



APLICACIONES WEB CON HTML5, CSS3 Y JAVASCRIPT

LUIS BASTO DÍAZ



Javascript

Objetos DOM

El Modelo de Objetos de Documentos (Document Object Model) representan a los elementos que pueden ser accedidos del documento HTML.

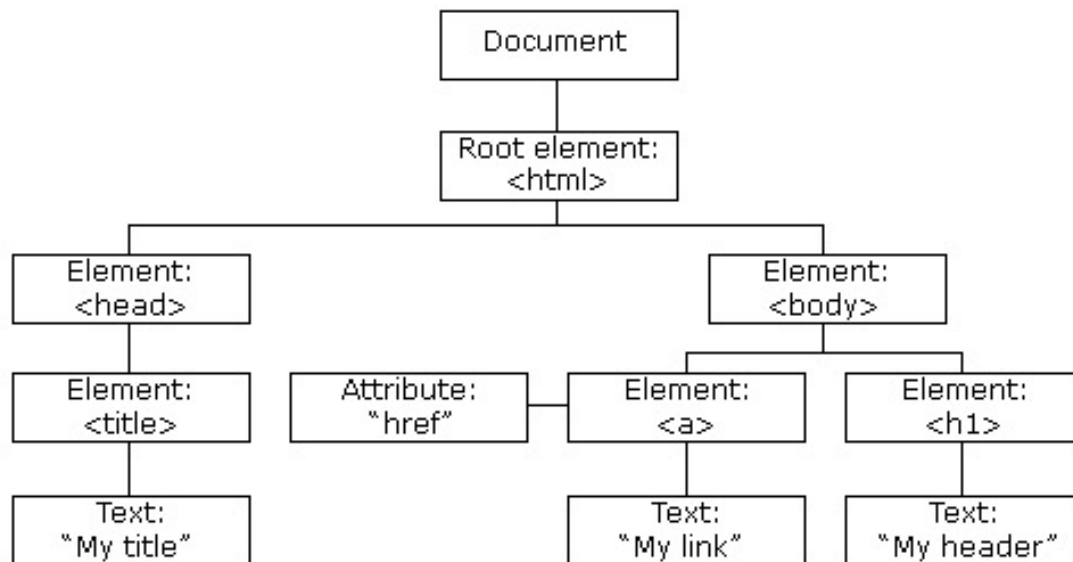
Representa una jerarquía de objetos que forma un modelo del documento HTML.

El DOM transforma los documentos HTML en un conjunto de elementos llamados nodos, que están interconectados y que representan los contenidos de las páginas web y las relaciones entre ellos.

Objetos DOM

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>My title</title>
  </head>
  <body>
    <a href="#">My Link</a>
    <h1>My Header</h1>
  </body>
</html>
```

Objetos DOM



Objetos DOM

La raíz del árbol es el nodo: Document.

Cada etiqueta se transforma en:

- Un nodo Element
- Nodo Text
- Eje: <title>Página sencilla</title>



Tipos de Nodos

Nodo	Descripción
Document	Nodo raíz del documento
Element	Cada una de las etiquetas. Puede contener atributos y otros nodos.
Attr	Representa a los atributos de las etiquetas. Uno por cada atributo-valor.
Text	Texto encerrado entre las etiquetas.
Comment	Comentarios incluidos en la página.

Accediendo objetos DOM

Algunos elementos del DOM regresan un solo elemento, otros regresan un arreglo de elementos.

Para los que regresan un arreglo, puede ser un arreglo dinámico o un arreglo estático de Lista de Nodos (NodeList).

- Dinámico: constantemente actualizado por el DOM.
- Estático: Representan una fotografía de elementos que no cambian.

Accediendo objetos DOM

Métodos de búsqueda DOM

`getElementById`:

- Regresa una referencia al primer objeto especificado en el id.
- `var btn = document.getElementById('btnSave');`

`getElementsByTagName`

- Regresa un `NodoLista` dinámico, esto es un arreglo con todas las etiquetas con atributo nombre especificado.
- `var imgs = document.getElementsByTagName('img');`

Accediendo objetos DOM

Métodos de búsqueda DOM

getElementsByTagName:

- Regresa un NodoLista dinámico, con el nombre especificado.
- Trabaja bien con los botones de opción cuando todos los botones tienen el mismo nombre.
- `var colores= document.getElementsByTagName('color');`

getElementsByClass

- Regresa un NodoLista dinámico con el nombre de la clase CSS especificada.
- Trabaja bien cuando se tienen muchos elementos, pero se necesita agruparlos (mostrarlos u ocultarlos).
- `var colores= document.getElementsByClassName('color');`

Manipulando objetos DOM

Para modificar un elemento en una página

1. Crear un nuevo objeto que contenga el nuevo dato.
2. Encontrar el elemento padre que debe contener el nuevo dato.
3. Añade, inserta o reemplaza el dato en el elemento con el nuevo dato.

Manipulando objetos DOM

Para remover un elemento o atributo en una página

1. Encontrar el elemento padre.
2. Usar `removeChild` o `removeAttribute` para remover el dato.

Manipulando objetos DOM

Para crear nuevo objeto para un documento

- `document.createElement(tagName)`
- `document.createTextNode(string)`
- `document.createAttribute(name,value)`
- Para encontrar el elemento padre
 - `document.getElementById()`

Manipulando objetos DOM

Métodos para insertar o reemplazar el objeto.

- `appendChild(newNode)`
- `insertBefore(newNode, existingNode)`
- `replaceChild(newNode, existingNode)`
- `replaceData(offset, length, string)`

Práctica 06

Utilice la página `eje01_dom.html` y con las funciones DOM, mostrar mediante funciones `alert()`, la siguiente información.

Número de enlaces de la página.

URL a la que enlaza el penúltimo enlace.

Numero de enlaces que enlazan a `http://prueba`

Número de enlaces del tercer párrafo.

añada la información de salida en un div dentro de la misma página html.

- `<div id=info> </div>`
- `info.innerHTML = info.innerHTML + variable;`

Tip: encierre todo su código en una función anónima:

- `window.onload = function() { }`

Práctica 07

Utilice la página eje02_dom.html y con las funciones DOM, realizar lo siguiente.

Añada nuevos elementos a la lista.

- Hágalo de las dos maneras, utilizando innerHTML y createTextNode

A cada elemento de la lista agregue un enlace “eliminar” para que al dar click sobre él, se elimine dicho elemento.

Nota: Sólo se debe saber que al hacer click al botón se debe ejecutar el botón.

Trabajando con Eventos

Los eventos hacen posible que el usuario transmita información a la aplicación.

Los scripts esperan a que el usuario “haga algo”

- Pulsar una tecla
- Mover el puntero del ratón
- Cerrar una ventana

Los scripts responden a la acción del usuario, procesando la información y generando el resultado.

Trabajando con Eventos

JavaScript permite definir varios eventos.

Javascript permite asignar una función a cada evento y ejecuta la función asociada.

A las funciones asociadas se les denomina “event handlers”.

Existen varios modelos de eventos

- Modelo de eventos básico
- Modelo de eventos estándar
- Modelo de eventos de IE

Modelo básicos de Eventos

El nombre del evento tiene el prefijo **on** seguido del nombre de la acción del evento.

Evento	Descripción	Elementos
onchange	Deseleccionar un elemento que se ha modificado	<input>, <select>, <textarea>
onclick	Dar click	Todos
onfocus onblur	Seleccionar un elemento Deseleccionar un elemento	<button>, <input>, <label>, <select>, <textarea>, <body>
onload	La página se ha cargado completamente	<body>

Modelo básicos de Eventos

Evento	Descripción	Elementos
onmousemove onmousedown onmouseout onmouseover onmouseup	Mover ratón Pulsar un botón del ratón El puntero sale del elemento El puntero entra en el elemento Soltar el botón	Todos los elementos
onselect	Seleccionar un texto	<input>, <textarea>
onsubmit	Enviar un formulario	<body>

Manejadores de Eventos

Formas de indicar los manejadores:

- Como atributos de los elementos
- Como funciones JavaScript
- Manejadores Semánticos

Manejadores de Eventos

Como atributos de los elementos

- `<input type="button" value="Click" onclick="alert('Hola');" />`

- `<div onclick="alert('Click');" onmouseover="alert('Pasar el ratón por encima');">`

Puedes hacer click sobre este elemento o simplemente pasar el ratón por encima

`</div>`

Manejadores de Eventos

Como funciones JavaScript

- consiste en extraer todas las instrucciones de JavaScript y agruparlas en una función externa.
- No se puede utilizar la variable `this`, es necesario pasar esta variable como parámetro a la función.
- Ver ejemplo `eje01_even.html`

Manejadores de Eventos

Manjedores Semánticos

- La idea principal es separar los contenidos (HTML) con el comportamiento (JavaScript).
- Se basa en utilizar las propiedades DOM de los elementos para asignar todas las funciones externas que actúan como manejadores de eventos.
- Ver `eje02_even.html`

Manejadores de Eventos

Manejadores Semánticos consiste en:

1. Asignar un id (único) al elemento (atributo id)
2. Crear la función JavaScript del evento
3. Asignar la función al evento correspondiente en el elemento deseado.

- Ver `eje03_even.html`

Práctica 08

Utilice el archivo eje04_even.html y realice lo siguiente mediante funciones javascript y manejador semántico:

1. Cuando se de click al primer enlace, se oculte su párrafo relacionado.
2. Cuando se vuelva a dar click sobre el mismo enlace, se muestre otra vez el párrafo relacionado.
3. Cuando un párrafo se oculte, debe cambiar el mensaje del enlace asociado a Mostrar Párrafo (tip: pruebe la propiedad innerHTML).
4. Guarde cada archivo como:
 1. eje04_even_sol1.html
 2. eje04_even_sol2.html