



# Lenguaje de programación ⚡

## Lenguaje de programación

Un **lenguaje de programación**, en palabras simples, es el **conjunto de instrucciones** a través del cual los humanos interactúan con las computadoras. Un lenguaje de programación nos permite comunicarnos con las computadoras a través de algoritmos e instrucciones escritas en una sintaxis que la computadora entiende e interpreta en lenguaje de máquina.

Existen **docenas de lenguajes de programación** utilizados en la industria hoy en día. Algunos lenguajes de programación populares incluyen C++, C#, Visual Basic, Go, Ruby, **JavaScript**, Java y Python, por mencionar algunos.

# JavaScript ⚡

JavaScript es un **lenguaje de programación** que nació para dinamizar la web, pero que hoy por hoy se usa en “todos lados”.

Si bien se llaman parecido, Java y JavaScript son dos mundos diferentes.

Es **multi-paradigma**, es decir que permite varios estilos de programación.

Está basado en el estándar **ECMAScript**.

## JavaScript - ¿Dónde se escribe nuestro JavaScript?

De la misma manera que lo hacemos con CSS, podemos escribir nuestro JS de dos maneras:

- Directamente en el HTML, dentro de las etiquetas **SCRIPT**
- En archivos separados, con la etiqueta **SCRIPT** y el atributo “**src**” apuntando al archivo. Por lo general lo colocamos al final, justo antes de cerrar la etiqueta BODY

```
<html>
<head>
  <style>
    h1 {
      color: red;
    }
  </style>
</head>
<body>
  <h1>Hola</h1>
  <p>Estamos en el curso <b>FED de Senpai</b></p>
  <script>
    setTimeout(() => {
      alert('Bienvenidos!');
    }, 3000);
  </script>
  <script src="./script.js"></script>
</body>
</html>
```

## JavaScript - Variables y constantes

### ¿Cuándo usar var, let y const?

Las **variables** y **constantes** nos permiten **guardar información**.

Podemos crearlas utilizando las palabras **var**, **let** y **const**, aunque se aconseja dejar de usar **var**.

Las variables y constantes son prácticamente lo mismo, con la diferencia que la reasignación de una constante nos daría un error.



```
// Declaración.  
let contador;  
  
// Asignación - dar un valor.  
contador = 2;  
  
// Declaración + Asignación  
let nombre = 'Senpai';  
  
// Constante  
const pi = 3.1415  
  
// Reasignación  
nombre = 'Academy';  
pi = 5; // Error!
```

## JavaScript - Tipos de datos: valores primitivos y objetos.

Hay varias diferencias entre los **valores primitivos** y los **objetos**, las cuales iremos viendo en el correr del curso.

Los **valores primitivos no están compuestos por otros tipos de datos**, mientras que los objetos pueden alojar varios.

Otros conceptos a tener en cuenta (pero revisar más adelante): inmutabilidad, herencia, referencias, instancias...

```
/* Valores primitivos */

// String (texto)
const nombre = "Sebastián";

// Number (número)
let edad = 27;

// Boolean (booleano)
let tieneSueño = true;

// null
let laNadaMisma = null;

// undefined
let noEstaDefinido;

/* Objetos */

let auto = {
  marca: 'Volkswagen',
  modelo: 'Beatle', // Fusca
  año: 1962
}

// Array (arreglos)
let profesiones = ['Desarrollador', 'Formador', 'Músico?'];
```



## JavaScript - Operadores algebraicos

Podemos utilizar operadores algebraicos para calcular valores.

Si bien lo normal es hacerlo con números, también podemos hacerlo con otros tipos de valores, e incluso mezclarlos.

Mezclar tipos de datos en operaciones algebraicas es generalmente un error.

¿Qué resultado tiene la siguiente suma:

**“Hola” + “Senpai” + 2020?**



```
// Teniendo dos variables:
const primerValor = 10;
const segundoValor = 6;

// Suma
const suma = primerValor + segundoValor; // 16

// Resta
const resta = primerValor - segundoValor; // 4

// Multiplicación
const multi = primerValor * segundoValor; // 60

// División
const div = primerValor / segundoValor; // 1.66..

// Módulo
const mod = primerValor % segundoValor; // 4
```

## JavaScript - console.log()

Pueden utilizar **console.log()** para imprimir en consola todo tipo de cosas.

Por ejemplo, pueden utilizarlo varias veces en el flujo del código, para ir viendo paso a paso lo que sucede (más adelante puede que veamos cómo debuggear en el navegador de manera elegante).

```
console.log('Hola senpai');
```

# JavaScript - Operadores comparativos y lógicos

Los operadores comparativos siempre devuelve **true** o **false**, y nos permiten comparar valores.

```
// Teniendo la siguiente variable
const miValor = 10;

const mayorQueDiez = miValor > 10 // False
const menorQueDiez = miValor < 10 // False
const menorOIgualQueDiez = miValor <= 10 // True
const esDiez = primerValor === 10; // True

// NOT ! (negación)
const cuatro = 4; // 4
const esCuatro = cuatro === 4; // True
const noEsCuatro = !esCuatro; // False

// AND &&
const totalCompra = 1500;
const balanceDeCuenta = 12000;
const cuentaBloqueada = false;
const puedeComprar = !cuentaBloqueada && (total < balanceDeCuenta)

// OR ||
const mostrarAyuda = cuentaBloqueada || (balanceDeCuenta < total)
```

# JavaScript - Control de flujo

Podemos modificar el comportamiento de nuestro código utilizando **condicionales** o **switches**.

Se utiliza **if/elseif/else** en la mayoría de los casos de control de flujo.

Se utiliza **switch** cuando se evalúa un único valor y este puede tomar varios estados.

```
// Condicionales
let mensaje;

if (itemsCarrito > 0) {
  mensaje = 'Tienes ' + itemsCarrito + ' en tu carrito';
} else {
  mensaje = 'Tu carrito está vacío 😞'
}

// Switch
const estado = 'Sin stock';

switch(estado) {
  case 'Sin stock':
    // Deshabilitar botón de comprar
    // Mostrar mensaje de re-stock
    break;
  case 'En stock':
    // Habilitar botón de comprar
    // Mostrar mensaje de unidades disponibles
    break;
  case 'Descontinuado':
    // Ocultar botón de comprar
    // Mostrar mensaje de producto descontinuado
    break;
}
```

¿Para qué nos  
sirve javascript  
entonces? 🙄

## JavaScript - ¿Para qué nos sirve javascript entonces?

Una vez nos adentramos en las posibilidades que nos ofrece un **lenguaje de programación**, vamos a poder:

- Cambiar (agregar/eliminar) que clases tiene aplicada un elemento seleccionado.
- Para crear nuevos elementos HTML y colocarlos.
- Para darle interactividad y funcionalidad real a nuestro sitio. Enviar un formulario, validar datos, etc.

# LINKS



## LINKS

- **ECMAScript (2019 y 2022)**
  - <https://tc39.es/ecma262/>
  - <https://262.ecma-international.org/10.0/>
- **Console.log**
  - <https://developer.mozilla.org/es/docs/Web/API/Console>
- **JS**
  - [https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
  - <https://overreacted.io/on-let-vs-const/>
  - <https://www.freecodecamp.org/news/var-let-and-const-whats-the-difference/>
  - <https://www.freecodecamp.org/learn>
- **Ejercicios**
  - [https://www.w3schools.com/js/exercise\\_js.asp?filename=exercise\\_js\\_variables1](https://www.w3schools.com/js/exercise_js.asp?filename=exercise_js_variables1)





[gustavgueez](#)



[gustavgueez](#)



[gustavgueez](#)

**GUSTAVO RODRIGUEZ**

FULL STACK DEVELOPER  
SOLCRE