

Unidad 7: Utilización del Modelo de Objetos del Documento (DOM-Document Object Model)

Desarrollo Web en Entorno Cliente

Objetivos



- Reconocer el modelo de objetos del documento de una página web, identificando sus objetos, propiedades y métodos.
- Generar y verificar código que acceda a la estructura del documento y crear elementos de la estructura.
- Asociar acciones a los eventos del modelo.
- Identificar las diferencias del modelo en distintos navegadores. Programar aplicaciones para que funcionen en los distintos navegadores.

Contenidos



1. El Modelo de Objetos del Documento (DOM).
2. Programación de eventos.
3. Diferencias en las implementaciones del modelo.
4. Aplicaciones Cross-Browser (multi-cliente).



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

- Es una de las innovaciones que más han influido en el desarrollo de paginas web dinámicas y de las aplicaciones web más complejas.
- Es una interfaz de programación de aplicaciones (API) de la plataforma de W3C, que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.
- Permite a los scripts acceder y actualizar dinámicamente su contenido, estructura y estilo de documento.



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

- Fue utilizado por primera vez con el navegador Netscape Navigator versión 2.0.
- A esta primera versión de DOM se le denomina DOM nivel 0 o modelo básico.
- El primer navegador de Microsoft que utilizó el DOM nivel 0 fue IE 3.0.
- Debido a las diferencias entre los navegadores, W3C emitió una especificación a finales de 1998 que llamó DOM nivel 1.
- En esta especificación ya se consideraba la manipulación de todos los elementos existentes en los archivos HTML y de XML.



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

- A finales del año 2000, W3C emitió DOM nivel 2, en la cual se incluía el manejo de eventos en el navegador y la interacción con hojas de estilo CSS.
- En 2004 se emitió DOM nivel 3, en la cual se utiliza la definición de tipos de documento (DTD) y la validación de documentos.



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

- Actualmente DOM se divide en tres partes o niveles según la W3C:
 - Núcleo del DOM (Core DOM): modelo estándar para cualquier documento estructurado. A este nivel se especifican las pautas para definir los objetos y propiedades de cualquier documento estructurado, así como para acceder a ellos.
 - XML DOM: modelo estándar para los documentos XML, define los objetos y propiedades de todos los elementos XML, así como para acceder a ellos.
 - **HTML DOM**: estándar para los documentos HTML. Interfaz de programación estándar para HTML independiente de la plataforma y el lenguaje. Define los objetos y propiedades de todos los elementos HTML, así como para acceder a ellos.



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

1.2.-Objetos del DOM HTML, propiedades y métodos

- Mediante el modelo DOM HTML, el navegador web transforma automáticamente todas las páginas web en una estructura de árbol, para que de esta manera se pueda acceder a los elementos de la página.
- Las páginas web son una sucesión de caracteres, por lo que resultaría excesivamente complicado manipular los elementos si no fuera por esta conversión.

1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

1.2.-Objetos del DOM HTML, propiedades y métodos

Sintaxis para acceder:

```
document.getElementById(objetoID).propiedad | metodo([parametros])
```

Listado de objetos del DOM en HTML				
Document	HTMLElement	Anchor	Area	Base
Body	Button	Evento	Form	Frame/iFrame
Input Hidden	Image	Input Button	Input Checkbox	Input File
Input Text	Link	Meta	Object	Option
Select	Style	Table	Table Cell	Table Row
textarea				



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

1.3.-Estructura del árbol DOM y tipos de nodos

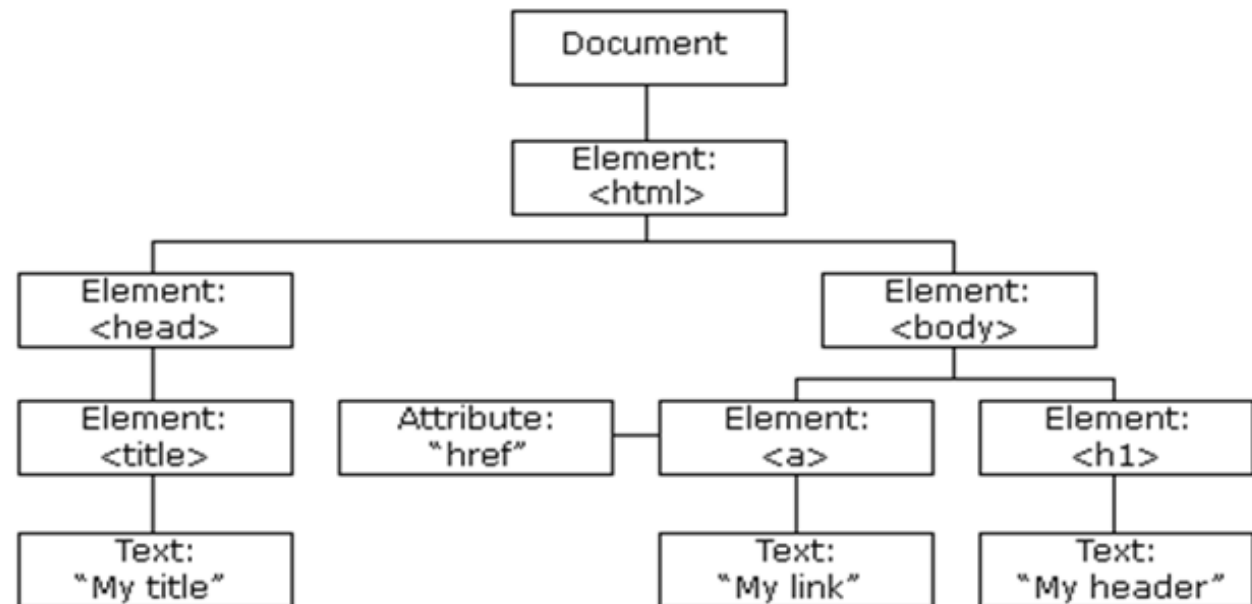
- DOM transforma los documentos HTML en elementos. Estos elementos se llaman [nodos](#). Cada nodo es un objeto.
- A su vez, los nodos están interconectados y muestran el contenido de la página web y la relación entre nodos.
- Cuando unimos todos estos nodos de forma jerárquica, obtenemos una estructura de árbol, que se le referencia como [árbol de nodos](#).

1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

- **Estructura del árbol DOM:**

- Dado el siguiente código HTML:

```
<html>
  <head>
    <title>My tittle</title>
  </head>
  <body>
    <a href="">My link</a>
    <h1>My header</h1>
  </body>
</html>
```



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

Tipos

- **Document**, es el nodo raíz y del que derivan todos los demás nodos del árbol.
- **Element**, representa cada una de las etiquetas XHTML. Es el único nodo que puede contener atributos y el único del que pueden derivar otros nodos.
- **Attr**, con este tipo de nodos representamos los atributos de las etiquetas XHTML, es decir, un nodo por cada atributo=valor.
- **Text**, es el nodo que contiene el texto encerrado por una etiqueta XHTML.
- **Comment**, representa los comentarios incluidos en la página XHTML.
- Los otros tipos de nodos pueden ser: **CdataSection**, **DocumentFragment**, **DocumentType**, **EntityReference**, **Entity**, **Notation** y **ProcessingInstruction**.

1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

1.4.- Acceso directo a los nodos:

Se puede hacer de dos métodos:

- a través de los nodos padre → partimos del nodo raíz y vamos accediendo a los nodo hijo y as sucesivamente hasta llegar al elemento deseado.
- usando el método de acceso directo → empleamos funciones del DOM, para ir directamente aun nodo sin pasar nodo a nodo.

Para que podamos acceder a los nodos, el árbol debe estar completamente construido.



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

- Para realizar la estructura del árbol, existe una serie de reglas:
 - En el árbol de nodos, al nodo superior (document), se le llama raíz.
 - Cada nodo, exceptuando el nodo raíz, tiene un padre.
 - Una hoja es un nodo sin hijos.
 - Los nodos que comparten el mismo padre, son hermanos.



1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

- Una vez que tenemos creada la estructura de árbol completa, ya disponemos de todos los nodos del árbol DOM.
- El estándar considera un nodo a cada una de las partes del árbol.
- A continuación veremos los tipos de nodos más importantes que pueden existir en un árbol DOM. Los tipos de nodos indicados son específicos del lenguaje XML, no obstante estos pueden ser aplicados a los lenguajes basados en XML como XHTML y HTML.

1.- EL MODELO DE OBJETOS DEL DOCUMENTO (DOM)

ACCESO DIRECTO A LOS NODOS

- Cuando se trabaja con una página web real, existen miles de nodos de todos los tipos en el árbol DOM. Así que no es eficiente acceder a un nodo descendiendo a través de todos los ascendentes.
- DOM añade una serie de métodos para acceder de forma directa a los nodos:
 - `getElementsByTagName()`
 - `getElementByName()`
 - `getElementById()`
 - `querySelector()`
 - `querySelectorAll()`

ACCESO DIRECTO A LOS NODOS

- **getElementsByTagName** (*nombre_etiqueta*) obtiene todos los elementos de la página cuya etiqueta HTML sea igual que el parámetro que se le pasa. Ejemplo:

```
var parrafos = document.getElementsByTagName ("p");
```

La función devuelve un array con todos los nodos que cumplen la condición de que su etiqueta coincide con el parámetro proporcionado. En realidad el array es una lista de nodos (nodeList), por tanto se debe procesar cada valor del array.

```
var primerParrafo = parrafos[0];
```

```
for (var i=0; i<parrafos.length; i++) {  
    var párrafo = parrafos [i];  
}
```

La función se puede aplicar de forma recursiva sobre cada uno de los nodos devueltos por la función:

```
var parrafos = document.getElementsByTagName ("p");
```

```
Var primerParrafo = párrafos[0];
```

```
Var enlaces = primerParrafo.getElementsByTagName ("a");
```

ACCESO DIRECTO A LOS NODOS

- **getElementsByName()** recupera todos los elementos de la página cuyo atributo `name` coincide con el parámetro pasado a la función.

```
var parrafoEspecial = document.getElementsByName("especial");
```

```
<p name="prueba">...</p>  
<p name="especial">...</p>  
<p>...</p>
```

Normalmente el atributo `name` es único para los elementos HTML que lo definen.

En el caso de elementos `radiobutton`, donde el atributo `name` es común a todos los `radiobutton` que están relacionados, devuelve un array `nodeList` con los elementos.

ACCESO DIRECTO A LOS NODOS

- **getElementById()** recupera el elemento HTML cuyo id coincide con el parámetro pasado a la función. Como el atributo id debe ser único para cada elemento, la función devuelve únicamente el nodo deseado. Es la función mas utilizada.

```
var cabecera = document.getElementById("cabecera");
```

```
<div id="cabecera">  
  <a href="URL" id="logo">...</a>  
</div>
```

ACCESO DIRECTO A LOS NODOS

- **querySelector()** acepta como parámetro un selector que identifica el elemento (o elementos) a seleccionar. En el caso de esta función, únicamente se devuelve el primer elemento que cumple la condición. Si no existe el elemento, el valor retornado es `null`.

```
var logo = document.querySelector (".enlace");
```

```
<div id = "cabecera">
```

```
    <a href="/" clas="enlace">...</a>
```

```
</div>
```

```
<div id = "cuerpo">
```

```
    <p>Lorem ipsu <a href="enlace">...</a></p>
```

```
</div>
```

- En este caso, a pesar de existir varios elementos de la clase `enlace`, únicamente es seleccionado el primero de ellos.

ACCESO DIRECTO A LOS NODOS

- **querySelectorAll()** acepta como parámetro un selector que identifica el elemento (o elementos) a seleccionar. Esta función devuelve un objeto de tipo `NodeList` con los elementos que coincidan con el selector.

```
var enlaces = document.querySelector (".enlace");  
<div id = "cabecera">  
    <a href="/" clas="enlace">...</a>  
</div>  
<div id = "cuerpo">  
    <p>Lorem ipsu <a href="enlace">...</a></p>  
</div>
```

Para acceder a los elementos almacenado en `NodeList`, recorreremos el objeto como si de un array se tratase.

```
for (var i=0; i<enlaces.length; i++)  
{  
    var enlaces = enlaces[i];  
}
```

ACCESO DIRECTO A LOS NODOS

- **Creación de elementos XHTML simples**

Consta de cuatro pasos:

1. Creación de un nodo `Element` que represente al elemento.
2. Creación de un nodo de tipo `Text` que represente el contenido del elemento.
3. Añadir el nodo `Text` como nodo de `Element`.
4. Añadir el nodo `Element` a la pagina, en forma de nodo hijo.

Si se quiere añadir un párrafo simple al final de la pagina, es necesario incluir el siguiente código:

```
// Crear el nodo element como hijo de la página
var párrafo = document.createElement ("p");
//Crear nodo de tipo Text
var contenido = document.createTextNode ("Hola mundo");
//Añadir el nodo Text como hijo del nodo Element
Parrafo.appendChild (contenido);
//Añadir el nodo Element como hijo de la página
document.body.appendChild (parrafo);
```

ACCESO DIRECTO A LOS NODOS

El proceso de creación de nuevos nodos implica la utilización de tres funciones DOM, por lo que puede resultar un poco tedioso:

`createElement(etiqueta)`: crea un nodo de tipo `Element` que representa al elemento XHTML cuya etiqueta se le pasa como parámetro.

`createTextNode(contenido)`: crea un nodo de tipo `Text` que almacena el contenido textual de los elemento XHTML.

`nodoPadre.appendChild(nodoHijo)`: añade un nodo como hijo de otro nodo. Se debe utilizar al menos dos veces con los nodos habituales, uno para añadir el nodo `Text` como hijo del nodo `Element` y a continuación para añadir el nodo `Element` como hijo de algún nodo de la página.

ACCESO DIRECTO A LOS NODOS

- **Inserción de elementos XHTML simples**

Se puede hacer de dos formas:

- 1.- Mediante el método **appendChild()**, lo cual lo inserta como elemento hijo del de referencia, pero al final de los elementos del padre.

```
padre = document.getElementById ("lateral");  
padre.appendChild(nuevo);
```

- 2.- Se basa en el método **insertBefore()**. Inserta el elemento nuevo delante de la etiqueta que hayamos seleccionado

```
var referencia = document.getElementById("insertar");  
var padre = referencia.parentNode;  
padre.insertBefore (nuevo, referencia);
```


ACCESO DIRECTO A LOS NODOS

- **Eliminación de elementos XHTML simples**

Se hace con la función `removeChild()`, esta función requiere como parámetro el nodo que se va a eliminar.

La función debe ser ejecutada desde el nodo padre.

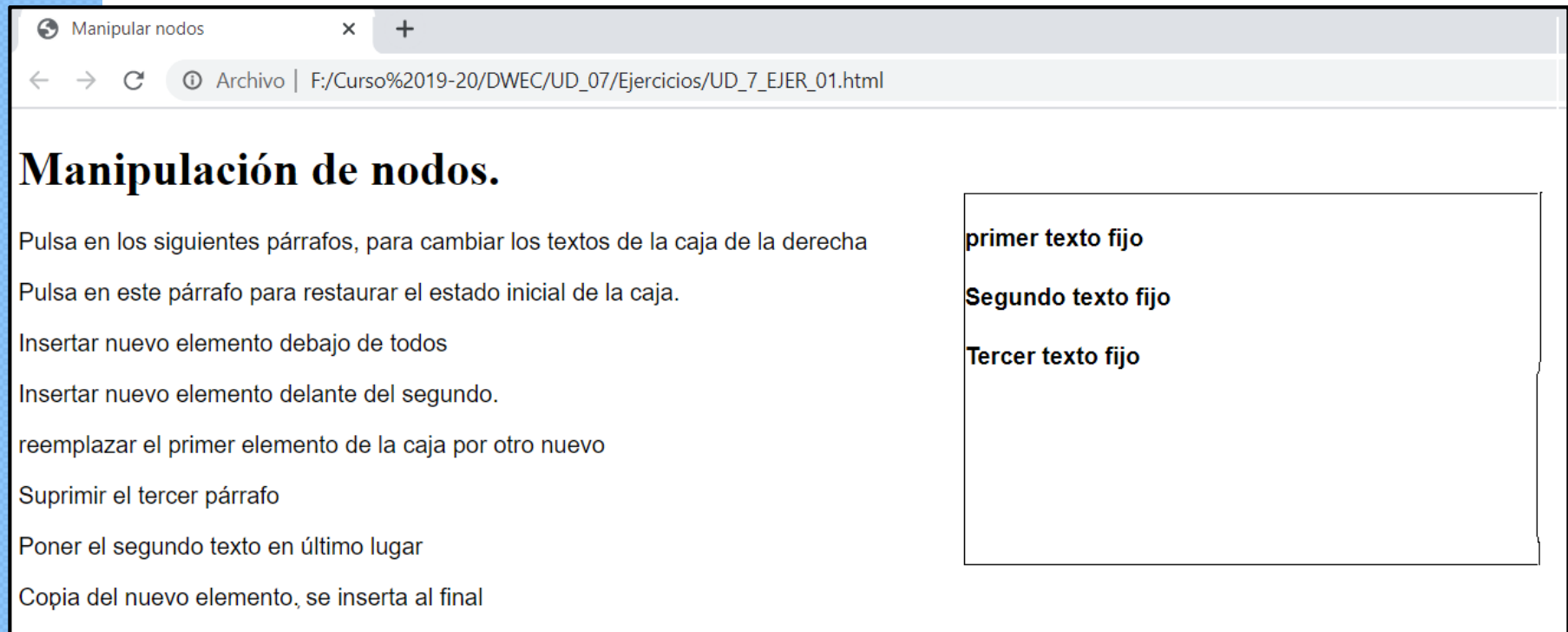
Para acceder al nodo padre se hace mediante la propiedad `nodoHijo.parentNode`.

Cuando se elimina un nodo, también se eliminan automáticamente todos los nodos hijos que tenga, por lo que no es necesario borrar manualmente cada nodo hijo.

Ejercicio_01 Manipular nodos

Ejercicio_01 Manipular nodos

Con todo lo visto hasta ahora de manipulación de nodos, crea una pagina web como la siguiente:



ACCESO DIRECTO A LOS ATRIBUTOS XHTML

Mediante DOM también es posible acceder de forma sencilla a los atributos XHTML y a todas las propiedades CSS de cualquier elemento de la página.

Los atributos XHTML de los elementos se transforman automáticamente en propiedades de los nodos.

```
var enlace = document.getElementById("enlace");  
console.log(enlace.href)
```

```
<a id="enlace" href=http://www.google.es>Enlace</a>
```

También podemos leer y escribir los atributos mediante los métodos **getAttribute(" ")**, para lectura; y **setAttribute(" ", " ")** para escritura.

ACCESO DIRECTO A LOS ATRIBUTOS XHTML

Ejemplo:

La lectura del atributo:

```
elemento = document.getElementsByTagName("a")[0];  
direccion = elemento.getAttribute("href");
```

La escritura del atributo:

```
elemento = document.getElementsByTagName("a")[0];  
elemento.setAttribute("href", http://google.es);
```

Mediante la escritura no solo se puede cambiar un valor de un atributo, sino que se pueden añadir nuevos atributos a las etiquetas.

ACCESO DIRECTO A LOS ATRIBUTOS XHTML

Acceso a las propiedades CSS:

Las propiedades CSS requieren un paso extra. Se debe utilizar el atributo **.style** y después el nombre de la propiedad.

```
var imagen = document.getElementById("parrafo");  
console.log(parrafo.style.fontWeight);
```

```
<p id="párrafo" style="font-weight: bold;"> ...</p>
```

La transformación del nombre de propiedades CSS compuesta se hace eliminando los guiones intermedios y escribiendo la siguiente letra en mayúscula.

Ejercicio_02 Acceso a atributos y propiedades nodos

Ejercicio_02 Acceso a atributos y propiedades nodos

Cuando hacemos clic sobre un texto lo cambia de color (rojo) y lo pone en negrita. Cuando pulsamos Enlace multinivel, va a la página del texto seleccionado.

