

GUÍA TEÓRICO-PRÁCTICA:

UNIDAD I

SEMANA 1: FUNDAMENTOS DE JAVASCRIPT

TEMA 1.1: INTRODUCCIÓN A JAVASCRIPT - VARIABLES, TIPOS DE DATOS Y OPERADORES

Teoría

¿Qué es JavaScript? JavaScript es un lenguaje de programación interpretado, dinámico y orientado a objetos que permite agregar interactividad a las páginas web. Fue creado por Brendan Eich en 1995.

Variables: Una variable es un contenedor para almacenar datos. En JavaScript existen tres formas de declarar variables:

```
1 var nombre = "Juan";      // Declaración antigua (ámbito de función)
2 let edad = 25;           // Declaración moderna (ámbito de bloque)
3 const PI = 3.1416;       // Constante (no se puede reasignar)
```

Tipos de Datos:

- **String (Cadena):** "Hola Mundo", 'JavaScript'
- **Number (Número):** 25, 3.14, -10
- **Boolean (Booleano):** true, false
- **Undefined:** Variable declarada sin valor
- **Null:** Ausencia intencional de valor
- **Object:** Colección de propiedades
- **Array:** Lista ordenada de valores

Pregunta de reflexión 1: ¿Cuál es la diferencia entre let y const? ¿Cuándo usarías cada una?

Operadores:

- **Aritméticos:** + (suma), - (resta), * (multiplicación), / (división), % (módulo), ** (potencia)

- **Asignación:** =, +=, -=, *=, /=
- **Comparación:** == (igualdad), === (igualdad estricta), !=, !==, >, <, >=, <=
- **Lógicos:** && (AND), || (OR), ! (NOT)

Ejemplo

```
javascript
```

```

1 // Declaración de variables
2 let nombre = "María";
3 let edad = 28;
4 const altura = 1.65;
5
6 // Operaciones aritméticas
7 let suma = 10 + 5; // 15
8 let resta = 10 - 5; // 5
9 let multiplicacion = 10 * 5; // 50
10 let division = 10 / 5; // 2
11 let modulo = 10 % 3; // 1 (residuo)
12
13 // Comparaciones
14 let esMayor = edad > 18; // true
15 let esIgual = 5 == "5"; // true (comparación débil)
16 let esEstricto = 5 === "5"; // false (comparación estricta)

```

Ejercicio 1.1

Crea un programa que:

1. Declare variables para: tu nombre, tu edad, tu altura
2. Realice las siguientes operaciones:
 - Suma tu edad con 10
 - Multiplica tu altura por 100
 - Calcula el módulo de tu edad entre 5
3. Muestra los resultados usando console.log()

TEMA 1.2: SALIDA BÁSICA - `console.log()` Y `alert()`

Teoría

JavaScript necesita formas de mostrar información. Las dos más básicas son:

`console.log()`: Muestra mensajes en la consola del navegador (herramientas de desarrollador). Es ideal para depuración y no interrumpe al usuario .

```
javascript
```

```
1 console.log("Mensaje en consola");
2 console.log("Valor:", variable);
3 console.log("Suma:", 5 + 3);
```

`alert()`: Muestra una ventana emergente con un mensaje. Detiene la ejecución hasta que el usuario hace clic en "Aceptar"

```
javascript
```

```
1 alert("Hola, bienvenido");
2 alert("Tu resultado es: " + resultado);
```

Otras salidas útiles:

- **`console.warn()`**: Muestra advertencias en amarillo
- **`console.error()`**: Muestra errores en rojo
- **`confirm()`**: Muestra cuadro de diálogo con "Aceptar" y "Cancelar"
- **`prompt()`**: Solicita entrada de texto al usuario

Pregunta de reflexión 2: ¿Cuándo es más apropiado usar `console.log()` en lugar de `alert()`?

Ejemplo

```
javascript
```

```

1 // Usando console.log
2 let nombre = "Carlos";
3 let edad = 30;
4
5 console.log("Nombre:", nombre);
6 console.log("Edad:", edad);
7 console.log("Tipo de dato de nombre:", typeof nombre);
8 console.log("Tipo de dato de edad:", typeof edad);
9
10 // Usando alert
11 alert("Bienvenido " + nombre);
12 alert("Tienes " + edad + " años");
13
14 // Combinando con prompt
15 let ciudad = prompt("¿En qué ciudad vives?");
16 console.log("El usuario vive en:", ciudad);
17 alert("¡Qué bonito es " + ciudad + "!");

```

Ejercicio 1.2

Crea un programa que:

1. Use `prompt()` para preguntar el nombre del usuario
2. Use `prompt()` para preguntar la edad
3. Use `console.log()` para mostrar ambos datos con mensajes descriptivos
4. Use `alert()` para dar la bienvenida al usuario con su nombre

```
javascript
```

```

1 // Tu código aquí:
2
3
4
5

```

TEMA 1.3: CONECTAR JS CON HTML (<script>)

Teoría

Existen tres formas de incluir JavaScript en HTML:

1. JavaScript Interno (Internal): El código se escribe dentro de etiquetas <script> en el HTML .

```
html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 |   <title>Mi Página</title>
5 </head>
6 <body>
7 |   <h1>Hola Mundo</h1>
8
9   <script>
10 |     console.log("JavaScript interno");
11 |     alert("Página cargada");
12 |   </script>
13 </body>
14 </html>
```

2. JavaScript Externo (External): El código se escribe en un archivo .js separado y se enlaza con src .

```
html
1 <!-- HTML -->
2 <script src="miarchivo.js"></script>
```

```
javascript
1 // miarchivo.js
2 console.log("JavaScript externo");
```

3. JavaScript en Línea (Inline): El código se escribe directamente en los atributos HTML (no recomendado).

```
html
1 <button onclick="alert('Click!')">Haz click</button>
```

Ubicación del script:

- **En <head>:** Se ejecuta antes de cargar el contenido (puede causar errores si busca elementos que aún no existen)
- **Al final de <body>:** Recomendado, asegura que el DOM esté cargado
- **Con atributo defer:** `<script src="app.js" defer></script>` (se ejecuta después de cargar el HTML)
- **Con atributo async:** `<script src="app.js" async></script>` (se ejecuta apenas se descarga)

Pregunta de reflexión 3: ¿Por qué es mejor colocar el script al final del body o usar defer?

Ejemplo

html

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Ejemplo Script</title>
6   <!-- Script externo con defer -->
7   <script src="main.js" defer></script>
8 </head>
9 <body>
10  <h1 id="titulo">Bienvenido</h1>
11
12  <!-- Script interno -->
13  <script>
14    console.log("El DOM está cargado");
15    document.getElementById("titulo").style.color = "blue";
16  </script>
17 </body>
18 </html>

```

javascript

```

1 // main.js
2 console.log("Archivo externo cargado");
3 let mensaje = "Hola desde archivo externo";
4 console.log(mensaje);

```

Ejercicio 1.3

Crea un archivo HTML que:

1. Tenga un título <h1> con tu nombre
2. Incluya un script interno que cambie el color del título a rojo
3. Incluya un script externo (archivo separado) que muestre un alert() de bienvenida
4. Ambos scripts deben usar console.log() para mostrar mensajes

archivo.html:

```
html

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Mi Ejercicio</title>
5
6 </head>
7 <body>
8   <h1 id="miNombre"></h1>
9
10
11 </body>
12 </html>
```

archivo.js:

```
javascript

1 // Tu código aquí:
2
```

TEMA 1.4: CREACIÓN DE UNA CALCULADORA SIMPLE

Teoría

Una calculadora básica requiere:

1. **Entrada de datos:** Obtener números del usuario
2. **Operaciones:** Realizar cálculos aritméticos
3. **Salida:** Mostrar resultados

Estructura básica:

```
javascript
```

```
1 // 1. Obtener valores
2 let numero1 = parseFloat(prompt("Ingresa el primer número"));
3 let numero2 = parseFloat(prompt("Ingresa el segundo número"));
4
5 // 2. Realizar operaciones
6 let suma = numero1 + numero2;
7 let resta = numero1 - numero2;
8 let multiplicacion = numero1 * numero2;
9 let division = numero1 / numero2;
10
11 // 3. Mostrar resultados
12 console.log("Suma:", suma);
13 console.log("Resta:", resta);
14 alert("Resultado de la suma: " + suma);
```

Consideraciones:

- Usar `parseFloat()` o `parseInt()` para convertir strings a números
- Validar que los valores no sean NaN (Not a Number)
- Manejar división entre cero

Pregunta de reflexión 4: ¿Qué sucede si el usuario ingresa texto en lugar de números? ¿Cómo lo evitarías?

Ejemplo

javascript

```

1 // Calculadora completa
2 let num1 = parseFloat(prompt("Ingresa el primer número:"));
3 let num2 = parseFloat(prompt("Ingresa el segundo número:"));
4
5 // Validación
6 if (isNaN(num1) || isNaN(num2)) {
7   alert("Error: Debes ingresar números válidos");
8 } else {
9   // Operaciones
10  let suma = num1 + num2;
11  let resta = num1 - num2;
12  let mult = num1 * num2;
13  let div = num2 !== 0 ? num1 / num2 : "No se puede dividir entre cero";
14
15  // Mostrar resultados
16  console.log("=== RESULTADOS ===");
17  console.log("Suma:", suma);
18  console.log("Resta:", resta);
19  console.log("Multiplicación:", mult);
20  console.log("División:", div);
21
22  alert("Suma: " + suma + "\nResta: " + resta);
23 }

```

Ejercicio 1.4

Crea una calculadora que:

1. Pida dos números al usuario
2. Pregunte qué operación quiere realizar (+, -, *, /)
3. Realice la operación correspondiente
4. Muestre el resultado con alert() y console.log()
5. Maneje el error de división entre cero

javascript

```
1 // Tu código aquí:  
2  
3  
4  
5  
6
```