



Gestión del DOM

Centro de Servicios y Gestión Empresarial
SENA Regional Antioquia



www.sena.edu.co

Contextualización

DOM significa *Document Object Model* (Modelo de Objetos del Documento).

- Es una representación estructurada en forma de árbol que el navegador crea a partir del código HTML de una página web.
- Cada etiqueta HTML, cada texto, cada atributo, se convierte en un nodo (objeto) que puede ser accedido, modificado o eliminado con JavaScript.

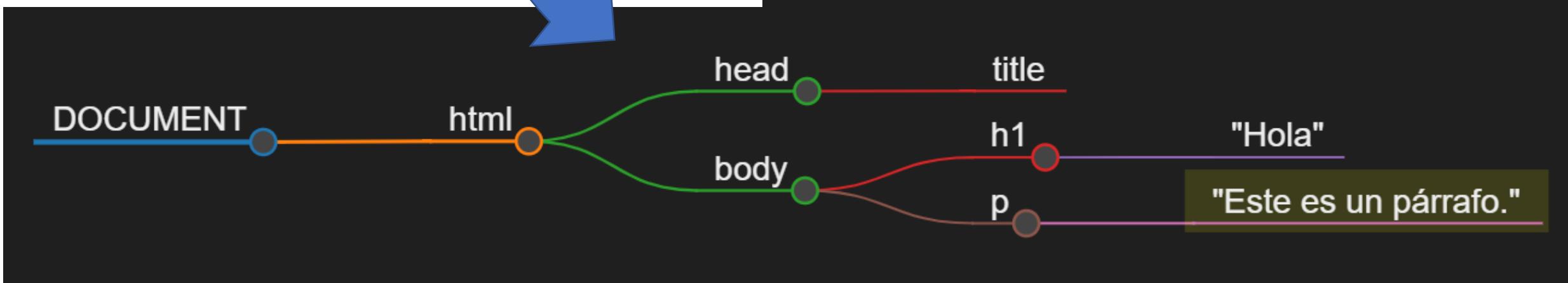
¿Cómo funciona el DOM?

Cuando cargas una página HTML, el navegador:

1. Lee el archivo HTML.
2. Crea un árbol de nodos que representa todos los elementos del documento.
3. Este árbol es el DOM: una interfaz que JavaScript puede usar para leer y manipular la estructura, el contenido y el estilo de la página.

DOM

```
<html>
  <head>
  </head>
  <body>
    <h1>Hola</h1>
    <p>Este es un párrafo.</p>
  </body>
</html>
```



Tipos de nodos en el DOM

Tipo de nodo	Ejemplo
Documento (document)	La raíz de todo el DOM
Elemento (<p>, <div>)	Etiquetas HTML como nodos
Texto	El contenido dentro de etiquetas
Atributo	class="rojo", id="titulo"
Comentario	<!-- esto es un comentario -->

¿Por qué es importante el DOM?

Porque es la puerta de entrada para manipular el contenido de la web con JavaScript.

Con el DOM puede:

- **Cambiar** texto e imágenes.
- **Agregar** o **eliminar** elementos.
- **Escuchar** eventos del usuario (clic, teclado).
- **Modificar** estilos CSS dinámicamente.
- **Crear** interfaces interactivas (menus, sliders, formularios, etc.).

Elementos del DOM

Acceder a Elementos del DOM



Método	Descripción	Ejemplo
getElementById()	Selecciona un elemento por su ID	<code>document.getElementById("titulo")</code>
getElementsByClassName()	Selecciona elementos por clase	<code>document.getElementsByClassName("item")</code>
getElementsByTagName()	Selecciona elementos por etiqueta	<code>document.getElementsByTagName("p")</code>
querySelector()	Selecciona el primer elemento que coincide con un selector CSS	<code>document.querySelector(".boton")</code>
querySelectorAll()	Selecciona todos los elementos que coincidan	<code>document.querySelectorAll("li")</code>

Manipulación de Elementos del DOM



Propiedad	Descripción	Ejemplo
innerHTML	Permite modificar el contenido de un elemento HTML	<code>document.getElementById(<i>id</i>).innerHTML = <i>new HTML</i></code>
write()	Permite escribir directamente en el flujo de salida HTML	<code>document.write("valor");</code>

Manipular Contenido y Atributos



```
// Manipular Contenido y Atributos
// Modificar texto:
document.getElementById("titulo").textContent = "El Increible Mundo de la Natación";
document.getElementById("subtitulo").innerHTML = "Definición:";

// Cambiar atributos:
document.querySelector("img").src = "multimedia/nataciontres.jpg";
document.getElementById("imagendos").setAttribute("src", "multimedia/nataciontres.jpg");

// Cambiar estilos:
document.getElementById("titulo").style.color = "red";
document.querySelectorAll("p").forEach(function(parrafo) {
    parrafo.style.color = "blue";
    parrafo.style.fontSize = "20px";
});
```



EXAMPLE



Crear, insertar y eliminar elementos



```
// Crear un nuevo elemento y agregarlo al DOM
function crearElemento() {
    // Crear un nuevo elemento
    var nuevoElemento = document.createElement("p");
    // Asignar texto al nuevo elemento
    nuevoElemento.textContent = "Este es un nuevo párrafo creado
dinámicamente.";
    // Insertar el nuevo elemento en el DOM
    document.getElementById("contenido").appendChild(nuevoElemento);
    alert("Elemento creado e insertado correctamente");
}
```

Crear, insertar y eliminar elementos



```
// Eliminar un elemento del DOM
function eliminarElemento() {
    // Seleccionar el elemento a eliminar
    var elementoAEliminar = document.getElementById("contenido");
    // Verificar si el elemento existe antes de intentar eliminarlo
    if (elementoAEliminar) {
        elementoAEliminar.parentNode.removeChild(elementoAEliminar);
        alert("Elemento eliminado correctamente");
    } else {
        alert("El elemento no existe en el DOM.");
    }
}
```

Eventos del DOM

Eventos del DOM



Un evento en JavaScript es cualquier acción del usuario o del sistema que puede ser detectada por el navegador, como:

- Un clic
- Presionar una tecla
- Mover el mouse
- Enviar un formulario
- Cambiar el valor de un input
- Cargar la página

El DOM permite "escuchar" estos eventos y responder con funciones llamadas manejadores de eventos.

Eventos del DOM



Ejemplo:

```
<button id="miBoton">Haz clic aquí</button>

<script>
  document.getElementById("miBoton").addEventListener("click", function () {
    alert("¡Hiciste clic!");
  });
</script>
```

Eventos en el DOM



Tipo	Evento	Se dispara cuando...
Mouse	click	Se hace clic en un elemento
Mouse	mouseover	El puntero entra al elemento
Mouse	mouseout	El puntero sale del elemento
Teclado	keydown	Se presiona una tecla
Teclado	keyup	Se suelta una tecla
Formulario	submit	Se envía un formulario
Formulario	change	Cambia el valor de un input/select
Carga	load	Termina de cargar la página o imagen

Eventos en el DOM



Buenas prácticas:

- ✓ Usar ***addEventListener()*** en lugar de onclick, onchange, etc.
- ✓ Mantener las funciones separadas para mayor claridad
- ✓ Usar ***event.preventDefault()*** si necesitas evitar el comportamiento por defecto (por ejemplo, en formularios)

Eventos en el DOM → mouse / clic



```
<form action="google.com" method="get" id="formulario">
  <input id="numid" type="text" placeholder="Número de identificación" required>
  <input type="submit" value="Validar" id="validar">
  <p id="mensaje"></p>
</form>
```



```
// Agregar un evento al hacer clic en un botón
document.getElementById("validar").addEventListener("click",(event) => {
  // Prevenir el comportamiento por defecto del formulario
  event.preventDefault();
  // Obtener el valor del campo de entrada
  let inputnumid = Number(document.getElementById("numid").value);
  let texterror;
  document.getElementById("mensaje").innerHTML = "";
  // Validar el número ingresado
  if (isNaN(inputnumid) || inputnumid < 1 || inputnumid > 10) {
    texterror = "Error: El número debe ser un valor entre 1 y 10.";
    document.getElementById("mensaje").style.color = "red";
    document.getElementById("mensaje").innerHTML = texterror;
  } else {
    texterror = "El número es correcto.";
    document.getElementById("mensaje").style.color = "green";
    document.getElementById("mensaje").innerHTML = texterror;
  }
});
```

Eventos en el DOM → mouse / clic



```
// Agregar un evento al hacer clic en un botón
document.getElementById("validar").addEventListener("click",(event) => {
    // Prevenir el comportamiento por defecto del formulario
    event.preventDefault();
    // Obtener el valor del campo de entrada
    let inputnumid = Number(document.getElementById("numid").value);
    let texterror;
    document.getElementById("mensaje").innerHTML = "";
    // Validar el número ingresado
    if (isNaN(inputnumid) || inputnumid < 1 || inputnumid > 10) {
        texterror = "Error: El número debe ser un valor entre 1 y 10.";
        document.getElementById("mensaje").style.color = "red";
        document.getElementById("mensaje").innerHTML = texterror;
    } else {
        texterror = "El número es correcto.";
        document.getElementById("mensaje").style.color = "green";
        document.getElementById("mensaje").innerHTML = texterror;
    }
});
```

Ejercicio de Aplicación



Generar un aplicación tipo To-Do List (lista de tareas) usando HTML, CSS y JavaScript. En esta aplicación debe aplicar los conceptos de DOM (Document Object Model) para agregar, eliminar y marcar tareas como completadas.

Características:

- Agregar tareas dinámicamente.
- Eliminar tareas.
- Marcar tareas como completadas.
- Aplicar Bootstrap para que se vea bonito.
- Usar funciones para modularizar el código.

Eventos en el DOM → formulario / change



```
<form action="google.com" method="get" id="formulario">
  <input id="numid" type="text" placeholder="Número de identificación" required>
  <input type="submit" value="Validar" id="validar">
  <p id="mensaje"></p>
</form>
```



```
// Agregar un evento a los cambios de los inputs del formulario
document.getElementById("numid").addEventListener("input", function() {
  // Obtener el valor del campo de entrada
  let inputnumid = Number(document.getElementById("numid").value);
  let texterror;
  document.getElementById("mensaje").innerHTML = "";
  // Validar el número ingresado
  if (isNaN(inputnumid) || inputnumid < 1 || inputnumid > 10) {
    texterror = "Error: El número debe ser un valor entre 1 y 10.";
    document.getElementById("mensaje").style.color = "red";
    document.getElementById("mensaje").innerHTML = texterror;
  } else {
    texterror = "El número es correcto.";
    document.getElementById("mensaje").style.color = "green";
    document.getElementById("mensaje").innerHTML = texterror;
  }
});
```

Eventos en el DOM → formulario / change



```
// Agregar un evento a los cambios de los inputs del formulario
document.getElementById("numid").addEventListener("input", function() {
    // Obtener el valor del campo de entrada
    let inputnumid = Number(document.getElementById("numid").value);
    let texterror;
    document.getElementById("mensaje").innerHTML = "";
    // Validar el número ingresado
    if (isNaN(inputnumid) || inputnumid < 1 || inputnumid > 10) {
        texterror = "Error: El número debe ser un valor entre 1 y 10.";
        document.getElementById("mensaje").style.color = "red";
        document.getElementById("mensaje").innerHTML = texterror;
    } else {
        texterror = "El número es correcto.";
        document.getElementById("mensaje").style.color = "green";
        document.getElementById("mensaje").innerHTML = texterror;
    }
});
```

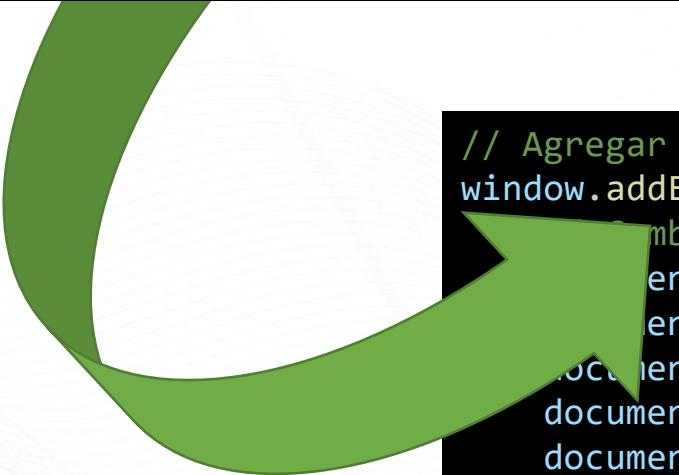
Ejercicio de Aplicación



Eventos en el DOM → carga / load



```
<form action="google.com" method="get" id="formulario">
  <input id="numid" type="text" placeholder="Número de identificación" required>
  <input type="submit" value="Validar" id="validar">
  <p id="mensaje"></p>
</form>
```



```
// Agregar un evento al cargar la página
window.addEventListener("load", function() {
  // Cambiar el estilo del botón
  document.getElementById("validar").style.backgroundColor = "#3498db";
  document.getElementById("validar").style.color = "#fff";
  document.getElementById("validar").style.border = "none";
  document.getElementById("validar").style.padding = "10px 20px";
  document.getElementById("validar").style.borderRadius = "5px";
  document.getElementById("validar").style.cursor = "pointer";
});
```

Eventos en el DOM → carga / load



```
// Agregar un evento al cargar la página
window.addEventListener("load", function() {
    // Cambiar el estilo del botón
    document.getElementById("validar").style.backgroundColor = "#3498db";
    document.getElementById("validar").style.color = "#fff";
    document.getElementById("validar").style.border = "none";
    document.getElementById("validar").style.padding = "10px 20px";
    document.getElementById("validar").style.borderRadius = "5px";
    document.getElementById("validar").style.cursor = "pointer";
    this.alert("La página ha cargado correctamente.");
});
```

Ejercicio de Aplicación





G R A C I A S

Presentó: Alvaro Pérez Niño

Instructor Técnico

Correo: aperezn@sena.edu.co

<http://centrodesserviciosygestionempresarial.blogspot.com/>

Línea de atención al ciudadano: 01 8000 910270

Línea de atención al empresario: 01 8000 910682



www.sena.edu.co