cedia | TMO Training Management Office

info@cedia.org.ec
cedia.edu.ec

CURSO DE JAVASCRIPT

Corporación Ecuatoriana para el Desarrollo de la Investigación y la Academia

Ing. María Fernanda Granda J., PhD

1-12-2025

“Conectando ideas, transformando sociedades”

Módulo 7: Trabajar con el Modelo Objeto Documento (DOM)

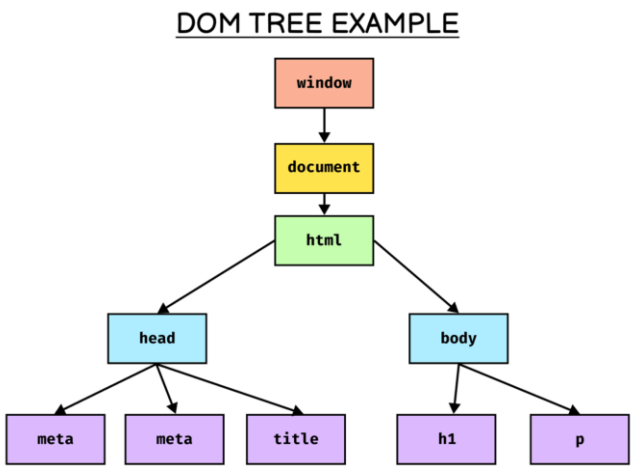
- Acceso al DOM-HTML.
- Modificación del DOM-HTML.

¿Qué es el Modelo de Objetos de Documento (DOM)?

- El DOM (Document Object Model) es un API con la representación estructurada de documentos HTML y XTML.
- JavaScript puede acceder y modificar cualquier parte del DOM para cambiar el contenido, estilo o comportamiento.
- El modelo DOM surgió como una especificación para permitir que los programas Java y los scripts de JavaScript fueran portables entre los navegadores web.

Características del DOM

- **Representación Árbol:** La página web se representa como un árbol de nodos, donde cada etiqueta HTML es un nodo de elemento, y el texto es un nodo de texto.
- **Manipulación Dinámica:** Los scripts pueden usar el DOM para agregar, eliminar, modificar y actualizar los elementos de la página en tiempo real.

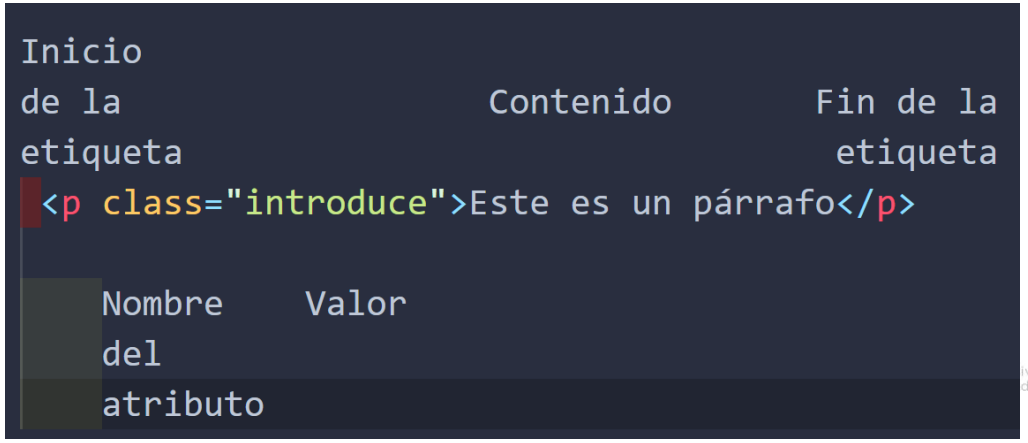


Estructura del DOM

- La estructura del DOM está integrada por **objetos (nodos)**, que se relacionan unos con otros en una estructura jerárquica, y cada objeto tiene un objeto superior, que es el objeto **padre**. Además, cada nodo puede tener 0, 1 o varios nodos dependientes de él, llamados nodos **hijos**.
- Los nodos que están a un mismo nivel, dependiendo todos ellos del mismo nodo padre, son nodos **hermanos**.
- Todo el árbol del DOM depende de un nodo principal a partir del cual se generan todos, el objeto **Document**.

Estructura del DOM

- Todas las propiedades, métodos y eventos son organizados en objetos.
- Los objetos son accedidos a través de lenguajes de programación y scripts.
- Java script usa el DOM API (implementación nativa para cada browser).



DOM API

- Consiste de objetos y métodos para interactuar con la página HTML
  - Puede agregar y remover elementos HYML
  - Puede aplicar estilos dinámicamente
  - Puede agregar y remover atributos HTML
- DOM introduce objetos que representan los elementos de HTML y sus propiedades
  - document.documentElement es <html>
  - document.body es el cuerpo de la página

Objetos del DOM

- Cada elemento HTML tiene un correspondiente tipo de objeto DOM
  - HTMLLIElement represents
  - HTMLAudioElement represents <audio>
- Cada uno de esos objetos tiene apropiadas propiedades
  - HTMLAnchorElement tiene la propiedad href
  - HTMLImageElement tiene la propiedad src
- El objeto documento es un objeto especial
  - Representa el punto de entrada para el API de DOM

Elementos HTML

- Los elementos HTML tienen propiedades que corresponden a sus atributos
  - Id, className, draggable, style, onclick, etc.
- Diferentes elementos HTML tienen sus atributos específicos
  - HTMLImageElement tiene la propiedad src
  - HTMLInputElement tiene la propiedad value
  - HTMLAnchorElement tiene la propiedad href

Elementos HTML

- Los elementos de HTML tiene propiedades correspondientes a su contenido
  - **innerHTML:** retorna como un string el contenido del elemento, sin el elemento
  - **outnerHTML:** retorna como un string el contenido del elemento, con el elemento.
  - **innerText/textContent:** retorna como un string el contenido del texto, sin las etiquetas.

Seleccionando los elementos del DOM

- Los elementos de HTML pueden ser encontrados y capturados en variables usando el API del DOM
  - **Selecciona un simple elemento:**
  - let cabecera= document.getElementById('header');
  - let nav= document.querySelector('#main-nav');
  - **Selecciona una colección de elementos:**
  - let entradas= document.getElementByTagName('li');
  - let radiosGrupo= document.getElementByName('generos[]');
  - let cabecera= document.querySelectorAll('#main-nav li');
  - **Usando colecciones predefinidas de elementos:**
  - let enlaces= document.links;
  - let formas= document.forms;

Usando los métodos getElementsBy

- El API del DOM contiene métodos para seleccionar elementos basados en algunas características:
  - **ById**
    - let cabecera= document.getElementById('header');
  - **ByClassName**
    - let posts= document.getElementByClassName('post-item');
  - **ByTagName**
    - let barraLateral = document.getElementByTagName('barra');
  - **ByName**
    - let grupoGenero= document.getElementByName('generos[]');

Ejemplo 1 de Modificación del DOM-HTML

- Selección y **modificación básica** del DOM
- **Objetivo**: Cambiar el texto de un párrafo al hacer clic en un botón.

```html
<html lang="es">
<head>
</head>
<body>
  <h2 id="titulo">Título original</h2>
  <button onclick="cambiarTitulo()">Cambiar título</button>

  <script src="codigo.js"> </script>
</body>
</html>
```

```javascript
function cambiarTitulo() {
// Accedemos al elemento por su ID y cambiamos su contenido
  document.getElementById("titulo").textContent = "🎉 Título
  actualizado desde JavaScript";
}
```

Ejemplo 2 de Modificación del DOM-HTML

- Acceso a **atributos** de elementos HTML
- **Objetivo**: Mostrar la URL de un enlace desde JavaScript

```html
<a id="miEnlace" href="https://www.google.com">Ir a Google</a>
<button onclick="mostrarHref()">Mostrar enlace</button>
```

```javascript
function mostrarHref() {
  let enlace = document.getElementById("miEnlace");
  alert("La URL del enlace es: " + enlace.href);
}
```

Ejemplo 3 de Modificación del DOM-HTML

- Modificar **estilos** dinámicamente
- **Objetivo**: Cambiar el color de un texto con classList

```html
<p id="texto">Este es un texto con estilo</p>
<button onclick="resaltar()">Resaltar</button>
```

```javascript
function resaltar() {
  document.getElementById("texto").classList.toggle("resaltado");
}
```

```css
.resaltado {
  color: white;
  background-color: darkblue;
  padding: 5px;
}
```

Ejemplo 4 de Modificación del DOM-HTML

- **Crear elementos dinámicamente** (crear un <li>)
- Objetivo: Agregar ítems a una lista usando el DOM

```html
<ul id="lista"></ul>
<input type="text" id="item" placeholder="Escribe un artículo">
<button onclick="agregarItem()">Agregar</button>
```

```javascript
function agregarItem() {
  let texto = document.getElementById("item").value;
  if (texto.trim() !== "") {
    let li = document.createElement("li");
    li.textContent = texto;
    document.getElementById("lista").appendChild(li);
    document.getElementById("item").value = "";
  }
}
```

Ejemplo 5 de Modificación del DOM-HTML

Realizar un gestor de tareas, que permita:

- Ingresar tareas pendientes
- Eliminar tarea individual
- Eliminar todas las tareas
- Resaltar las tareas

cedia | TMO Training Management Office

GRACIAS

cedia.edu.ec
info@cedia.org.ec
(+593) 7-407-9300