



IT Academy
by KIBERNUM

Programa: Desarrollo de Aplicaciones Front-End

Módulo 2: Fundamentos de Desarrollo Front-End



Aprendizaje esperado

Utilizar código Javascript para la personalización de eventos sencillos dentro de un documento html dando solución al problema planteado.



Eventos Básicos: onClick

- ◆ **Sintaxis:** El evento onClick se activa cuando el usuario hace clic en un elemento. Se puede asociar a cualquier elemento interactivo, como botones, enlaces o divs.

```
<button onclick="miFuncion()">Haz clic aquí</button>
```

- ◆ **Ejemplo:** En este ejemplo, al hacer clic en el botón, se ejecuta la función miFuncion().

```
var boton = document.getElementById("miBoton");  
console.log(boton); // Muestra el elemento botón en la consola
```

Ejemplo: onClick

- ◆ Botón con onClick
- ◆ HTML

```
<body>  
  <button id="miBoton">Haz clic en mí</button>  
</body>
```

- ◆ JavaScript

```
var boton = document.getElementById("miBoton");  
boton.onclick = function() {  
  alert("¡Has hecho clic!");  
};
```


Eventos Básicos: onChange

- ◆ **Sintaxis:** El evento onChange se activa cuando el valor de un campo de formulario cambia. Es comúnmente utilizado en elementos como <input>, <select>, o <textarea>.

```
<input type="text" id="miInput" value="Texto inicial">
```

- ◆ **Ejemplo:**

```
var input = document.getElementById("miInput");
input.onChange = function() {
  alert("El valor ha cambiado a: " + input.value);
};
```

Ejemplo: onChange

- ◆ Detectar cambios en un campo de entrada
- ◆ HTML

```
<body>  
  <input type="text" id="miInput" value="Texto inicial">  
</body>
```

- ◆ JavaScript

```
var input = document.getElementById("miInput");  
input.onChange = function() {  
  alert("El valor ha cambiado a: " + input.value);  
};
```


TIPS: Evento onMouseOver / onMouseOut

- **Objetivo:** detectar cuando el usuario pasa el mouse sobre un elemento o lo retira.
- **Uso típico:** destacar botones, mostrar tooltips o cambiar estilos.
- **Ejemplo:**

```
<button onMouseOver="this.style.background='lightblue'" onMouseOut="this.style.background=''">
    Pasa el mouse
</button>
```

Consejo: Ideal para mejorar la experiencia visual sin usar CSS avanzado.

TIPS: Evento onSubmit

- **Objetivo:** ejecutar código al enviar un formulario.
- **Uso típico:** validar datos antes de enviarlos.
- **Ejemplo:**

```
<form onSubmit="return validarFormulario()">
  <input type="text" placeholder="Nombre">
  <button type="submit">Enviar</button>
</form>
<script>
function validarFormulario() {
  alert("Formulario enviado correctamente!");
  return false; // Evita el envío real
}
</script>
```

Consejo: útil para prevenir errores antes de que lleguen al servidor.

TIPS: Evento onKeyUp/ onKeyDown

- **Objetivo:** detectar cuando el usuario presiona o suelta una tecla.
- **Uso típico:** validación en tiempo real o búsqueda instantánea.
- **Ejemplo:**

```
<input type="text" onKeyUp="console.log(this.value)" placeholder="Escribe algo...">
```

Consejo: perfecto para crear experiencias más dinámicas sin recargar la página.



Actividad Práctica Guiada

Formulario interactivo con eventos

Objetivo:

Que el estudiante aprenda a usar eventos básicos de JavaScript (onClick, onChange, onInput) para interactuar con elementos HTML y modificar el DOM dinámicamente.

Requerimientos:

- Editor de código (VS Code o similar)
- Navegador web (Chrome, Edge, etc.)
- Conocimientos básicos de HTML y JavaScript



Actividad Práctica Guiada

- Crea un archivo `index.html`
- Escribe la estructura base del documento:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Eventos en acción</title>
</head>
<body>
  <h2>Mi primer formulario interactivo</h2>
</body>
</html>
```

Guarda el archivo y ábrelo en tu navegador para comprobar que funciona.



Actividad Práctica Guiada

- Dentro del `<body>`, agrega un campo de texto y un botón:

```
<input type="text" id="nombre" placeholder="Escribe tu nombre" onChange="mostrarMensaje()">  
<button onClick="saludar()">Saludar</button>  
<p id="mensaje"></p>
```

- `onChange` se activa cuando el valor del input cambia.
- `onClick` se activa al presionar el botón.



Actividad Práctica Guiada

- Agrega un bloque `<script>` al final del `<body>`:

```
<script>
  function mostrarMensaje() {
    const nombre = document.getElementById("nombre").value;
    document.getElementById("mensaje").innerText = "Has escrito: " + nombre;
  }

  function saludar() {
    const nombre = document.getElementById("nombre").value;
    alert("¡Hola, " + nombre + "!");
  }
</script>
```




Actividad Práctica Guiada

1- Agrega un nuevo párrafo dinámico:

```
<input type="text" id="edad" placeholder="Escribe tu edad" onInput="mostrarEdad()">
<p id="edadMensaje"></p>
```

2.- Agrega la función correspondiente:

```
function mostrarEdad() {
  const edad = document.getElementById("edad").value;
  document.getElementById("edadMensaje").innerText = "Tienes " + edad + " años.";
}
```

- onInput responde en tiempo real mientras el usuario escribe.
- Ideal para validaciones instantáneas.



Mini Reto

- Agrega un nuevo botón que borre todo el contenido del formulario usando el evento onClick.



Resumen

- ✓ Historia y evolución de JavaScript
- ✓ Importancia y aplicaciones en la web moderna
- ✓ Diferentes formas de incluir JavaScript en HTML
- ✓ Selectores del DOM y manipulación de elementos
- ✓ Eventos y cómo capturar interacciones del usuario
- ✓ Variables, expresiones aritméticas y estructuras condicionales
- ✓ Creación y uso de funciones
- ✓ Uso de la consola del navegador y herramientas de depuración

Si tuvieras que resolver un problema usando JavaScript, ¿por dónde empezarías?

¿Cómo podrías enseñar a otros lo que aprendiste hoy?



Conceptos Fundamentales

Variables

- ✓ Usa `const` para valores que no cambian.
 - ✓ Usa `let` para valores que pueden cambiar.
 - ✓ Evita `var` porque puede causar errores inesperados.
-

Depurar

- ✓ `console.log()` te ayuda a ver valores en la consola del navegador.
 - ✓ Úsalo para revisar si una variable tiene el valor correcto.
-

Comentario en el código

- ✓ Usa `//` para comentarios de una línea y `/* */` para varios renglones.

Conceptos Fundamentales

Nombres de variables

- ✓ Evita nombres genéricos como x o dato.
- ✓ Usa nombres claros como `precioProducto` o `nombreUsuario`.

Funciones

- ✓ Si haces algo varias veces, usa una función.



IT Academy

by KIBERNUM