

# Operadores en Java



# Qué son los Operadores



Java proporciona muchos tipos de operadores que se pueden usar según la necesidad. Se clasifican según la funcionalidad que brindan

Sirven para realizar cálculos matemáticos, comparar valores, para unir identificadores y literales, para formar expresiones lógicas, toma de decisión etc.

aritméticos

relacionales

lógicos

asignación

unarios, incrementales, ternarios, combinados, de bits, de instancia



# Operadores Aritmético

- Se utilizan para realizar operaciones aritméticas simples en tipos de datos primitivos

Descripción	Símbolo	Expresión de ejemplo	Resultado
Multiplicación	*	2*7	14
División	/	6/3	2
Modulo o resto	%	5%2	1
Suma	+	5 + 3	8
Resta	-	5-3	2
Incremento	++	++2	3
Decremento	--	--2	1
Menos Unario	-	-(4 + 2)	-6

# Operadores Combinados

- Operadores combinados o compuestos: -=, +=, /= .\*= etc.
- $a += 5$  es equivalente:  $a = a + 5$

```
int a = 3;  
a += 5; // a vale 8
```

- $a /= b$  es equivalente a  $a = a / b$

# Operador

## Incremento y Decremento

- Utilizado para incrementar o decrementar el valor en 1
- Hay dos variedades de operador de incremento pre y post

Ejemplo	Nombre	Efecto
++a	Pre-incremento	Incrementa a en uno, y luego devuelve a.
a++	Post-incremento	Devuelve a, y luego incrementa a en uno.
--a	Pre-decremento	Decrementa a en uno, luego devuelve a.
a--	Post-decremento	Devuelve a, luego decrementa a en uno.

```
int a = 1;  
// En este primer caso, a valdrá 2 y b 2 también  
int b = ++a;  
a = 1;  
// Ahora, a sigue valiendo 2, pero b es ahora 1  
b = a++;
```

- Se pueden colocar tanto antes como después de la expresión que deseemos modificar pero sólo devuelven el valor modificado si están delante.

# Operador

## Ternario o condicional

- El operador ternario es una versión abreviada de la sentencia if-else.
- Tiene tres operandos y de ahí el nombre ternario:

```
(condición) ? valor1 : valor2
```

- Si la condición se cumple devuelve el primer valor y, en caso contrario, el segundo.
- El siguiente ejemplo asignaría a la variable 'a' un 2:

```
int a = 2 > 3 ? 1 : 2;
```

# Operadores Relacionales

- Se utilizan para comprobar relaciones de igualdad, mayor que, menor que
- Devuelven un resultado booleano después de la comparación, true si la comparación fue exitosa y false en caso contrario
- Se usan ampliamente en las sentencias if/else, así como en bucles.

Operador	Utilización	Resultado
>	a > b	verdadero si a es mayor que b
>=	a >= b	verdadero si a es mayor o igual que b
<	a < b	verdadero si a es menor que b
<=	a <= b	verdadero si a es menor o igual que b
==	a == b	verdadero si a es igual a b
!=	a != b	verdadero si a es distinto de b

# Operadores Lógicos

- Los operadores lógicos permiten evaluar expresiones lógicas y trabajan con operandos booleanos
- Realizan las operaciones lógicas de conjunción (AND), disyunción (OR) y negación (NOT)

Nombre	Operador	Utilización	Resultado
And	&&	A && B	verdadero cuando A y B son verdaderos. Evaluación condicional.
Or		A    B	verdadero cuando A o B son verdaderos. Evaluación condicional.
Not	!	!A	verdadero si A es falso.
And	&	A & B	verdadero cuando A y B son verdaderos. Siempre evalúa ambos operandos.
Or		A   B	verdadero cuando A o B son verdaderos. Siempre evalúa ambos operandos.
XOr	^	A ^ B	verdadero cuando A y B son diferentes



# Tabla de verdad

## operadores lógicos

A	B	And	Or	Not	Xor
true	true	true	true	false	0
true	false	false	true	false	1
false	true	false	true	true	1
false	false	false	false	true	0

# Precedencia de los operadores

Operador	Tipo	Asociatividad
++ -- + - !	Aritmético, Booleano	De derecha a izquierda
()	Paréntesis	
* / %	Aritmético	De derecha a izquierda
+ -	Aritmético	De derecha a izquierda
> >= < <= instanceof	Aritmético, De tipo	De derecha a izquierda
== !=	Booleano	De derecha a izquierda
&&	Booleano	De derecha a izquierda
	Booleano	De derecha a izquierda
()? :	Booleano	De derecha a izquierda
= += -= *= /= %=	Asignación y/o operación	De derecha a izquierda