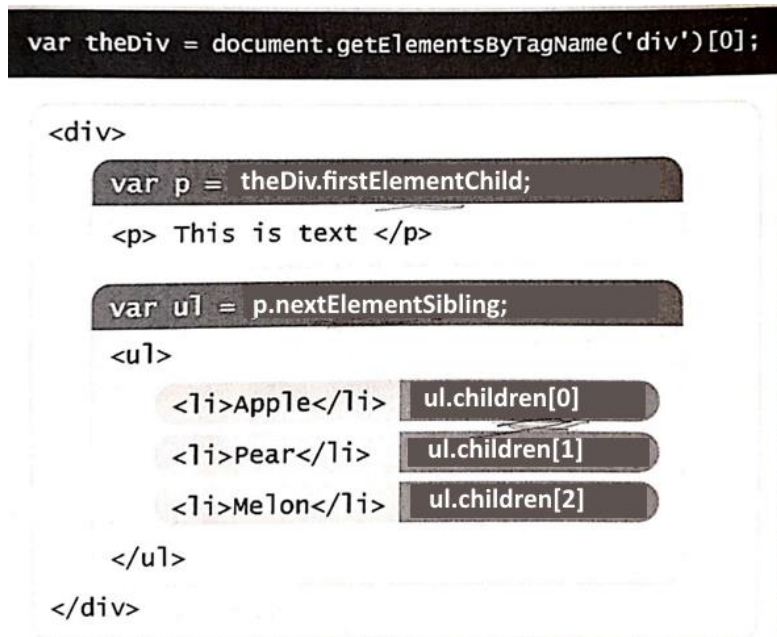


MANIPULACIÓN AVANZADA DEL DOM

1 Navegación del DOM

La imagen adjunta nos permite ver de forma gráfica el resultado de utilizar alguno de los métodos de acceso al DOM:



```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <meta charset="UTF-8">
5.   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6.   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7.   <title>Document</title>
8. </head>
9. <body>
10.  <div id="bloque">
11.    <p>This is text</p>
12.    <ul>
13.      <li>Apple</li>
14.      <li>Pear</li>
15.      <li>Melon</li>
16.    </ul>
17.  </div>
18.  <script>
19.    var theDiv=document.getElementsByTagName('div')[0]; //Referencia
    a elto div
20.    var p=theDiv.firstChild; //Referencia a elto p
21.    var ul=p.nextElementSibling; //Referencia a elemento ul
22.    var eltol1=ul.children[0]; //Referencia a elto li Apple
23.    var eltol2=ul.children[1]; //Referencia a elto li Pear
24.    var eltol3=ul.children[2]; //Referencia a elto li Melon
25.
26.    alert("Contenido del párrafo: " + p.innerHTML);
```

```

27.         alert("Contenido del primer elto li: " + eltolii.innerHTML);
28.         alert("Contenido del segundo elto li: " + eltolii.innerHTML);
29.         alert("Contenido del tercer elto li: " + eltolii.innerHTML);
30.     </script>
31. </body>
32. </html>

```

(Código en fichero ej_navegacion_dom.html)

2 Manipulación del DOM mediante código HTML

Esta técnica se base en el uso de cadenas de caracteres que contienen código HTML:

En la siguiente tabla se describen las propiedades y el método con los que podemos manipular el DOM mediante código HTML.

Propiedad/método	Descripción
innerHTML	Propiedad asociada al código HTML que contiene el elemento
outerHTML	Propiedad asociada al código HTML del elemento, incluido su contenido
insertAdjacentHTML(<i>posición</i> , <i>codigo</i>)	Este método sirve para insertar código HTML en una posición relativa al elemento actual. Los valores posibles de <i>posición</i> son las cadenas 'beforebegin' (inserta el código como hermano mayor del elemento actual), 'afterbegin' (inserta el código como hijo primogénito del elemento actual), 'beforeend' (inserta el código como hijo benjamín del elemento), y 'afterend' (inserta el código como hermano menor del elemento)

Veamos un ejemplo:

```

1. <!DOCTYPE html>
2. <html lang="es">
3. <head>
4.     <title>Modificar el DOM</title>
5.     <meta charset="utf-8" >
6. </head>
7. <body>
8. <script>
9.     var total=0;
10.    function addNewItem(){
11.        total++;
12.        document.getElementsByTagName('ul')[0].innerHTML += '<li>Elemento'
+ total + '</li>';
13.    }
14.    function copiarLista(){
15.        var lista=document.getElementsByTagName('ul')[0];
16.        lista.insertAdjacentHTML('afterend',lista.outerHTML);
17.    }
18. </script>

```

```

19. <div id="elementosTag">
20. <h3>Modificar el DOM</h3>
21. <input type="button" id="btnNuevoItem" onclick="addNewItem();" value="Agregar
    nuevo elemento" />
22. <input type="button" id="btnCopiarLista" onclick="copiarLista();" value="Copia
    r lista" />
23. <p>Lorem ipsum dolor sit amet</p>
24. <ul id="listaItems"></ul>
25. </div>
26. </body>
27. </html>

```

(Código en fichero ej_4.4.5_3.html)

3 Manipulación del DOM mediante objetos de tipo Element

En la siguiente table se describen los métodos que nos permiten manipular el DOM mediante objetos de tipo Element; los dos primeros pertenecen al objeto document y el resto puede aplicarse sobre cualquier objeto de tipo Element.

Método	Descripción
<u>document.createElement(etiqueta)</u>	Devuelve un objeto de tipo nodo de elemento y, más concretamente, del tipo de elemento correspondiente a la <i>etiqueta</i> HTML. Este método crea el nodo, pero no lo inserta en ninguna posición del DOM, por lo que no será visible hasta que lo ubiquemos
<u>document.createTextNode(texto)</u>	Devuelve un objeto de tipo nodo de texto con el <i>texto</i> indicado. Este método crea el nodo, pero no lo inserta en ninguna posición del DOM, por lo que no será visible hasta que lo ubiquemos
<u>appendChild(nodo)</u>	Inserta el <i>nodo</i> en el DOM como hijo menor del elemento actual
<u>insertBefore(nodo1,nodo2)</u>	Inserta el <i>nodo1</i> como hermano inmediatamente mayor de <i>nodo2</i> dentro del elemento actual
<u>removeChild(nodo)</u>	Elimina el nodo del DOM y lo devuelve como una referencia que podemos recoger en una variable para rescatarlo posteriormente
<u>replaceChild(nuevo,viejo)</u>	Sustituye el nodo <i>viejo</i> por el <i>nuevo</i> dentro del elemento actual
<u>cloneNode(incluirDescendientes)</u>	Devuelve un elemento que es la copia exacta del actual, salvo porque no incluye sus detectores de eventos (los eventos se tratarán posteriormente en este mismo tema). <i>incluirDescendientes</i> es un valor booleano; si es <i>true</i> duplica el elemento y todos sus nodos descendientes; si es <i>false</i> duplica el nodo de elemento sin ninguno de sus descendientes (ni siquiera los nodos de tipo texto)

El siguiente ejemplo aplicamos estos métodos en elemento de tipo table donde añadimos, borramos y copiamos filas dinámicamente:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.     <meta charset="uft-8" />
5.     <title>Acceso DOM</title>
6. </head>
7. <body>
8.     <span>Texto</span>
9.     <input type="text" id="txtCaja">
10.    <input type="button" onclick="insertar();" value="Insertar texto nuevo">
11.    <table id="tabla" style="border:1px dashed green;"></table>
12.    <script>
13.    function insertar(){
14.        var texto=document.getElementById('txtCaja').value;
15.        var codigoFila="<td>" + texto + "</td>";
16.        codigoFila += "<td><input type='button' onclick='copiarEncima(this);' valu
17.        e='Copiar encima'></td>";
18.        codigoFila += "<td><input type='button' onclick='borrar(this);' value='Bor
19.        rar'></td>";
20.        var fila=document.createElement('tr');
21.        fila.innerHTML=codigoFila;
22.        document.getElementById('tabla').appendChild(fila);
23.    }
24.    function borrar(boton){
25.        var filaABorrar=boton.parentElement.parentElement; // Será un elto tr
26.        // Manera 1 de eliminar el elto tr
27.        // filaABorrar.remove();
28.        // Manera 2 de eliminar el elto tr
29.        document.getElementById('tabla').removeChild(filaABorrar);
30.    }
31.    function copiarEncima(boton){
32.        var filaActual=boton.parentElement.parentElement;
33.        var filaACopiar=boton.parentElement.parentElement.cloneNode(true);
34.        document.getElementById('tabla').insertBefore(filaACopiar,filaActual);
35.    }
36.    </script>
37. </body>
38. </html>
```

(Código en fichero ej_4.4.5_4.html)