Introducción a JavaScript

1. Tipos de Datos en JavaScript

JavaScript es un lenguaje con tipado dinámico, lo que significa que el tipo de una variable es determinado por el valor asignado. Los principales tipos de datos incluyen:

- Cadena (String): Texto encerrado en comillas, `let mensaje = "Hola Mundo";`
- Número (Number): Incluye tanto enteros como decimales, `let edad = 30;`
- Booleano (Boolean): Valores `true` o `false`, `let isActive = true;`
- Indefinido (Undefined): Una variable declarada pero no inicializada tiene el valor `undefined`, `let dato;`
- Nulo (Null): Un valor que representa "sin valor", `let vacio = null;`
- Objeto (Object): Almacena datos complejos en propiedades, `{ nombre: "Pepe", edad: 30 }`
- Array: Un tipo especial de objeto, que contiene una lista de valores, `let frutas = ["manzana", "banana"];`

Ejemplos

JavaScript

```javascript
let entero = 8;
let float = 2.34;
let booleano = true;
let indefinido = undefined;
let nulo = null;
```

2. Declaración y Ámbito de Variables

- var: Ámbito global o de función, puede ser redeclarada.
- let: Ámbito de bloque, no puede ser redeclarada dentro del mismo bloque.
- const: Ámbito de bloque, no puede ser reasignada ni redeclarada.

JavaScript

```
var mensajeVar = "Hola";
let mensajeLet = "Hola desde let";
const CONSTANTE = 300;
```

3. Operadores en JavaScript

- Operadores aritméticos: `+`, `-`, `*`, `/`, `%`
- Operadores de comparación: `=`, `==`, `≠`, `≠=`, `<`, `⩽`, `>`, `⩾`
- Operadores lógicos: `&&`, `||`, `!`

Ejemplo

JavaScript

```
let suma = 7 + 1;
let multiplicacion = 7 * 2;
let esIgual = (5 === "5"); // false
```

4. Estructuras Condicionales

- if...else: Evalúa una condición y ejecuta un bloque de código dependiendo de si es verdadera o falsa.
- switch: Evalúa una expresión y ejecuta el bloque de código correspondiente al caso coincidente.

Ejemplo de if...else

JavaScript

```
let numero = 5;
if (numero > 10) {
    console.log("Mayor que 10");
} else {
    console.log("Menor o igual a 10");
}
```

Ejemplo de switch

JavaScript

```
let valor = "A";
switch (valor) {
    case "A":
        console.log("El caso era A");
        break;
    case "B":
        console.log("El caso era B");
        break;
    default:
        console.log("Caso no definido");
}
```

5. Funciones en JavaScript

Función Declarada

JavaScript

```
function suma(a, b) {
    return a + b;
}
console.log(suma(5, 3));
```

Funciones Flecha (Arrow Functions)

Sintaxis concisa introducida en ES6.

JavaScript

```
const cuadrado = (x) => x * x;
console.log(cuadrado(4));
```

Función Anónima

JavaScript

```
let multiplicar = function (a, b) {
    return a * b;
};
console.log(multiplicar(2, 3));
```

Callback

Permite pasar una función como argumento a otra.

JavaScript

```
function operar(a, b, callback) {
    return callback(a, b);
}
console.log(operar(10, 20, (x, y) => x * y));
```

6. Estructuras de Control e Iterativas

for

JavaScript

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

for...of

Ideal para iterar sobre elementos de un array.

JavaScript

```
let frutas = ["manzana", "banana", "fresa"];
for (let fruta of frutas) {
    console.log(fruta);
}
```

forEach

Método específico para iterar arrays.

JavaScript

```
let numeros = [1, 2, 3, 4];
numeros.forEach((numero) => console.log(numero));
```

while

JavaScript

```
let i = 0;
while (i < 3) {
    console.log(i);
    i++;
}
```

7. Manejo de Arrays

- Acceder a elementos: `array[indice]`
- Modificar elementos: `array[indice] = valor`
- Métodos de arrays:
  - `push()`, `pop()`, `shift()`, `unshift()`: Añadir y quitar elementos.
  - `indexOf()`: Buscar un elemento.
  - `map()`, `filter()`, `reduce()`: Manipulación avanzada.

Ejemplo

JavaScript

```
let numeros = [1, 2, 3, 4, 5];
let dobles = numeros.map(n => n * 2); // [2, 4, 6, 8, 10]
```

8. Manipulación del DOM

Selectores

- `document.getElementById("id")`
- `document.querySelector(".clase")`

Ejemplo

JavaScript

```
let titulo = document.getElementById("titulo");
console.log(titulo.textContent);
```

9. Eventos

Los eventos permiten ejecutar funciones cuando ocurren ciertas acciones (clics, presiones de teclado, etc.).

Tipos de Eventos

- Eventos de Ratón: `click`, `contextmenu`, `dblclick`, `mouseover`
- Eventos de Teclado: `keydown`, `keyup`, `keypress`
- Eventos de Navegador: `load`, `resize`, `beforeunload`

Ejemplo de Evento de Click

JavaScript

```
let boton = document.querySelector("#miBoton");
boton.addEventListener("click", () => {
    console.log("¡Botón clickeado!");
});
```

10. Consola y Salida de Información

- console.log(): Para imprimir mensajes.
- console.error(): Para mensajes de error.
- console.warn(): Para advertencias.
- console.table(): Para representar objetos en tablas.

Ejemplo

JavaScript

```
let usuarios = [
    { id: 1, nombre: "Ana" },
    { id: 2, nombre: "Luis" }
];
console.table(usuarios);
```

11. Programación Orientada a Objetos (POO)

Definición de Clases y Objetos

JavaScript

```
class Persona {
    constructor(nombre, edad) {
        this.nombre = nombre;
        this.edad = edad;
    }
    saludar() {
        console.log(`Hola, soy ${this.nombre} y tengo ${this.edad} años.`);
    }
}

let persona1 = new Persona("Juan", 25);
persona1.saludar();
```