

Material de apoyo

DOM

Document Object Model

Tema: JavaScript

Fase: Desarrollo

Programa: Producción de Multimedia

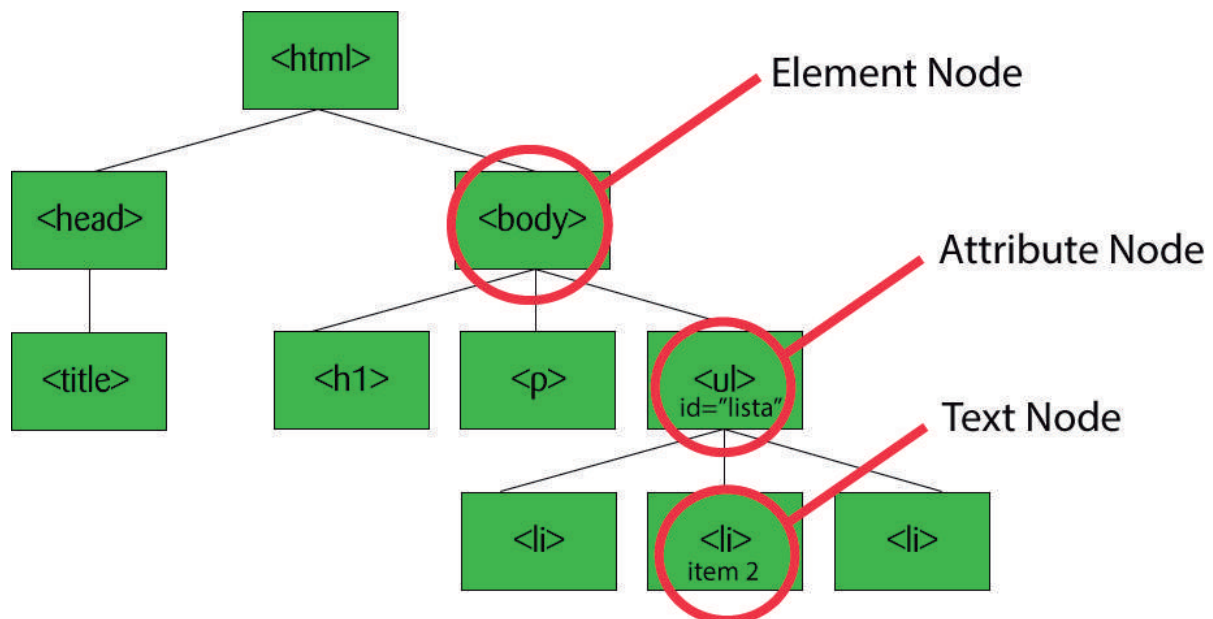
DOM

El DOM es la estructura de objetos que genera el navegador cuando se carga un documento HTML, mediante Javascript se puede cambiar dinámicamente los contenidos y aspecto de la página.

Con el Document Object Model JavaScript tiene el poder de realizar acciones como:

- Cambiar los elementos y atributos HTML en la página
- Cambiar los estilos CSS
- Agregar o Eliminar elementos y atributos HTML
- Animar elementos HTML
- Reaccionar a todos los eventos HTML existentes en la página
- Crear nuevos eventos.

El modelo del DOM en HTML se construye como un árbol de objetos:



► Árbol de nodos generado automáticamente por DOM a partir del código HTML de la página

Métodos DOM

- ▶ Los métodos son **acciones** que se pueden ejecutar. (Como añadir o borrar elementos HTML).
- ▶ Las propiedades son **valores** que se pueden obtener o ajustar. (Como cambiar el contenido de un elemento HTML).

Acceso a elementos del DOM

Una vez construido automáticamente el árbol completo de nodos DOM, ya es posible utilizar las funciones DOM para acceder de forma directa a cualquier nodo del árbol. Como acceder a un nodo del árbol es equivalente a acceder a "un trozo" de la página, una vez construido el árbol, ya es posible manipular de forma sencilla la página: acceder al valor de un elemento, establecer el valor de un elemento, mover un elemento de la página, crear y añadir nuevos elementos, etc.

getElementsByTagName()

La función **getElementsByTagName(nombreEtiqueta)** obtiene todos los elementos de la página HTML cuya etiqueta sea igual que el parámetro que se le pasa a la función.

El siguiente ejemplo muestra cómo obtener todos los párrafos de una página HTML:

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

El valor que devuelve la función es un array con todos los nodos que cumplen la condición de que su etiqueta coincide con el parámetro proporcionado.

De este modo, se puede obtener el primer párrafo de la página de la siguiente manera:

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

En el siguiente ejemplo se cambia el contenido de un elemento `<p>`:

```
<p class="demo"></p>

<script>
document.getElementsByTagName("demo")[0].innerHTML = "Hola Mundo!";
</script>
```

`getElementsByTagName` es el método mientras que `innerHTML` es la propiedad.

getElementsByTagName()

Es similar al anterior, pero en este caso se buscan los elementos cuyo atributo `name` sea igual al parámetro proporcionado. En el siguiente ejemplo, se obtiene directamente el único párrafo con el nombre indicado:

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

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

getElementById()

La función `getElementById()` es la más utilizada cuando se desarrollan aplicaciones web dinámicas. Esta función devuelve el elemento HTML cuyo atributo `id` coincide con el parámetro indicado en la función. Como el atributo `id` debe ser único para cada elemento de una misma página, la función devuelve únicamente el nodo deseado.

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

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

Acceso directo a los atributos HTML

Una vez que se ha accedido a un nodo, el siguiente paso natural consiste en acceder y/o modificar sus atributos y propiedades. Mediante DOM, es posible acceder de forma sencilla a todos los atributos HTML y todas las propiedades CSS de cualquier elemento de la página.

El siguiente ejemplo obtiene de forma directa la dirección a la que enlaza el enlace:

```
var enlace = document.getElementById("enlace");
alert(enlace.href); // muestra http://www...com

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

Para obtener el valor de una propiedad margin CSS de una imagen:

```


var imagen = document.getElementById("imagen");
alert(imagen.style.margin); // muestra 0px
```

Si el nombre de una propiedad CSS es compuesto, se accede a su valor modificando ligeramente su escritura:

```
var parrafo = document.getElementById("parrafo");
alert(parrafo.style.fontWeight); // muestra "bold"

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

La transformación del nombre de las propiedades CSS compuestas consiste en eliminar todos los guiones medios (-) y escribir en mayúscula la letra siguiente a cada guión medio. A continuación se muestran algunos ejemplos:

CSS	JAVASCRIPT
font-weight	fontWeight
line-height	lineHeight
border-top-style	borderTopStyle
list-style-image	listStyleImage

Otros métodos DOM para modificar elementos HTML:

Método	Descripción
<i>elemento.innerHTML = nuevo contenido HTML</i>	Cambia el HTML interno de un elemento
<i>elemento.attribute = nuevo valor</i>	Cambia el valor de atributo de un element HTML
<i>elemento.setAttribute(atributo, valor)</i>	Cambia el valor de atributo de un element HTML
<i>elemento.style.propiedad = nuevo estilo</i>	Cambia el estilo de un elemento HTML