

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**COMMUNITY MANAGER Y
MARKETING DIGITAL**

Tema

**EVENTOS EN JAVASCRIPT PARA
MANIPULAR EL DOM DE FORMA
DINÁMICA**

Eventos en JavaScript para Manipular el DOM de Forma Dinámica

1. ¿Qué es el DOM y cómo se representa en JavaScript?

El **DOM (Document Object Model)** es una representación en forma de árbol de los elementos HTML de una página web. JavaScript puede acceder y manipular estos elementos para modificar su contenido, estructura o apariencia de manera dinámica.

💡 Ejemplo de estructura del DOM:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de DOM</title>
</head>
<body>
  <h1 id="titulo">Hola, Mundo</h1>
  <button id="boton">Haz clic</button>
</body>
</html>
```

En este ejemplo:

- <html> es el nodo raíz.
- <head> y <body> son nodos hijos.
- <h1> y <button> son elementos dentro de <body>.
- Cada elemento tiene atributos como id, los cuales JavaScript puede utilizar para seleccionarlos y manipularlos.

2. Eventos en JavaScript

Un **evento** es cualquier acción que ocurre en una página web y que JavaScript puede detectar, como un clic del usuario, el envío de un formulario o el cambio en un campo de entrada.

2.1 Evento click (cuando el usuario hace clic en un elemento)

Este evento se dispara cuando el usuario hace clic en un botón, enlace u otro elemento.

❖ Ejemplo:

```
document.getElementById("boton").addEventListener("click", function()
{
    alert("¡Botón
});                                              clickeado!");
```

2.2 Evento submit (cuando se envía un formulario)

Se activa cuando un formulario es enviado. Es útil para validar datos antes del envío.

❖ Ejemplo:

```
document.getElementById("miFormulario").addEventListener("submit",
function(event)
{
    event.preventDefault(); // Evita el envío automático
    alert("Formulario
);                                              enviado
                                              correctamente.");
```

2.3 Evento change (cuando cambia el valor de un input)

Se activa cuando el usuario cambia el valor de un campo de entrada (input, select, textarea).

 **Ejemplo:**

```
document.getElementById("miInput").addEventListener("change",  
function()  
    console.log("Nuevo valor:", this.value);  
});
```

2.4 Evento mouseover (cuando el cursor pasa sobre un elemento)

Se dispara cuando el cursor del mouse pasa sobre un elemento.

 **Ejemplo:**

```
document.getElementById("caja").addEventListener("mouseover",  
function()  
    this.style.backgroundColor = "yellow";  
});
```

3. Métodos para manejar eventos

Existen diferentes maneras de asociar eventos en JavaScript.

3.1 onclick en el HTML (no recomendado)

Definir eventos directamente en HTML es una mala práctica, ya que hace que el código sea menos organizado y más difícil de mantener.

 **Ejemplo (no recomendado):**

```
<button onclick="alert('Haz hecho clic')>Clic aquí</button>
```

3.2 addEventListener() (forma recomendada)

Este método permite agregar múltiples eventos sin modificar el HTML y es más flexible.

❖ Ejemplo recomendado:

```
document.getElementById("boton").addEventListener("click", function()
{
    alert("Evento manejado con addEventListener");
});
```

4. Demostración práctica

4.1 Creación de un formulario con un botón de envío

❖ HTML:

```
<form id="miFormulario">
    <label for="nombre">Nombre:</label>
    <input type="text" id="nombre">
    <button type="submit">Enviar</button>
</form>
<p id="mensaje"></p>
```

❖ JavaScript:

```
document.getElementById("miFormulario").addEventListener("submit",
function(event)
{
    event.preventDefault();
    let nombre = document.getElementById("nombre").value;
    document.getElementById("mensaje").textContent = "Hola, " + nombre;
});
```

4.2 Uso de addEventListener() para capturar eventos y modificar el DOM

Cambiar el texto de un elemento con `textContent`

❖ Ejemplo:

```
document.getElementById("boton").addEventListener("click", function()
{
    document.getElementById("titulo").textContent = "Texto cambiado con
JavaScript";
});
```

Modificar estilos de un elemento con `style`

❖ Ejemplo:

```
document.getElementById("boton").addEventListener("click", function()
{
    document.getElementById("titulo").style.color = "red";
});
```

Agregar y eliminar clases con `classList.add()` y `classList.remove()`

❖ CSS:

```
.destacado
{
    font-size: 20px;
    color: blue;
    background-color: yellow;
}
```

❖ JavaScript:

```
document.getElementById("boton").addEventListener("click", function()
{
    let titulo = document.getElementById("titulo");
    titulo.classList.toggle("destacado"); // Agrega o quita la clase
});
```

Conclusión

- El DOM es la representación estructurada de una página web y se puede modificar con JavaScript.
- Los eventos permiten capturar acciones del usuario como clics, cambios en formularios y movimientos del mouse.
- addEventListener() es la mejor forma de manejar eventos.
- Se pueden modificar elementos del DOM dinámicamente usando textContent, style, y classList.

