

Guía práctica: JavaScript, DOM y Eventos

1. ¿Qué es el DOM y para qué sirve?

El DOM (Document Object Model) representa la estructura del documento HTML como un árbol de nodos. Es la forma que tiene JavaScript de "ver" y manipular la página web.

Con DOM podemos: cambiar textos, imágenes, clases CSS, añadir o quitar nodos, manejar eventos del usuario y actualizar dinámicamente el contenido del sitio.

2. Selección de elementos

Métodos básicos:

```
- document.getElementById('id')
- document.getElementsByClassName('clase')
- document.getElementsByTagName('etiqueta')
- document.querySelector('selector CSS')
- document.querySelectorAll('selector CSS')
```

querySelector/querySelectorAll permiten usar selectores CSS como '#miId', '.miClase', 'div p'. Son los más versátiles.

Ejemplo práctico de selección

Seleccionar y cambiar texto de elementos:

```
var titulo = document.getElementById('titulo');
titulo.innerHTML = "Nuevo título";

var parrafos = document.querySelectorAll('.parrafo');
parrafos.forEach(p => p.textContent += " (actualizado)");
```

3. Manipulación del DOM

Con los elementos seleccionados, podemos:

- Cambiar texto (innerHTML, textContent)
- Cambiar estilos (element.style.prop)
- Añadir clases (element.classList.add)
- Crear y borrar nodos (createElement, remove)
- Cambiar atributos (setAttribute, removeAttribute)

Ejemplo de manipulación:

```
var caja = document.createElement("div");
caja.textContent = "Soy nueva";
caja.style.backgroundColor = "lightblue";
document.body.appendChild(caja);
```

4. Eventos en JavaScript

Los eventos permiten reaccionar a acciones del usuario: click, keydown, mouseover, submit, etc.

Método recomendado: addEventListener

Ejemplo básico:

```
document.getElementById('btn').addEventListener('click', () => {
    alert('¡Haz hecho clic!');
});
```

Guía práctica: JavaScript, DOM y Eventos

```
});
```

Eventos comunes:

- click, dblclick, mouseover, mouseout
- keydown, keyup, keypress
- change, input, submit, focus, blur
- load, scroll, resize, contextmenu

5. Ejercicio práctico: cambiar color de fondo aleatorio

Creamos un botón que al hacer clic cambie el fondo del body:

```
document.getElementById("cambiarFondo").addEventListener("click", () => {
  const r = Math.floor(Math.random() * 256);
  const g = Math.floor(Math.random() * 256);
  const b = Math.floor(Math.random() * 256);
  document.body.style.backgroundColor = `rgb(${r}, ${g}, ${b})`;
});
```

6. Ejercicio: ToDo List básico con JS puro

Creamos una lista de tareas donde podemos agregar tareas nuevas con un input:

```
const btn = document.getElementById("agregar");
btn.addEventListener("click", () => {
  const tarea = document.getElementById("tarea").value;
  if (tarea.trim() !== "") {
    const li = document.createElement("li");
    li.textContent = tarea;
    document.getElementById("lista").appendChild(li);
    document.getElementById("tarea").value = "";
  }
});
```

7. Buenas prácticas

- Usar `addEventListener`, evitar eventos inline
- Separar lógica JS del HTML
- Usar funciones puras y reutilizables
- Aprovechar funciones flecha para claridad
- Optimizar accesos al DOM (guardar referencias)
- Evitar `document.write()`
- Usar `classList` para manipular clases CSS
- Nombrar IDs y clases de forma semántica

8. Recursos para profundizar

- MDN Web Docs: <https://developer.mozilla.org/es/>
- FreeCodeCamp.org (Frontend)
- Curso JavaScript de Eniun.com
- JavaScript.info
- YouTube: Fazt, HolaMundo, Midudev
- Proyecto 100 días de JS (100 proyectos prácticos)

Guía práctica: JavaScript, DOM y Eventos

¡Sigue practicando y probando código en CodePen, JSFiddle o tu editor local!