# Java Operators

ava operators can be divided into following categories:

* Arithmetic operators
* Relation operators
* Logical operators
* Bitwise operators
* Assignment operators
* Conditional operators

## **Arithmetic operators**

Arithmetic operators are used to perform arithmetic operations like: addition, subtraction etc and helpful to solve mathematical expressions. The below table contains Arithmetic operators.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | adds two operands |
| - | subtract second operands from first |
| \* | multiply two operand |
| / | divide numerator by denumerator |
| % | remainder of division |
| ++ | Increment operator increases integer value by one |
| -- | Decrement operator decreases integer value by one |

## **Relation operators**

Relational operators are used to test comparison between operands or values. It can be use to test whether two values are equal or not equal or less than or greater than etc.

The following table shows all relation operators supported by Java.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| == | Check if two operand are equal |
| != | Check if two operand are not equal. |
| > | Check if operand on the left is greater than operand on the right |
| < | Check operand on the left is smaller than right operand |
| >= | check left operand is greater than or equal to right operand |
| <= | Check if operand on left is smaller than or equal to right operand |

## **Logical operators**

Logical Operators are used to check conditional expression. For example, we can use logical operators in if statement to evaluate conditional based expression. We can use them into loop as well to evaluate a condition.

Java supports following 3 logical operator. Suppose we have two variables whose values are: **a=true** and **b=false**.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| && | Logical AND | (a && b) is false |
| || | Logical OR | (a || b) is true |
| ! | Logical NOT | (!a) is false |

## **Bitwise operators**

Bitwise operators are used to perform operations bit by bit.

Java defines several bitwise operators that can be applied to the integer types long, int, short, char and byte.

The following table shows all bitwise operators supported by Java.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| & | Bitwise AND |
| | | Bitwise OR |
| ^ | Bitwise exclusive OR |
| << | left shift |
| >> | right shift |

Now lets see truth table for bitwise &, | and ^

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **a** | **b** | **a & b** | **a | b** | **a ^ b** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |

The bitwise **shift operators shifts the bit value**. The **left operand specifies the value to be shifted** and the **right operand specifies the number of positions** that the bits in the value are to be shifted. Both operands have the same precedence.

**Example:**

Lets create an example that shows working of bitwise operators.

a = 0001000

b = 2

a << b = 0100000

a >> b = 0000010

A=00000101

A<<2 -->00010100

A=00000101

A>>2-->00000001