
SENPAI
ACADEMY

Operadores lógicos 

SENPAI
ACADEMY

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)
```

SENPAI ACADEMY

JavaScript - Operadores comparativos y lógicos

Operador	Símbolo	Ejemplo
Mayor que	>	`3 > 4 false` `8 > 5 true`
Mayor o igual que	>=	`3 >= 3 true` `3 >= 4 false`
Menor que	<	`4 < 6 true` `7 < 4 false`
Menor igual que	<=	`2 <= 2 true` `3 <= 2 false`
Igual	==	`8 == 8 true` `7 == 8 false`
Distinto	!=	`5 != 6 true` `5 != 5 false`

```
console.log("3 > 4",3 > 4);
console.log("8 > 5",8 > 5);
console.log("3 >= 3",3 >= 3);
console.log("3 >= 4",3 >= 4);
console.log("4 < 6",4 < 6);
console.log("7 < 4",7 < 4);
console.log("2 <= 2",2 <= 2);
console.log("3 <= 2",3 <= 2);
console.log("8 == 8",8 == 8);
console.log("7 == 8",7 == 8);
console.log("5 != 6",5 != 6);
console.log("5 != 5",5 != 5);
```

SENPAI ACADEMY

JavaScript - Operadores comparativos y lógicos

Los operadores lógicos responden a las operaciones algebraicas del código binario.

expresión A	expresión B	&& (AND)	|| (OR)
V	V	V	V
V	F	F	V
F	F	F	F

Esta operación es conmutativa (cambiar el orden de los operandos no altera el resultado).
El otro operador frecuente es el NOT (!), el cual devuelve el valor opuesto al actual;. !Verdadero == FALSO y !Falso == Verdadero

SENPAI ACADEMY

Control de flujo

SENPAI
ACADEMY

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;
}
```

SENPAI ACADEMY

¿Cómo pensar?

SENPAI
ACADEMY

Hay una diferencia entre **saber programar** y **saber escribir código**.

- Saber escribir código es, saber la sintaxis del lenguaje, saber las posibilidades del lenguaje, etc.

- Mientras que saber programar es saber pensar como programador, pensar en algoritmos.

¿Cómo podemos aprender a pensar como programador o por lo menos practicarlo? **Tratar de pensar todo como algoritmos**

EJEMPLO: Si quisiéramos describir la acción de salir de la cama y llegar al trabajo (En una época sin pandemia).

El algoritmo sería el siguiente:

1. Salir de la cama
2. Quitarle el pijama
3. Darse una ducha
4. Vestirse
5. Desayunar
6. Salir de casa y viajar al trabajo.

Pseudocódigo

SENPAI
ACADEMY

JavaScript - Pseudocódigo

Escribir **pseudocódigo** es escribir instrucciones en lenguaje natural de forma de poder luego **traspasarlas** cómodamente al lenguaje de programación que corresponda.

Si bien en la diaria no se utiliza este lenguaje de programación, es bueno conocerlo a esta altura para poder tratar de pensar todo de forma lógica y poder luego pasarle de forma eficaz al código.

Consideremos el siguiente enunciado:

Una clase de diez alumnos hizo un examen. Las calificaciones (enteros en el rango de 0 a 100) correspondientes a este examen están a su disposición. Determine el promedio de la clase en este examen.

Sabemos que el promedio de algo es la suma de todos sus casos (en este caso suma de calificaciones), dividido la cantidad de calificaciones sumadas.

Por lo que sabemos que vamos a sumar valores a un total, que vamos a iterar sobre un listado de alumnos y luego vamos a imprimir en pantalla el resultado. Por lo tanto en pseudocódigo tenemos lo siguiente:

JavaScript - Pseudocódigo

- Inicializar total en 0
- Inicializar nro_alumno a 1
- Mientras nro_alumno menor o igual a 10
  - Leer siguiente calificación
  - Sumar calificación a total
  - sumar 1 a nro_alumno
- Calcular promedio = total / nro_alumno
- Imprimir promedio

```
const qualifications = [5,2,3,7,8,9,10,11,1,10]
let total  = 0;
let counter = 1;
while (counter <= 10){
  total += qualifications[counter -1] // Los arreglos arrancan en 0
  counter ++;
}
let average = total/counter;
console.log(average)
```

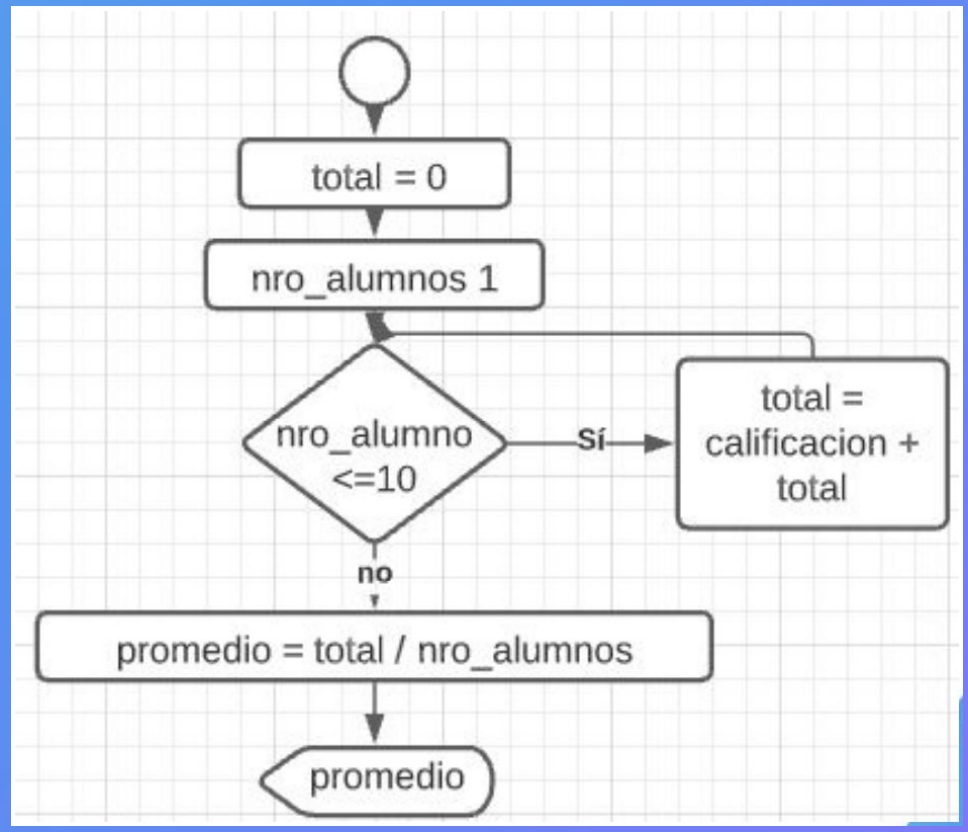
SENPAI ACADEMY

Diagramas de flujo 

SENPAI
ACADEMY

JavaScript - Diagramas de flujo

Otra forma de visualizar código antes de escribirlo es por intermedio de diagramas de flujo:



SENPAI ACADEMY

LINKS 🔗

SENPAI
ACADEMY

LINKS

- **Operadores de comparación:**
  - https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Expressions_and_Operators#Comparacion
  - https://developer.mozilla.org/es/docs/Learn/JavaScript/Building_blocks/conditionals
- **If y Switches**
  - https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/switch
  - https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Statements/if...else
- **Pseudocódigo:**
  - https://medium.com/@galiciandeveloper/extrapolando-el-pseudoc%C3%B3digo-a-la-vida-real-5fc121ace470
  - https://es.wikipedia.org/wiki/Pseudoc%C3%B3digo
- **Diagramas de flujo:**
  - https://es.wikipedia.org/wiki/Diagrama_de_flujo

SENPAI ACADEMY

SENPAI
ACADEMY

gustavguez

gustavguez

gustavguez

GUSTAVO RODRIGUEZ

FULL STACK DEVELOPER
SOLCRE