



JavaScript

clase n°4

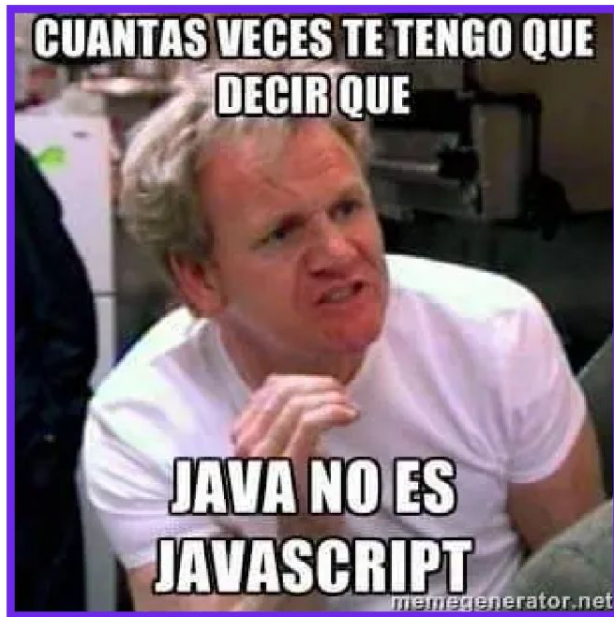
*REC



recordá
poner a grabar la clase

introducción a JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para **crear contenido web interactivo**. Es uno de los tres pilares fundamentales de la web junto con HTML y CSS.



vamos a mentimeter



<https://www.mentimeter.com/es-ES>

Actividad iniciadora: ¿Qué cosas creen que se pueden realizar utilizando JavaScript?

variables y tipos de datos

Las **variables** son **contenedores para almacenar datos**. En JavaScript, las palabras clave **var**, **let** y **const** se utilizan para declarar variables, pero tienen diferencias importantes en cuanto a su alcance (scope), re-asignación y comportamiento en general.

Tipos de datos:

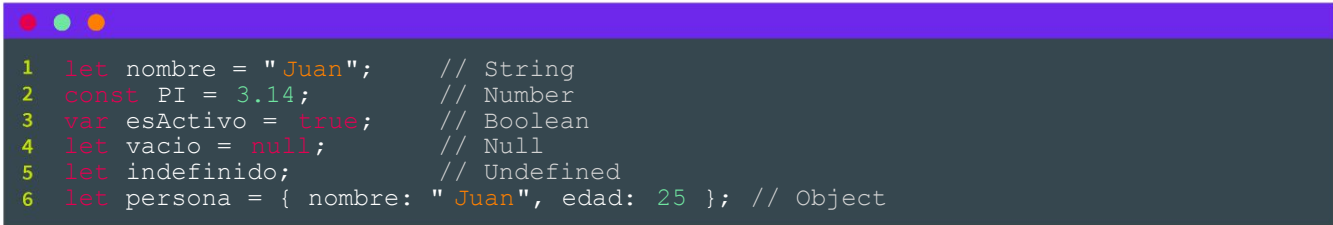
- **Number:** Representa tanto números enteros como decimales.
- **String:** Cadenas de texto. Se definen usando comillas simples o dobles.
- **Boolean:** Solo puede tener dos valores: true o false.
- **Null:** Representa "nada" o un valor vacío.
- **Undefined:** Indica que una variable no ha sido asignada a ningún valor.
- **Object:** Colección de propiedades.

¿para qué sirven las variables?

Las variables **nos permiten almacenar y manipular datos** en nuestros programas.

Por ejemplo, podemos usar variables para almacenar la información del usuario, realizar cálculos, controlar el flujo del programa, entre otros.

Sintaxis:



```
1 let nombre = "Juan";    // String
2 const PI = 3.14;        // Number
3 var esActivo = true;     // Boolean
4 let vacio = null;        // Null
5 let indefinido;          // Undefined
6 let persona = { nombre: "Juan", edad: 25 }; // Object
```

operadores básicos y concatenación

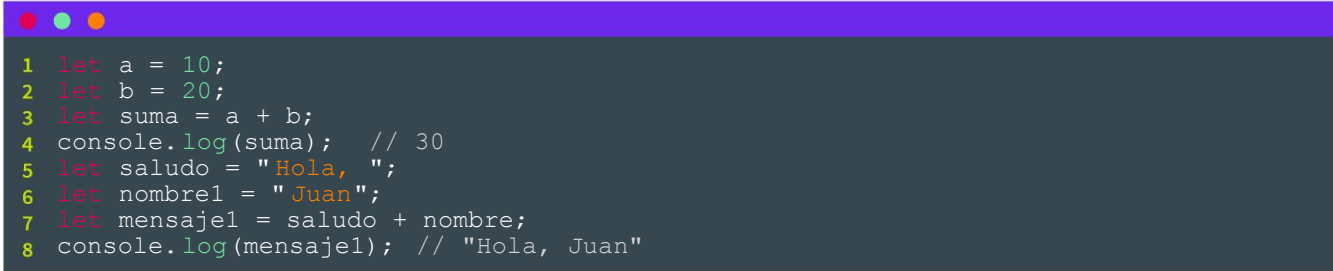
Los **operadores** se utilizan para realizar **operaciones sobre variables y valores**. Los operadores básicos incluyen:

aritméticos + (suma), - (resta), * (multiplicación), / (división), % (módulo)

de asignación =, +=, -=, *=, /=

de comparación ==, ===, !=, !==, >, <, >=, <=

La concatenación se utiliza para unir dos o más cadenas de texto.



```
1 let a = 10;
2 let b = 20;
3 let suma = a + b;
4 console.log(suma); // 30
5 let saludo = "Hola, ";
6 let nombre1 = "Juan";
7 let mensaje1 = saludo + nombre1;
8 console.log(mensaje1); // "Hola, Juan"
```


funciones

Son **bloques de código reutilizables** que realizan una tarea específica. Se definen con la palabra clave **function** seguida del nombre de la función, paréntesis y un bloque de código.

Las funciones **permiten organizar y reutilizar código**, hacer que el código sea más limpio y manejable, y realizar tareas repetitivas de manera eficiente.



partes de una funciones

- **Nombre de la función:** Identifica la función.
- **Parámetros:** Valores que se pasan a la función.
- **Cuerpo de la función:** El bloque de código que se ejecuta cuando se llama a la función.
- **Valor de retorno:** El resultado que devuelve la función (opcional).

Parámetros: Una función puede tener uno o más parámetros. Los parámetros actúan como variables locales dentro de la función.

```
1 // Funcion con un parametro
2 function saludar(nombre) {
3     return "Hola, " + nombre;
4 }
5 let mensaje = saludar("Juan");
6 console.log(mensaje); // Devuelve: "Hola, Juan"
7 // Funcion con multiples parametros
8 function sumar (a,b) {
9     return a + b;
10 }
11 let resutlado = sumar(5,3);
12 console.log(resutlado); // Devuelve: 8
```

actividad

1. Función con un parámetro:

- Declara una función llamada *mostrarEdad* que tome un parámetro *edad* y retorne una cadena que diga "Tienes X años".
- Llama a la función con tu edad y muestra el resultado con `console.log`.
- Comparte tu resultado en el chat.

2. Función con múltiples parámetros:

- Declara una función llamada *calcularArea* que tome dos parámetros *base* y *altura* y retorne el área de un rectángulo.
- Llama a la función con valores para la base y la altura y muestra el resultado con `console.log`.
- Comparte tu resultado en el chat.

condicionales

Se utilizan para ejecutar diferentes bloques de código según una condición.

Tipos de condicionales:

if Ejecuta un bloque de código si la condición es verdadera.

else Ejecuta un bloque de código si la condición if es falsa.

else if Verifica múltiples condiciones.

switch Selecciona uno de muchos bloques de código a ejecutar.

Sintaxis con ejemplos

```
1 // if
2 let edad = 18;
3 if (edad >= 18) {
4     console.log("Eres mayor de edad");
5 }
6 // else
7 let edad1 = 16;
8 if (edad >= 18) {
9     console.log("Eres mayor de edad");
10 } else {
11     console.log("Eres menor de edad");
12 }
13 //else if
14 let nota = 85;
15 if (nota >= 90) {
16     console.log("Excelente");
17 } else if (nota >= 70) {
18     console.log("Aprobado");
19 } else {
20     console.log("Necesitas Mejorar");
21 }
```

```
1 // Switch
2 let dia = 3;
3 switch (dia) {
4     case 1:
5         console.log("Lunes");
6         break;
7     case 2:
8         console.log("Martes");
9         break;
10    case 3:
11        console.log("Miércoles");
12        break;
13    case 4:
14        console.log("Jueves");
15        break;
16    case 5:
17        console.log("Viernes");
18        break;
19    case 6:
20        console.log("Sábado");
21        break;
22    case 7:
23        console.log("Domingo");
24        break;
25    default:
26        console.log("Día no válido");
27 }
```

estructuras repetitivas

Las estructuras repetitivas se utilizan para ejecutar un bloque de código varias veces.

Tipos:

for Repite un bloque de código un número específico de veces.

forEach Ejecuta una función para cada elemento de un array.

while Se utiliza cuando no sabes cuántas veces se debe ejecutar el bloque de código, pero quieres seguir ejecutándolo mientras una condición sea verdadera.

do-while es similar al while, pero se asegura de que el bloque de código se ejecute al menos una vez antes de verificar la condición.

Sintaxis con ejemplos

```
1 // Ejemplo for - Imprimir los números del 0 al 4
2 for (let i = 0; i < 5; i++) {
3     console.log(i);
4 }
5 // Ejemplo - forEach - Imprimir cada número en el array
6 let numeros = [1, 2, 3, 4, 5];
7 numeros.forEach(function(numero) {
8     console.log(numero);
9 });
10 // Ejemplo con while
11 let i = 0;
12 while (i < 5) {
13     console.log(i);
14     i++;
15 }
16 // Ejemplo con do-while
17 do {
18     console.log(i);
19     i++;
20 } while (i < 5);
```

actividad

*Dado 5 números, el programa debe detectar cuántos de esos números son pares
Independientemente de si lo logran o no, usemos CHATGPT para consultarle cómo se podría solucionar
este problema.*



revolución*
digital_