

A yellow square containing the letters 'JS' in a large, bold, black sans-serif font.

# JS

A wide horizontal band with a grayscale, textured background. The word 'JAVASCRIPT' is centered in a large, bold, black, distressed serif font. Below it, on the left, is the text 'Tema 4 – DOM' in a smaller, black, sans-serif font.

# JAVASCRIPT

Tema 4 – DOM

# JAVASCRIPT

JS

- DOM: Árbol de nodos
  - “El modelo de objeto de documento (DOM) es una interfaz de programación para los documentos HTML y XML. Facilita una representación estructurada del documento y define de qué manera los programas pueden acceder, al fin de modificar, tanto su estructura, estilo y contenido. El DOM da una representación del documento como un grupo de nodos y objetos estructurados que tienen propiedades y métodos. Esencialmente, conecta las páginas web a scripts o lenguajes de programación.”

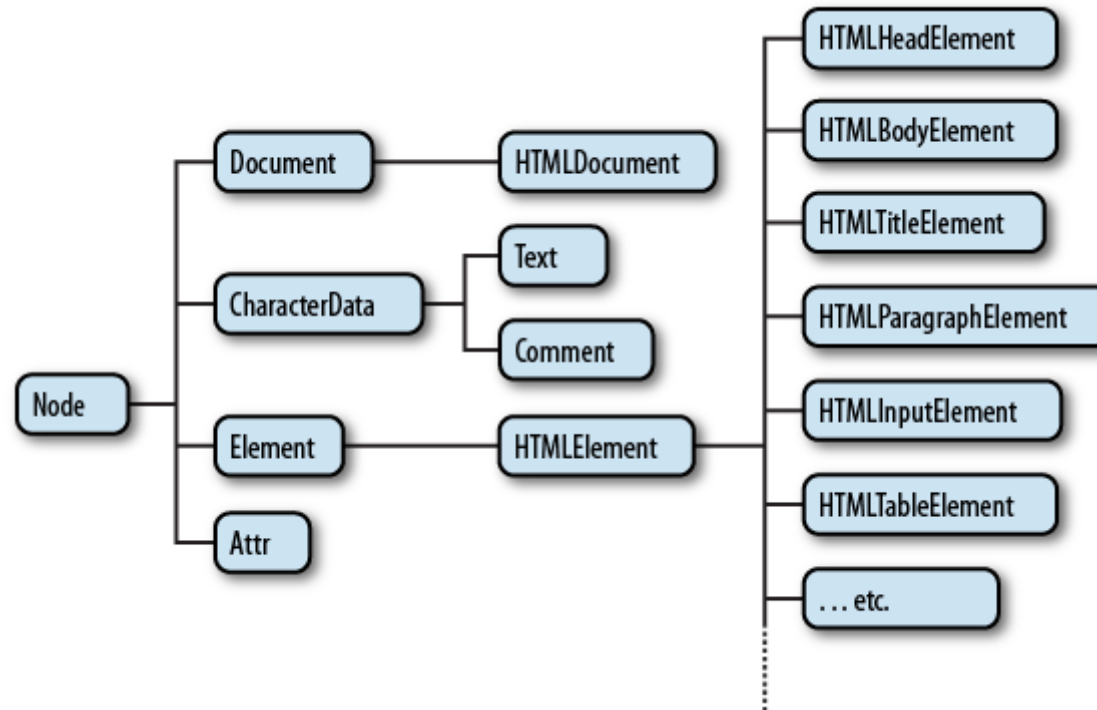
Fuente: [https://developer.mozilla.org/es/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/es/docs/Web/API/Document_Object_Model/Introduction)

# JAVASCRIPT

JS

- Interfaces más relevantes.
  - **Node**. Es la clase base de todas las interfaces del DOM.
  - **Element**. Es la clase base de todos los elementos definidos con etiquetas HTML.
  - **Document**. El nodo raíz. Representa una página cargada en el navegador.
  - **Attr**. Atributos asociados a los elementos.
  - **CharacterData**. Elemento que contiene texto.
    - **Text**. El texto contenido dentro de un Element.
    - **Comment**. Comentarios de la página.
  - **DocumentFragment**. Es una representación ligera de un Document.
  - **NodeList**. Colección de nodos.
  - **NamedNodeMap**. Colección de atributos (Attr).
  - **Event**. Representa un evento.

# JAVASCRIPT



# JAVASCRIPT



- Acceso directo a los nodos.
  - A partir del objeto document.
  - Método: getElementById.//Id
  - Método: getElementsByName.//Class
  - Método: getElementsByTagName.//Etiqueta
  - Método: getElementsByTagNameNS.//Etiqueta y NameSpace

# JAVASCRIPT



- Acceso directo a los nodos.
  - A partir del objeto document.
  - Método: querySelector
  - Método: querySelectorAll
  - Selectores:
    - [https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_selectors](https://developer.mozilla.org/es/docs/Web/CSS/CSS_selectors)
    - <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-classes>
    - <https://developer.mozilla.org/es/docs/Web/CSS/Pseudo-elements>

# JAVASCRIPT



- Creación y eliminación
  - A partir del objeto document.
  - Método: createElement
  - Método: createAttribute
  - Método: createTextNode
  - Método: createComment
  - Método: createDocumentFragment

# JAVASCRIPT

JS

- Ejemplo de uso de DocumentFragment:

```
<body>
  <div id="div-principal"></div>
</body>
</html>
<script>
  let fragmento = document.createDocumentFragment();
  let elemento = document.createElement("h1");
  elemento.appendChild(document.createTextNode("Contenido"));
  fragmento.appendChild(elemento);
  document.querySelector("#div-
principal").appendChild(fragmento);
</script>
```



# JAVASCRIPT

A yellow square containing the letters 'JS' in a bold, black, sans-serif font.

- Acceso directo a los atributos.
  - Las características de JavaScript permiten crear y acceder a los atributos de manera directa:
    - `elemento.atributo = "valor";`

# JAVASCRIPT

JS

- Acceso directo a los atributos.

- Ejemplo:

```
<body>
  
</body>
</html>
<script>
  setTimeout (()=>{
    document.querySelector("#imagen").width="200";
  }, 1000);
</script>
```

# JAVASCRIPT

JS

- Acceso directo a los atributos.
  - Atributo style: permite modificar el estilo, como CSS.
  - A través de este objeto se pueden acceder a todas las propiedades del CSS:

- [https://www.w3schools.com/jsref/dom\\_obj\\_style.asp](https://www.w3schools.com/jsref/dom_obj_style.asp)

- Ejemplo:

```
let texto = document.querySelector("#texto");  
texto.style.color = "#FF0000";
```

# JAVASCRIPT

JS

- Asociación de clases.
  - El atributo **classList** mantiene la lista de clases asociadas a un elemento.
  - Mediante los métodos del objeto **add** y **remove** se puede asignar y desasignar clases en tiempo de ejecución.