

# Bootcamp Java Developer

Fase 2 - Java Web Developer  
Módulo 17



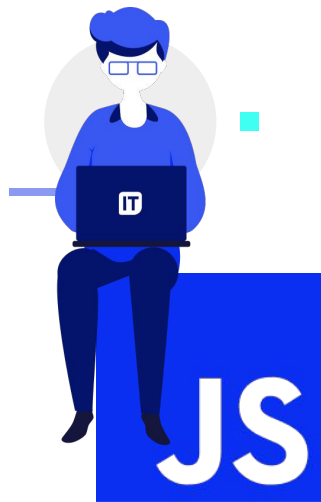
# Operadores

# Operadores: introducción

Como vimos, poco se puede hacer en **JS** si no contamos con **variables**. Los **operadores nos brindan herramientas** para generar que nuestra aplicación o programa cumpla su fin.

Los operadores permiten trabajar con variables, realizar operaciones matemáticas y compararlas. También, avanzar en el programa y cumplir el fin que nos hayamos propuesto.

Existen diferentes **tipos de operadores** para trabajar en **JavaScript**, veámoslos a continuación.



# Asignación, incremento y decremento

El **operador de asignación** se utiliza para **guardar un valor en una variable**. Ya lo hemos utilizado previamente, veamos el ejemplo de arriba a la derecha.

```
var iva = 21;
```

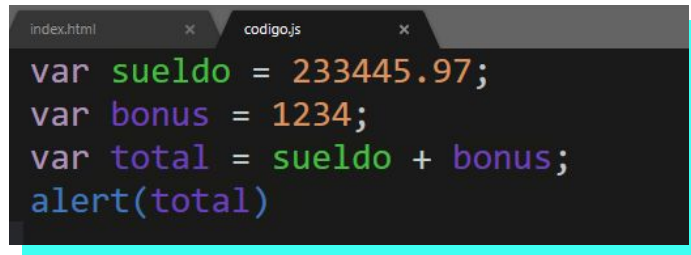
El **operador de incremento** se indica mediante el **prefijo ++**. **Incrementa** la variable **en una unidad**, como se muestra en la segunda imagen.

El **operador de decremento** se indica mediante el **prefijo --**. **Decrementa** la variable **en una unidad**.

```
index.html x código.js  
var numero = 12;  
++numero;  
alert(numero); // numero = 13
```

# Operadores matemáticos

JavaScript permite hacer **operaciones matemáticas**: suma (+), resta (-), multiplicación (\*) y división (/). Por ejemplo:



```
index.html x  codigo.js x
var sueldo = 233445.97;
var bonus = 1234;
var total = sueldo + bonus;
alert(total)
```

Esta página dice

234679.97

Aceptar

El operador matemático se puede combinar con el de asignación para lograr un resultado más rápido y conciso:

```
index.html x código.js x
var numero1 = 9;
numero1 += 4; // numero1 = numero1 + 4
numero1 -= 1; // numero1 = numero1 - 1
numero1 *= 2; // numero1 = numero1 * 2
numero1 /= 6; // numero1 = numero1 / 6
numero1 %= 5; // numero1 = numero1 % 5
```

También con el (+) es posible concatenar texto para obtener resultados accesibles y entendibles para el usuario, por ejemplo:

```
var sueldo = 233445.97;
var bonus = 1234;
var total = sueldo + bonus;
alert("El total del sueldo y el bonus es" + total)
```

Se puede concatenar información para lograr mejores resultados para el usuario, también incluir etiquetas de **HTML**:

```
index.html x código.js x
var sueldo = 233445.97;
var bonus = 1234;
var total = sueldo + bonus;
document.write("<p>El total del sueldo y el <strong>bonus</strong> es " + total + "</p>")
```

La última línea fue reemplazada por **document.write()**, ya que **las ventanas de alertas no soportan formato**, es decir que una etiqueta en una ventana de alerta se vería de la siguiente forma:

```
index.html x código.js x
alert('<p> Esta etiqueta, se ve de verdad acá </p>')
```

El resultado del slide anterior será el siguiente:

Esta página dice

`<p>` Esta etiqueta, se ve de verdad acá `</p>`

Aceptar

Un detalle interesante es que, seguramente, llamó la atención el **operador de módulo (%)**, ya que no es utilizado en matemática. Este operador se utiliza para calcular **el resto de la división entre 2 valores enteros. Tiene la misma precedencia que el producto y el cociente.**





# Operadores de negación

Hemos visto que existen **variables booleanas** (**true/false**), estos operadores siempre dan como resultado que algo sea **verdadero o falso**.

El operador de negación se utiliza para obtener el contrario o la negación de la variable.



```
index.html x codigo.js
var dato = 2;
var vacio = !dato;
alert(vacio) //resultado false
```

## Operadores and, or

La operación **and** obtiene un resultado donde **ambos valores deben ser verdaderos**:

```
index.html x código.js
var valorA = true;
var valorB = false;
resultado = valor1 && valor2; // resultado = false

valorA = true;
valorB = true;
resultado = valor1 && valor2; // resultado = true
```

En el caso de **or**, **los dos no pueden ser verdaderos**. Solo uno de los valores debe ser verdadero para obtener un **resultado true**.

```
index.html x código.js x
var valorA = true;
var valorB = false;
resultado = valor1 || valor2; // resultado = true

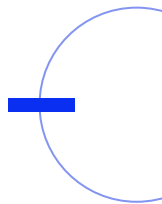
valorA = false;
valorB = false;
resultado = valor1 || valor2; // resultado = false
```

# Operadores relacionales

Son iguales a los que usamos en matemáticas. Permiten realizar programas complejos y **comparar datos**, el resultado será también, al igual **que los operadores lógicos**, **true/false**.

```
index.html x código.js
var numero1 = 1;
var numero2 = 20;
resultado = numero1 > numero2; // resultado = false, mayor que
resultado = numero1 < numero2; // resultado = true, menor que

numero1 = 10;
numero2 = 10;
resultado = numero1 >= numero2; // resultado = true, mayor o igual
resultado = numero1 <= numero2; // resultado = true, menor o igual
resultado = numero1 == numero2; // resultado = true, igual en valor
resultado = numero1 != numero2; // resultado = false, no es igual
```



## Revisión

- Repasar qué es un operador.
- Trabajar con variables y operadores en sus **diferentes tipos**.
- Implementar operadores y mostrar los datos en una ventana.
- Mostrar datos producto de operadores a través de **document.write()**
- Aplicar todas las propiedades en el **proyecto integrador**.
- Realizar las preguntas necesarias al/la docente antes de continuar.



**¡Sigamos  
trabajando!**

