

UNIDAD 6	2
DOM. Acceso y modificación de elementos. Texto y atributos	2
DOM. Creación, sustitución y borrado de elementos	4
DOM. Acceso y navegación por elementos	8

UNIDAD 6

DOM. Acceso y modificación de elementos. Texto y atributos

Modelo de Objetos del Documento (DOM) es un estándar del W3C para acceder a los elementos de una página HTML o XML.

En el DOM se definen los **objetos** y **propiedades** de todos los elementos del documento, y los métodos para acceder a ellos.

Podemos **acceder a los diferentes elementos de una página web, modificar su texto y atributos**. Para ello tenemos varias formas de seleccionar elementos, entre ellas, los métodos más antiguos:

- A partir de su identificador, mediante **getElementById**.
- A partir de su clase, mediante **getElementsByClassName**.
- A partir de su etiqueta, mediante **getElementsByTagName**.
- A partir de su propiedad name, mediante **getElementsByName**.

<https://lenguajejs.com/javascript/dom/seleccionar-elementos-dom/#m%C3%A9todos-tradicionales>

y los más modernos:

- A partir de su selector CSS, mediante **querySelector() y querySelectorAll()**

<https://lenguajejs.com/javascript/dom/seleccionar-elementos-dom/#m%C3%A9todos-modernos>

Acceso a los atributos de un elemento con

document.getElementById("apellidos").attributes[x]; y para conocer cuál es el tipo de atributo haríamos: **document.getElementById("apellidos").attributes[x].nodeName** y para acceder o modificar el valor del atributo haríamos:

document.getElementById("apellidos").attributes[x].nodeValue

Otra forma de acceder a los atributos sería poniendo el atributo, en este ejemplo accedemos o modificamos el atributo type , así:

document.getElementById("apellidos").type

Podemos acceder a los elementos de un formulario a través de la propiedad `elements` del formulario de la siguiente manera (se accedería únicamente a los elementos de formulario de tipo input) `document.forms[0].elements[0]`

<https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement/elements>

No es lo mismo utilizar un **método que devuelva un único elemento**, como es el primer y último caso, que utilizar cualquiera de los otros tres, que **devuelven una lista iterable (no es exactamente lo mismo que un array, por lo que no podríamos utilizar los métodos del objeto array)** y, por lo tanto, deben ser procesados como tal, aunque podemos convertirlos a un array fácilmente con el spread operator (`array = [...iterable]`).

Podemos también **modificar el texto** que contiene un elemento «no vacío», como es el caso de un párrafo, utilizando `.innerHTML`; también podemos **añadir un atributo a un elemento**, mediante el método `.setAttribute` o acceder con `.getAttribute` o borrarlo con `.removeAttribute`; y **añadir o quitar clases CSS** utilizando `.className`.

Tenemos un buen Resumen en:

[Seleccionar elementos del DOM - Javascript en español](#)

también en : https://www.w3schools.com/js/js_htmlDOM_elements.asp

Ejemplos:

```
<html>
  <head>
    <title>Document Object Model - Creación y modificación de
elementos</title>
    <meta charset="UTF-8">
    <style>
      .miClase{
        color:red;
      }
    </style>
    <script>
      window.addEventListener("load",inicio);

      function inicio(){

document.getElementById("cambiarTextos").addEventListener("click",cambiarTextos);

document.getElementById("cambiarClases").addEventListener("click",cambiarClases);

document.getElementById("quitarClases").addEventListener("click",quitarClases);
      }
```

```

    }

    function cambiarTextos(){
        //Selecciónar UN ÚNICO elemento según su id
        document.getElementById("parrafo1").innerHTML = "Primer
párrafo cambiado";
        //Seleccionar TODOS los elementos según su etiqueta
        document.getElementsByTagName("p")[1].innerHTML = "Segundo
párrafo cambiado";
        //Seleccionar TODOS los elementos según su clase
        document.getElementsByClassName("miClase")[0].innerHTML =
"Terceer párrafo cambiado";

        //Seleccionar el PRIMER elemento p con clase=miclase
        con un selector CSS
        /*document.querySelector("p.miClase");
        document.querySelector("p.miClase")[0];*/
    }

    function cambiarClases(){
        //Dos formas
document.getElementById("parrafo1").setAttribute("class","miClase");
        document.getElementById("parrafo2").className = "miClase";
    }

    function quitarClases(){
//Quitar todas las clases a la vez de un tipo de elemento
        for(var i = 0; i<document.getElementsByTagName("p").length;
i++){
            document.getElementsByTagName("p")[i].className="";
        }
    }
</script>
</head>
<body>
    <div id="main">
        <p id="parrafo1">Este es el primer párrafo</p>
        <p id="parrafo2">Este es el segundo párrafo</p>
        <p id="parrafo3" class="miClase">Este es el tercer párrafo</p>
    </div>
    <br/>
    <button id="cambiarTextos">Cambiar texto de párrafos</button>
    <button id="cambiarClases">Cambiar clases de párrafos</button>
    <button id="quitarClases">Quitar clases de todos los párrafos</button>
</body>
</html>

```

DOM. Creación, sustitución y borrado de elementos

Aquí vemos cómo **crear, modificar y borrar elementos en un documento HTML**. Concretamente:

- Cómo **crear un nuevo elemento** con el método `.createElement`.
- Cómo **crear un nuevo elemento textual** para incluirlo en elementos HTML «no vacíos», mediante el método `.createTextNode`.
- Cómo **incluir un elemento dentro de otro** o del documento con `.appendChild`.
- Cómo **sustituir y borrar elementos** con `.replaceChild`, `.removeChild` y `.remove` respectivamente.
- Y cómo **trabajar con atributos** utilizando `.setAttribute`, `.getAttribute` y `.removeAttribute`

Crear elementos del DOM: <https://lenguajejs.com/javascript/dom/crear-elementos-dom/>

Insertar elementos del DOM: <https://lenguajejs.com/javascript/dom/insertar-elementos-dom/>

Gestión de atributos: <https://lenguajejs.com/javascript/dom/atributos-del-dom/>

https://www.w3schools.com/js/js_htmlDOM_nodes.asp

Ejemplos:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
  <title>Document Object Model</title>
  <style>
    .miClase {
      color: red;
    }
  </style>
  <script>
    window.addEventListener("load", inicio);

    function inicio() {
      document.getElementById("crearParrafo").addEventListener("click",
crearParrafo);
      document.getElementById("crearImagen").addEventListener("click",
crearImagen);
    }
  </script>
</head>

<body>
  <div id="crearParrafo">
    <button>Crear Parrafo</button>
  </div>
  <div id="crearImagen">
    <button>Crear Imagen</button>
  </div>
</body>
</html>
```

```

        document.getElementById("borrarUltimo").addEventListener("click",
borrarUltimo);
        document.getElementById("borrarPrimero").addEventListener("click",
borrarPrimero);

document.getElementById("sustituirPrimeroVacio").addEventListener("click",
sustituirPrimero);
    }

    function crearParrafo() {
        //Crear elemento
        var parrafo = document.createElement("p");
        //Crear texto
        var texto =
document.createTextNode(document.getElementById("texto").value);
        parrafo.appendChild(texto);

        //Añadir atributos
        parrafo.setAttribute("className", "miClase");
        //parrafo.className = "miClase";

        var cont = document.getElementById("div1");
        cont.appendChild(parrafo);
    }

    function crearImagen() {
        //Crear elemento
        var imagen = document.createElement("img");

        //Añadir atributos
        var ruta = prompt("Introduce la ruta");
        imagen.setAttribute("alt", prompt("Introduce el texto
alternativo"));
        imagen.src = ruta;

        var cont = document.getElementById("div1");
        cont.appendChild(imagen);
    }

    function borrarUltimo() {
        var cont = document.getElementById("div1");
        var hijo = cont.lastChild; //Seleccionar el último hijo
        //Elimina un elemento hijo
        cont.removeChild(hijo);
    }

    function borrarPrimero() {
        var cont = document.getElementById("div1");
        var hijo = cont.firstChild; //Seleccionar el primer hijo
        //Elimina un elemento hijo
        cont.removeChild(hijo);
    }

    function sustituirPrimero() {
        var cont = document.getElementById("div1");

```

```
        var parrafo =
document.createElement("p").appendChild(document.createTextNode("Vacio"));

        //Sustituir un elemento
        cont.replaceChild(parrafo, cont.firstChild); //Nuevo, viejo
    }
</script>
</head>

<body>
    <div id="div1"></div>
    <br/>
    <textarea id="texto"></textarea>
    <br/>
    <button id="crearParrafo">Crear parrafo</button>
    <button id="crearImagen">Crear imagen</button>
    <button id="borrarUltimo">Borrar ultimo</button>
    <button id="borrarPrimero">Borrar primero</button>
    <button id="sustituirPrimeroVacio">Sustituir primer párrafo por
vacío</button>

</body>

</html>
```

DOM. Acceso y navegación por elementos

Veamos cómo **movernos entre los elementos del DOM**. Y para ello deberemos entender el «parentesco» entre nodos: ¿quién es el **padre** de quién? ¿y el **hermano**? ¿y el **primer hijo**? ¿y el **último**?

veamos cómo:

- Seleccionar el **padre de un elemento** con `.parentElement`.
- Acceder a **todos los hijos**, con `.childNodes` y `.children` (childNodes incluye texto y comentarios y children no)
- Acceder a **primer o último hijo** con `.firstChild` o `.lastChild`
- Encontrar el **siguiente hermano**, con `.nextSibling` y `.nextElementSibling`.

Dado este elemento `<p>` :

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

este código, accedería a “texto HTML”

```
document.getElementsByTagName("p")[0].childNodes[1].childNodes[0].nodeValue
```

A continuación se muestra la página HTML básica que se va a emplear los siguientes ejemplos:

```
<html>
<head>
  <title>Aprendiendo DOM</title>
</head>
<body>
  <p>Aprendiendo DOM</p>
  <p>DOM es sencillo de aprender</p>
  <p>Además, DOM es muy potente</p>
</body>
</html>
```

La operación básica consiste en obtener el objeto que representa el elemento raíz de la página:

```
var objeto_html = document.documentElement;
```

Después de ejecutar la instrucción anterior, la variable `objeto_html` contiene un objeto de tipo `HTMLElement` y que representa el elemento `<html>` de la página web. Según el árbol de nodos DOM, desde el nodo `<html>` derivan dos nodos del mismo nivel jerárquico: `<head>` y `<body>`.

Utilizando los métodos proporcionados por DOM, es sencillo obtener los elementos <head> y <body>. En primer lugar, los dos nodos se pueden obtener como el primer y el último nodo hijo del elemento <html>:

```
var objeto_head = objeto_html.firstChild;
var objeto_body = objeto_html.lastChild;
```

Otra forma directa de obtener los dos nodos consiste en utilizar la propiedad childNodes del elemento <html>:

```
var objeto_head = objeto_html.childNodes[0];
var objeto_body = objeto_html.childNodes[1];
```

Si se desconoce el número de nodos hijo que dispone un nodo, se puede emplear la propiedad length de childNodes:

```
var numeroDescendientes = objeto_html.childNodes.length;
```

Además, el DOM de HTML permite acceder directamente al elemento <body> utilizando el atajo document.body:

```
var objeto_body = document.body;
```

Además de las propiedades anteriores, existen otras propiedades como previousSibling y parentNode que se pueden utilizar para acceder a un nodo a partir de otro. Utilizando estas propiedades, se pueden comprobar las siguientes igualdades:

```
objeto_head.parentNode == objeto_html
objeto_body.parentNode == objeto_html
objeto_body.previousSibling == objeto_head
objeto_head.nextSibling == objeto_body
objeto_head.ownerDocument == document
```

Más información en:

<https://lenguajejs.com/javascript/dom/navegar-elementos-dom/>

https://www.w3schools.com/js/js_htmlDOM_navigation.asp

Ejemplos:

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8" />
```

```

<title>Document Object Model</title>
<script>
    window.addEventListener("load", inicio);

    function inicio() {
//Asignamos al click de todos los h2 del documento la función recorrerAdelante
        for (var i = 0; i < document.getElementsByTagName("h2").length; i++)
        {
document.getElementsByTagName("h2")[i].addEventListener("click",
recorrerAdelante);
        }
    }

function recorrerAdelante(e) {
    var texto = "";
    var h2 = e.target;
    texto += "Has elegido " + h2.textContent;

    //Seleccionamos el padre
    var padre = h2.parentElement;
    texto += " que está situada en el " + padre.id + "<br/>";

    //Seleccionamos el hermano
    //var ul = h2.nextSibling; //No lo utilizamos porque incluye nodos
de tipo texto
    var ul = h2.nextElementSibling;

    //Mostrar el número de hijos
    //¡OJO! .childNodes incluye texto y comentarios; children no.
    texto += "El número de provincias es " + ul.children.length + ":
";

    var hijo = ul.firstChild;
    //Si quisiéramos seleccionar el último hijo, pondríamos
.lastChild.

    while (hijo != null) {
        texto += " " + hijo.textContent;
        hijo = hijo.nextElementSibling;
        //Si quisiéramos seleccionar el hermano anterior, pondríamos
.previousElementSibling
    }

    //Mostramos el texto
    document.getElementById("texto").innerHTML = texto;
    //Si ponemos .textContent mostraríamos texto plano, y no coge
etiquetas html.
    }
}
</script>
</head>

<body>
    <div id="norte">
        <h2>Galicia</h2>

```

```
<ul id="lista">
  <li>A Coruña</li>
  <li>Lugo</li>
  <li>Ourense</li>
  <li>Pontevedra</li>
</ul>
</div>
<div id="sur">
  <h2>Canarias</h2>
  <ul id="lista">
    <li>Santa Cruz de Tenerife</li>
    <li>Las Palmas</li>
  </ul>
</div>
<div id="texto"></div>
</body>
</html>
```