

[Ir directamente al contenido de esta página](#)

codexemplar.org

Buscar:

Sobre la modificación de los nodos del árbol de un documento

appendChild

Vamos a empezar viendo cómo se **añadiría** un **nodo al árbol** de un **documento**. Primero veamos el ejemplo en funcionamiento...

En el interior de este recuadro aparecerán los párrafos añadidos.

Añadir un párrafo

...y ahora su código:

```
function anadir_parrafo(){
  var parrafo = document.createElement('p');
  var texto = document.createTextNode('Párrafo añadido.');
```

Primero, **como ya vimos**, creo el párrafo —`document.createElement('p')`— y luego el texto que va a contener —`document.createTextNode('Párrafo añadido.')`—. Después empleo dos veces `appendChild`, una para insertar el nodo de texto dentro de mi elemento `p`, y otra para incluir ese mismo párrafo en el `div` que he llamado `ej-aCh`.

insertBefore

Ahora vamos a insertar el mismo párrafo, pero antes del párrafo que ya existe en el siguiente `div`:

Sobre esta línea aparecerán los párrafos añadidos.

Insertar un párrafo

El código:

```
function insertar_antes(){
  var parrafo = document.createElement('p');
  parrafo.appendChild(document.createTextNode('Párrafo añadido.'));
  var elementoReferencia = document.getElementById('ej-iB').getElementsByTagName('p')[0];
  document.getElementById('ej-iB').insertBefore(parrafo, elementoReferencia);
}
```

La sintaxis de `insertBefore` puede resultar un poco más enrevesada que la de `appendChild`, por lo que vamos a verla paso a paso:

1. `document.createElement('p')`: Igual que para el ejemplo anterior, creo el párrafo.
2. `parrafo.appendChild(document.createTextNode('Párrafo añadido.'))`: De la misma forma, le incluyo un texto, sólo que en una sola línea de código sin emplear —a diferencia del caso anterior— una variable para almacenar el nodo de texto, sino que lo incluyo «al vuelo».
3. `document.getElementById('ej-iB').getElementsByTagName('p')[0]`: Con esta línea selecciono el primer párrafo de `ej-iB`, valor que almaceno en *elementoReferencia*.
4. `document.getElementById('ej-iB').insertBefore(parrafo, elementoReferencia)`: Ahora selecciono el padre del elemento antes del cual quiero insertar mi nuevo párrafo —el propio `div` identificado como `ej-iB`, y al método le paso como parámetros la variable que almacena el nuevo párrafo y el que he seleccionado como punto de referencia.

replaceChild

Vamos a hora a reemplazar un párrafo por otro:

Este párrafo será reemplazado cuando haga clic en el botón...

Reemplazar el párrafo

El código:

```
function reemplazar(){
```

```

    var nuevoParrafo = document.createElement('p');
    nuevoParrafo.appendChild(document.createTextNode('...como queríamos demostrar.'));
    var viejoParrafo = document.getElementById('ej-rCh').getElementsByTagName('p')[0];
    document.getElementById('ej-rCh').replaceChild(nuevoParrafo,viejoParrafo);
}

```

Si se ha comprendido antes `insertBefore`, la sintaxis de `replaceChild` no es ningún misterio. Al igual que para la primera, lo que hago es crear mi nuevo párrafo, almacenar el que quiero sustituir en una variable, y por último seleccionar `ej-rCh` para aplicar el método que reemplaza con el nuevo párrafo el viejo.

removeChild

Seguidamente vamos a eliminar un nodo:

Este párrafo será eliminado cuando haga clic en el botón.

Eliminar el párrafo

El código:

```

function eliminar(){
    var parrafoAEliminar = document.getElementById('ej-rmCh').getElementsByTagName('p')[0];
    document.getElementById('ej-rmCh').removeChild(parrafoAEliminar);
}

```

Éste es fácil: sólo hay que seleccionar el nodo que se quiere eliminar y emplear el método `removeChild` del padre, que es `ej-rmCh`.

cloneNode

Por último, vamos a clonar un nodo, tanto sin su contenido como con él:

Este elemento contiene un párrafo y una lista.

- Ítem.
- Ítem.
- Ítem.

Clonar el elemento

Clonar el elemento y sus descendientes

El último código:

```

function clonar(x){
    var clon = document.getElementById('ej-cN').cloneNode(x);
    document.getElementById('referente-clon').parentNode.insertBefore(clon,document.getElementById('referente-clon'));
}

```

Para aprovechar la misma función, paso el parámetro que necesita `cloneNode` como parámetro de la función desde la llamada en los botones, `false` para clonar sólo en nodo `ej-cN`, y `true` para clonarlo con su contenido.

En este caso ocurre como con `createElement` o `createTextNode`: la primera línea de la función es la que crea el clon, pero éste permanece fantasmal hasta que no lo incluyamos en el documento. Por ello en la segunda línea lo que hago es emplear `insertBefore` para situarlo antes de los botones, que se encuentran en un párrafo identificado como `referente-clon`.

[Volver al curso](#)

(cc) CodexExempla.org, 2007–2018 [Mapa del sitio](#) | [XHTML](#) | [CSS](#) | [AA](#)