replaceChild` que hará todo ese trabajo por nosotros:

```
padre.replaceChild(p,viejo);
```

El código completo del ejemplo sería el siguiente:

```
//PASO 1
var p = document.createElement('p');
var texto = document.createTextNode("Este parrafo es
nuevo");
p.appendChild(texto);

//PASO 2
var viejo = document.getElementById('capa');
var padre = viejo.parentNode

//PASO 3 - Podría borrar y luego enlazar o mejor usar:
padre.replaceChild(p,viejo);
```