

Alexander Ocoro



Mentor



Warner Fdo. Valencia



Ejecutor

JavaScript Básico

Objetivo: Introducir los conceptos fundamentales de JavaScript y cómo usarlo para agregar interactividad básica a una página web.

Fundamentos de JavaScript

Objetivo: Comprender los conceptos básicos de JavaScript: variables, tipos de datos y operadores.

Fundamentos de JavaScript

Teoría: Fundamentos de JavaScript

1. ¿Qué es JavaScript?

- JavaScript es un lenguaje de programación de alto nivel que se utiliza para añadir interactividad a las páginas web.
- Es un lenguaje de scripting interpretado en el navegador, lo que permite ejecutar código directamente en el cliente.

2. Variables

- Las variables se utilizan para almacenar datos y pueden ser modificadas durante la ejecución del programa.
- `var`, `let` y `const` son las palabras clave utilizadas para declarar variables:
 - `var`: Tiene un alcance global o de función, y es la forma más antigua de declarar variables.
 - `let`: Permite declarar variables con un alcance de bloque.
 - `const`: Declara constantes cuyo valor no puede cambiar.

3. Tipos de Datos

- String**: Cadena de texto, delimitada por comillas simples (`'texto'`) o dobles (`"texto"`).
- Number**: Números enteros o decimales (por ejemplo, `10`, `3.14`).
- Boolean**: Representa valores lógicos, `true` o `false`.

4. Operadores

- Aritméticos**: `+`, `-`, `*`, `/` para realizar operaciones matemáticas.
- De comparación**: `==`, `!=`, `>`, `<` para comparar valores.
- De asignación**: `=`, `+=`, `-=`, etc., para asignar valores a variables.

Ejemplo Práctico: Crear y Manipular Variables en la Consola

1. **Paso 1:** Abre la consola en el navegador (puedes hacerlo con **F12** o haciendo clic derecho y seleccionando "Inspeccionar" > "Consola").
2. **Paso 2:** Escribe los siguientes ejemplos en la consola:

```
let nombre = "Juan";  
const edad = 25;  
let esEstudiante = true;  
  
console.log("Nombre:", nombre);  
console.log("Edad:", edad);  
console.log("¿Es estudiante?",  
esEstudiante);
```

3. **Paso 3:** Prueba con operadores básicos.

```
let numero1 = 10;  
let numero2 = 5;  
console.log("Suma:", numero1 + numero2);  
console.log("Resta:", numero1 - numero2);
```

Estructuras de Control

Objetivo: Comprender y aplicar estructuras de control condicionales y bucles para realizar tareas repetitivas y condicionales.

Teoría: Estructuras de Control

1. Condicionales (if, else)

- La estructura `if` permite ejecutar código solo si una condición es verdadera. Con `else`, se puede ejecutar código alternativo si la condición es falsa.
- Ejemplo:

```
let edad = 20;  
if (edad >= 18) {  
  console.log("Es mayor de edad");  
} else {  
  console.log("Es menor de edad");  
}
```

2. Bucles

- **For:** Ejecuta un bloque de código un número específico de veces.

```
for (let i = 0; i < 5; i++) {  
  console.log("Número:", i);  
}
```

- **While:** Ejecuta el código mientras se cumpla una condición.

```
let i = 0;  
while (i < 5) {  
  console.log("Número:", i);  
  i++;  
}
```


Ejemplo Práctico: Recorrer una Lista de Elementos

1. **Paso 1:** Define una lista de nombres y recórrela usando un bucle `for`.

```
let nombres = ["Ana", "Luis", "Pedro", "María"];  
for (let i = 0; i < nombres.length; i++) {  
  console.log("Nombre:", nombres[i]);  
}
```

Actividad Práctica

Recorrer y Condicionar: Los estudiantes deben crear su propio array de elementos y recorrerlo, mostrando solo los elementos que cumplen una condición específica (por ejemplo, nombres con más de cuatro letras).

Funciones Básicas

Objetivo: Aprender a crear y usar funciones en JavaScript para organizar código reutilizable.

Teoría: Funciones en JavaScript

1. ¿Qué es una Función?

- Una función es un bloque de código reutilizable que realiza una tarea específica.
- Se declara con la palabra clave `function`, seguida de un nombre, paréntesis y un bloque de código.

2. Parámetros y Retorno

- Los parámetros permiten pasar valores a una función.
- La palabra clave `return` se usa para devolver un valor desde la función.
- Ejemplo:

```
function sumar(a, b) {  
  return a + b;  
}  
let resultado = sumar(3, 4);  
console.log("Resultado de la suma:",  
resultado);
```

Ejemplo Práctico: Crear una Función que Muestre un Mensaje de Bienvenida

1. **Paso 1:** Define una función para mostrar un mensaje de bienvenida.

```
function mostrarBienvenida(nombre) {  
  return "¡Bienvenido, " + nombre + "!";  
}
```

2. **Paso 2:** Llama a la función con un nombre de ejemplo y muestra el resultado en la consola.

```
let mensaje = mostrarBienvenida("Carlos");  
console.log(mensaje);
```

Actividad Práctica

Crear una Función Personalizada: Los estudiantes deben crear una función que tome un parámetro y devuelva un mensaje personalizado.

Manipulación del DOM

Objetivo: Aprender a seleccionar y manipular elementos de la página web usando JavaScript.

Teoría: Manipulación del DOM (Document Object Model)

1. ¿Qué es el DOM?

- El DOM es la estructura de un documento HTML que JavaScript puede manipular.
- Permite cambiar el contenido, el estilo y la estructura de la página en respuesta a eventos.

2. Métodos de Selección

- `document.getElementById`: Selecciona un elemento por su ID.
- `document.querySelector`: Selecciona el primer elemento que coincida con un selector CSS.
- Ejemplo:

```
<p id="saludo">¡Hola!</p>
```

```
let saludo =  
document.getElementById("saludo");  
saludo.textContent = "¡Bienvenido!";
```


Ejemplo Práctico: Cambiar el Contenido de un Elemento

1. **Paso 1:** En `index.html`, agrega un elemento con un ID.

```
<p id="mensaje">Este es un mensaje inicial.</p>
```

2. **Paso 2:** En un archivo de JavaScript vinculado (por ejemplo, `script.js`), selecciona el elemento y cambia su contenido.

```
let mensaje = document.getElementById("mensaje");  
mensaje.textContent = "Este es el nuevo mensaje  
después de usar JavaScript";
```

Actividad Práctica

Cambio de Contenido Dinámico: Los estudiantes practicarán seleccionando y modificando el contenido de varios elementos en la página.

Proyecto de Interactividad

Objetivo: Usar lo aprendido para añadir un botón que, al hacer clic, cambie el estilo de un elemento en la página.

Práctica Guiada: Crear Interactividad con Eventos

1. **Paso a Paso:** Añadir un botón en el HTML que desencadene un cambio de estilo al hacer clic.

1. **Paso 1:** En `index.html`, añade un botón y un elemento para cambiar su estilo.

```
<button id="cambiarEstilo">Cambiar
Estilo</button>
<div
  id="caja"
  style="width: 100px; height: 100px;
background-color: lightblue;"
></div>
```

2. **Paso 2:** En `script.js`, selecciona el botón y la caja, y añade un evento `click` para cambiar el estilo de la caja.

```
let boton =
document.getElementById("cambiarEstilo");
let caja = document.getElementById("caja");

boton.addEventListener("click", function () {
  caja.style.backgroundColor =
    caja.style.backgroundColor === "lightblue" ?
    "salmon" : "lightblue";
});
```

Actividad Práctica

Cambio de Contenido Dinámico: Los estudiantes practicarán seleccionando y modificando el contenido de varios elementos en la página.