sustantiva

CON•SENTIDO

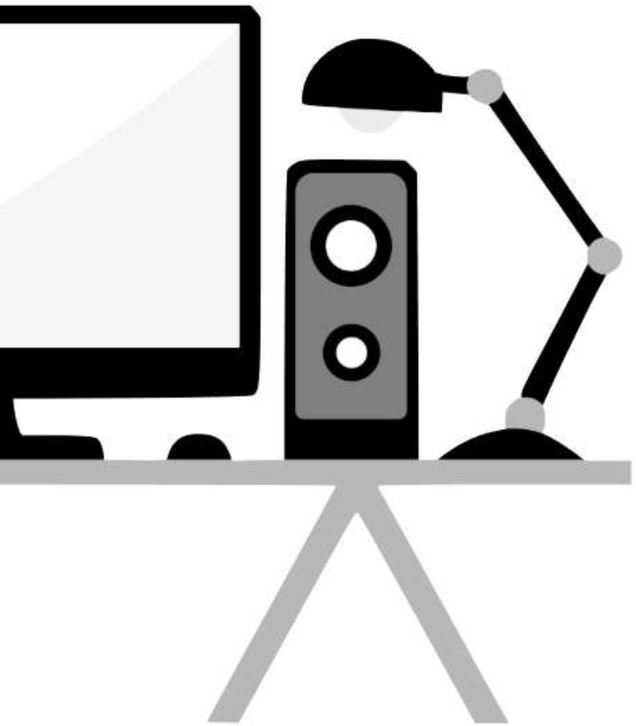
EVENTOS Y MANIPULACIÓN DEL DOM

Lección 03

Competencia General

Reconocer los elementos fundamentales del Domain Object Model y los mecanismos para la manipulación de elementos en un documento HTML.

Contenidos



Document Object Model

1. Qué es el DOM
2. Jerarquía de elementos del DOM

Manipulación DOM

1. Seleccionar elementos
2. Modificar elementos
3. Agregar elementos
4. Modificar estilos

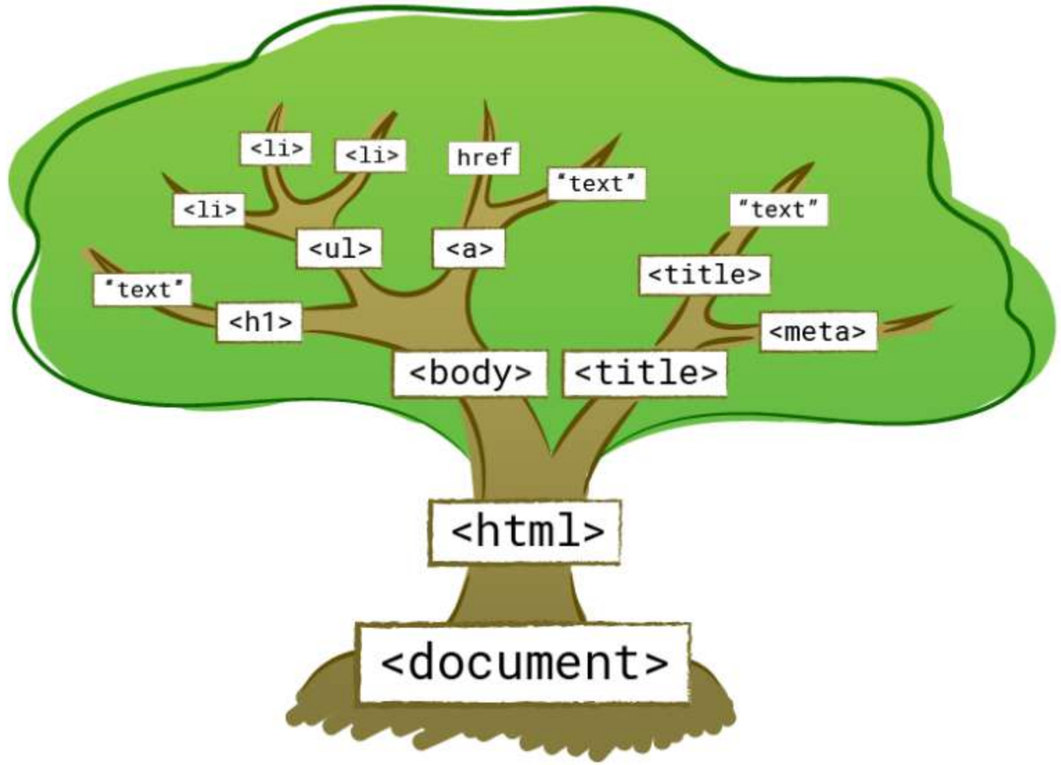
Eventos

1. Qué es un evento
2. Eventos frecuentes

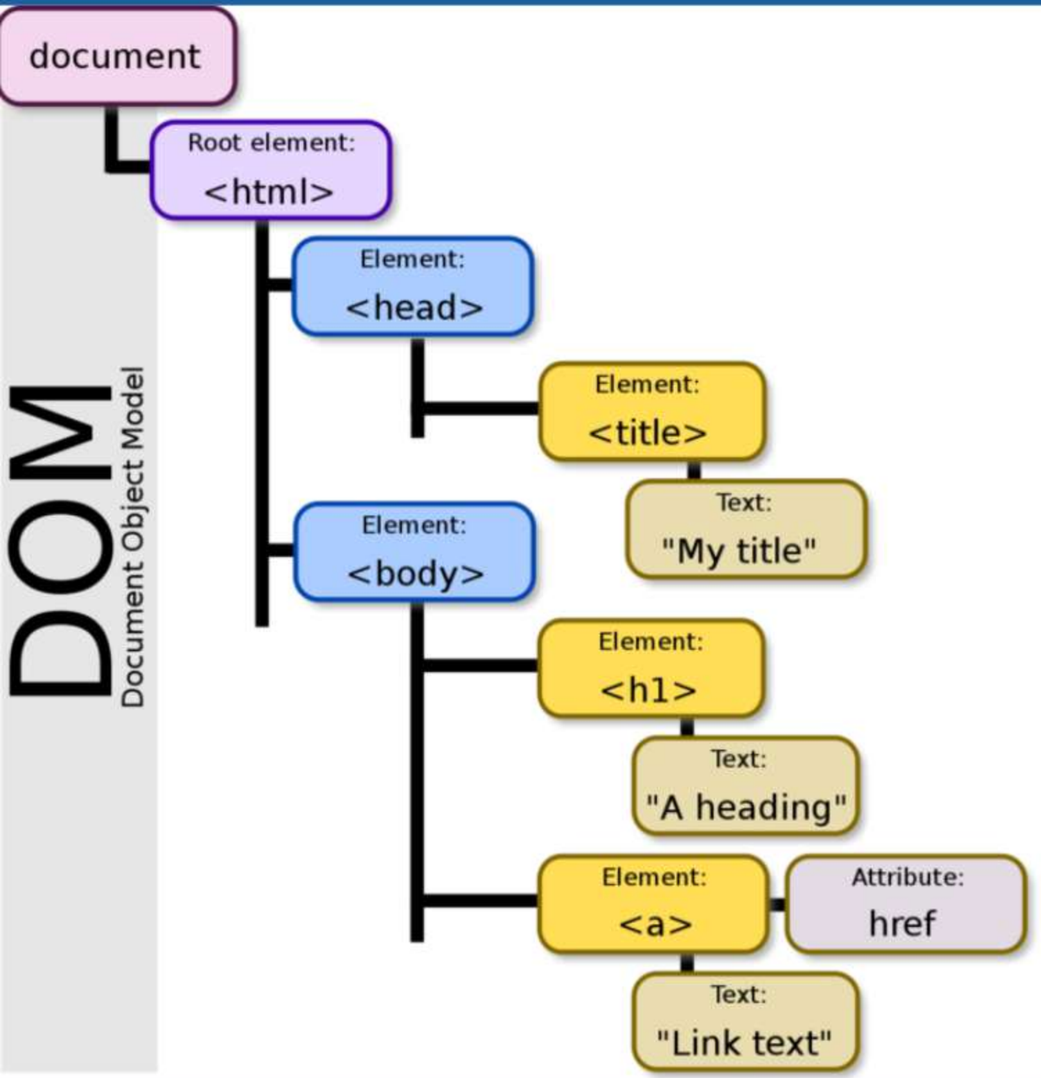
Qué es el DOM

El DOM (Document Object Model) es una jerarquía de objetos del navegador y se encuentra definido y administrado por la W3C.

Cada objeto en la jerarquía del DOM modelan tanto la ventana del navegador como el historial y los elementos de la pagina como párrafos, tablas, formularios, etc.



Jerarquía de elementos del DOM



El objeto superior padre de todos es **window**, de el derivan objetos con funcionalidades distintas entre esos objetos se encuentra **document** que contiene cada elemento del html con sus atributos para desplegarse en pantalla.

Selectores

El uso de selectores en JavaScript permite encontrar y seleccionar elementos del DOM para extraer o manipular información.

Los selectores funcionan de manera similar a los de CSS.

Tenemos dos métodos de seleccionar elementos del DOM en JavaScript.

- El método tradicional mediante las funciones getElement
- El método moderno mediante querySelector

Selectores

El método tradicional

Consta de cuatro funciones y se llama método tradicional porque son funciones que se incluyen desde versiones antiguas de JavaScript. Estas cuatro funciones son:

.getElementById(id)
.getElementsByClassName(class)
.getElementsByName(name)
.getElementsByTagName(tag)

```
const page = document.getElementById("page");
// <div id="page"></div>

const items = document.getElementsByClassName("item")
// [div, div, div]

// Obtiene todos los elementos con  name="nickname"
const nicknames = document.getElementsByName("nickname");

// Obtiene todos los elementos <div> de la página
const divs = document.getElementsByTagName("div");
```

Selectores

El método moderno

Este consta de 2 funciones de búsqueda más cómodo y práctico

.querySelector(sel)
.querySelectorAll(sel)

```
// <div id="page"></div>
const page = document.querySelector("#page");

// <div class="info"></div>
const info = document.querySelector(".main .info");

// Obtiene todos los elementos con clase "info"
const infos = document.querySelectorAll(".info");

// Obtiene todos los elementos con atributo name="nickname"
const nicknames = document.querySelectorAll('[name="nickname"]');

// Obtiene todos los elementos <div> de la página HTML
const divs = document.querySelectorAll("div");
```

Modificar elementos del DOM

Ya seleccionado el elemento, podemos realizar una serie de modificaciones desde JS.

Para modificar los atributos de un elemento tenemos cuatro métodos

Método	Descripción
`hasAttribute( )`	Muestra true si existe el atributo en el elemento
`getAttribute ( )`	Muestra el valor de un atributo
`setAttribute ( )`	Agrega o actualiza un atributo en el elemento
`removeAttribute ( )`	Elimina el atributo de un elemento

Modificar elementos del DOM

También JavaScript incluye métodos para modificar clases en un atributo

Método	Descripción
Element.className	Muestra o establece un valor de clase
classList.add( )	Agrega una o mas clases al elemento
classList.toggle ( )	Activa o desactiva una clase
classList.contains ( )	Comprueba si existe una clase en un elemento
classList.replace ( )	Sustituye una clase por otra
classList.remove ( )	Elimina una clase de un elemento

Agregar elementos al DOM

Existen una serie de métodos que permiten crear de forma eficiente elementos HTML y agregar estos al DOM

Lo primero es crear el elemento a insertar usando el método

.createElement(tag, options)

Este método crea un elemento HTML de acuerdo con el tag

```
const div = document.createElement("div");
 // Creamos un <div></div>
const span = document.createElement("span");
 // Creamos un <span></span>
const img = document.createElement("img");
 // Creamos un <img>
```

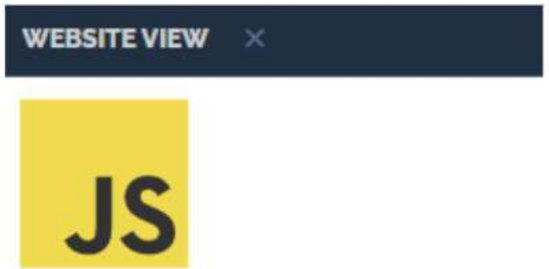
Agregar elementos al DOM

Ya creado el elemento debemos insertarlo al DOM, para ello, debemos referenciar a su elemento padre y usar el método

.appendChild(node)

Este método añade como hijo al elemento ***node***

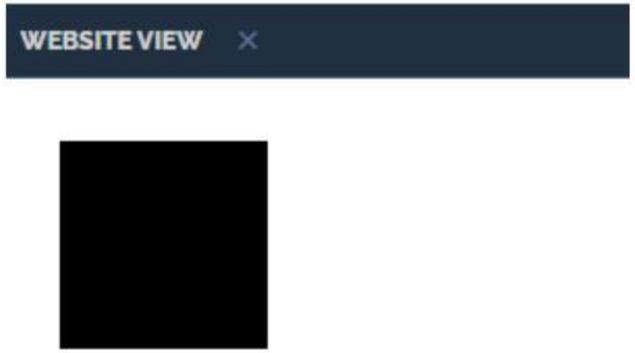
```
const img = document.createElement("img");
img.src = "https://lenguajejs.com/assets/logo.svg";
img.alt = "Logo Javascript";
img.width = 100

document.body.appendChild(img);
```



Modificar estilos

Para modificar estilos CSS seleccionamos el elemento a modificar y usando la propiedad ***style*** podemos modificar los estilos desde JS.

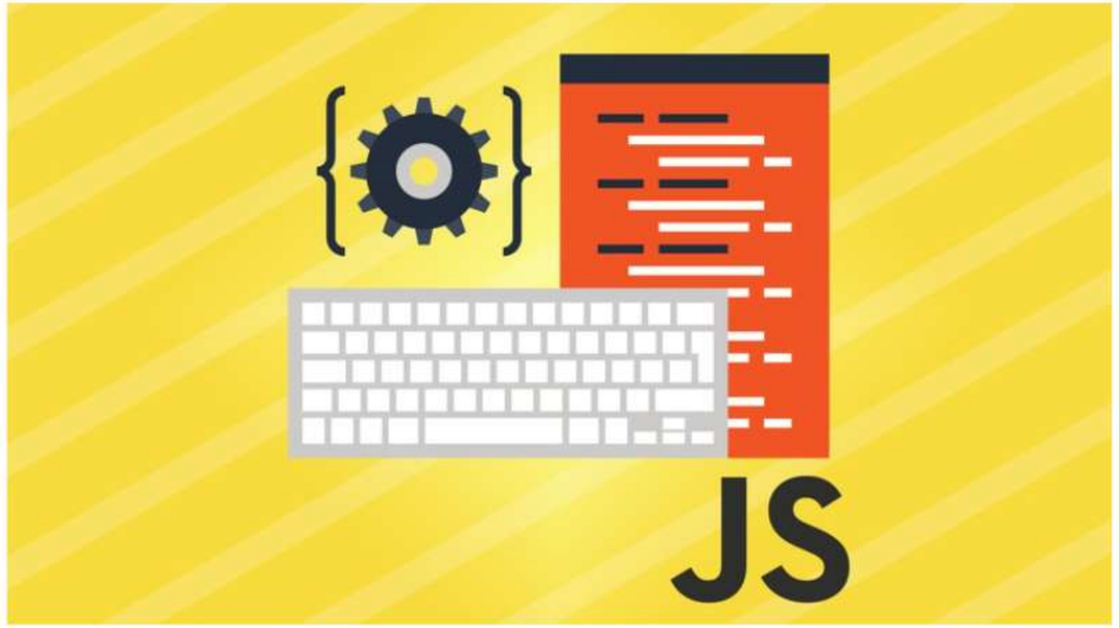
```
const img = document.createElement("div");
img.style.marginTop = "30px"
img.style.marginLeft = "30px"
img.style.height = "100px"
img.style.width = "100px"
img.style.backgroundColor = "#000000";

document.body.appendChild(img);
```



Qué es un evento

En JavaScript, la interacción con el usuario se consigue mediante la captura de eventos. Un evento es una acción del usuario ante la cual puede realizarse algún proceso.

Por ejemplo validar un campo de un formulario.



Eventos frecuentes

Evento onClick

El evento onClick permite ejecutar una función cuando se le da clic a algún elemento.

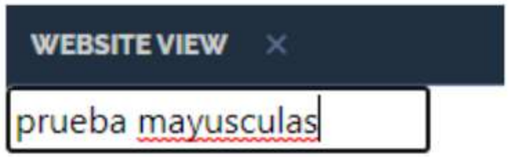
```
let p = document.getElementById("foo");
// Encuentra el elemento "p" en el sitio
  p.onclick = muestraAlerta;
// Agrega función onclick al elemento

  function muestraAlerta(evento) {
    alert("Evento onclick ejecutado!");
  }
```

Eventos frecuentes

Evento onBlur

El evento onBlur permite ejecutar una acción al dejar de hacer focus en un elemento

```
let input = document.getElementById("fname");

input.addEventListener("blur",() => {
  input.value = input.value.toUpperCase();
})
```



Eventos frecuentes

Evento onKeyDown

El evento onKeyDown permite ejecutar una acción al presionar una tecla

```
let input = document.getElementById("fname");

input.addEventListener("keydown",(e) => {
  alert('tecla presionada ' + e.key)
})
```

Eventos frecuentes

Otros eventos importantes de JS:

Evento	Descripción
onFocus	Un elemento de algún formulario recibe foco
onChange	Un valor de un formulario cambia
onDblClick	Se hace doble clic en algún elemento
onAbort	Se interrumpe alguna accion
onSubmit	Se envia un formulario
onLoad	Se cargan todos los elementos del DOM
onReset	Se limpia un formulario

Ejercicio Práctico

Diseña una aplicación web que mediante un formulario, se introduzca una temperatura en °C y al presionar un botón la página muestre la temperatura en °F.

sustantiva

CON•SENTIDO