

# JavaScript

## Básico

The JavaScript logo, consisting of the letters "JS" in a bold, black, sans-serif font, centered within a solid yellow square. A thin brown L-shaped line is positioned at the bottom right corner of the yellow square.

**JS**

# Índice

1. Breve Historia.
2. Variables.
3. Operadores.
4. Condicionales.
5. Array.
6. Bucles.
7. Funciones.
8. Clases y Objetos.

# Breve Historia

JavaScript es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, que se utiliza principalmente en el desarrollo web. Fue creado originalmente por **Brendan Eich** en 1995 mientras trabajaba para Netscape Communications. En ese momento, el objetivo era desarrollar un lenguaje de programación sencillo que permitiera la interacción con elementos de la página web en el navegador Netscape Navigator.

En sus primeros días, JavaScript se llamaba LiveScript, pero poco después, para aprovechar el creciente interés en el lenguaje de programación Java, se cambió el nombre a JavaScript. Sin embargo, a diferencia de Java, JavaScript no está relacionado directamente con el lenguaje de programación Java.

# Variables

Contenedores en los que puedes almacenar valores

Globales y Locales

```
var miNombre = "Marc";  
function mostrarMiNombre() {  
    var miNombre = "Carlos";  
    console.log(miNombre);  
}  
mostrarMiNombre(); // Muestra Carlos, prioridad en una función.  
console.log(miNombre); // Muestra Marc, se usa en todo el programa.
```

**String** → Secuencia de texto (cadena).

→ let miVariable = 'Marc';

**Number** → Número (sin comillas)

→ let mi Variable = 10;

**Boolean** → Valor verdadero/Falso (sin comillas) → let miVariable = true;

# Operadores (+, -, \*, /, =, ===, !, !==)

Operador Igualdad y igualdad estricta:

```
console.log(9 == 9); // true
console.log("9" == 9); // Se convierten al mismo tipo de dato por eso es true
console.log(9 === "9"); // En este caso pedimos comparación Estricta porque el dato no se convierte
```

Operador Desigualdad y desigualdad estricta:

```
console.log(1 != "1"); // false, es distinto ya que no mira el tipo de dato
console.log(1 !== "1"); // true, si son distintos ya que el tipo de dato es diferente
```

# Condicionales (if, switch / case )

Estructuras de código que permiten comprobar si una expresión devuelve true o no.

## IF

```
let operacion = 5 + 2;
if (operacion > 7) {
    console.log("Es mayor a 7");
} else if (operacion <= 7 && operacion >= 2) {
    console.log("Está entre 7 y 2");
} else {
    console.log("Es menor de 2");
}
```

## SWITCH

```
let sumar = 5 + 3;
switch(sumar) {
    case 0:
        console.log("El resultado es 0");
        break;
    case 7:
        console.log("El resultado es 7");
        break;
    default:
        console.log("El resultado no es ni 0 ni 7");
}
```

# Array

Te permite almacenar varios valores en una sola referencia.

Ejemplo de Array dentro de Array

```
var accesoArreglo = [[1, 2, 3], [4, 5, 6], [7, 8, 9]];

/*
Arreglo:          [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Índices:           0           1           2
Índices internos: 0 1 2      0 1 2      0 1 2
*/

console.log(accesoArreglo[0]); // [1, 2, 3]
console.log(accesoArreglo[1]); // [4, 5, 6]
console.log(accesoArreglo[2]); // [7, 8, 9]

console.log(accesoArreglo[1][2]); // 6
console.log(accesoArreglo[2][0]); // 7
console.log(accesoArreglo[0][0]); // 1
```

# Bucles

```
const amigos = ["Freddy", "Alberto", "Joan"];

for (let i = 0; i < 3; i++){
    console.log(amigos[i]);
}

let i = 0;
while (i <= amigos.length-1) {
    console.log(amigos[i])
    i++;
}

amigos.forEach(function (amigo) {
    console.log(amigo);
});
```



# Funciones

```
// Argumentos
function sumar(a, b) { // a y b son parametros
    var suma = a + b;
    console.log("El resultado de " + a + " + " + b + " es: " + suma);
}
sumar(5, 3); // Estos son los argumentos que asignamos a los parametros a y b
sumar(8, 9);
sumar(2, 4);
sumar(5, 5);
```

# Objetos

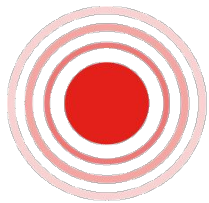
- Javascript está diseñado en un paradigma simple de programación orientada a objetos. Los objetos se asignan con llaves "{ }".
- Un objeto es una colección de propiedades, y una propiedad es una asociación entre una *key* o nombre y un valor

```
const perro = {  
  nombre: "Jack",  
  edad: 10,  
  raza: "San Bernardo",  
}
```

# Clases y métodos

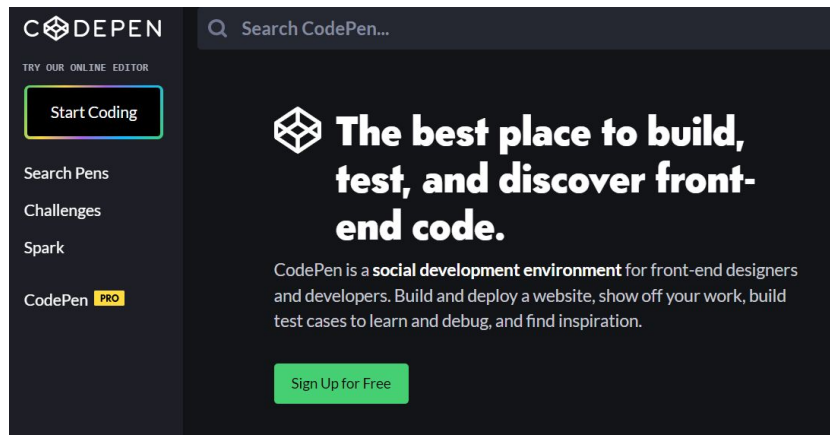
- Las clases son una mejora sintáctica sobre la herencia basada en prototipos de JavaScript.
- Un método es básicamente funciones que pertenecen a una clase.
- Las clases en JavaScript tienen una sintaxis mucho más clara y simple para crear objetos y lidiar con la herencia

```
class Rectangulo {  
  
    constructor( ancho, longitud ){  
        this.base  = ancho;  
        this.altura = longitud;  
    }  
  
    getAreaRectangulo(){  
        return ( this.base * this.altura )  
    }  
  
}
```



**EN VIVO**

*Ejemplo de Clases y Objetos*



<https://codepen.io>