



## Cuadro Sinóptico sobre JavaScript



### Características básicas del lenguaje

- **Interpretado:** No necesita compilación.
- **Dinámico:** No se declara el tipo de dato.
- **Multiparadigma:** Soporta programación funcional y orientada a objetos.

### Estructuras de control

- **Condicionales**
  - if, else, switch
  - Ej: `if (x > 5) { ... }`
- **Bucles**
  - for, while, do...while
  - Ej: `for (let i = 0; i < 10; i++)`

### Tipos de datos

- **Primitivos**
  - string, number, boolean, null, undefined, symbol, bigint
  - Ej: "Hola", 42, true
- **Referencia**
  - Objetos, arrays, funciones
  - Ej: `{ nombre: "Ana" }`, `[1, 2, 3]`

### Funciones

- **Declaradas**
  - `function saludar() { }`
- **Expresadas**
  - `const suma = function(a, b) { return a + b }`
- **Flecha**
  - `const resta = (a, b) => a - b`

### Objetos y arrays

- **Objeto**
  - Colección de pares clave-valor.
  - Ej: `{ nombre: "Luis", edad: 20 }`
- **Array**
  - Lista ordenada.
  - Ej: `[10, 20, 30]`

### Eventos y manipulación del DOM

- **Eventos**
  - onclick, onmouseover, onchange
  - Ej: `boton.addEventListener('click', funcion)`
- **DOM**
  - Modificación de elementos HTML.
  - Ej: `document.getElementById("titulo").textContent = "Hola"`

### Asincronia

- **Callbacks**
  - Función pasada como argumento.
  - Ej: `setTimeout(() => { ... }, 1000)`
- **Promesas**
  - `.then(), .catch()`
  - Ej: `fetch(url).then(res => res.json())`
- **Async/Await**
  - Sintaxis más clara.
  - Ej: `async function cargar() { let data = await fetch(url); }`