Lec-1

19/01/25 9:46 AM

variable

Java divides the into the following groups:

- Arithmetic
- Assignment operators
- Comparison operators
- Logical operators
- Bitwise operators

Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example	Try it
+	Addition	Adds together two values	x + y	Try it »
-	Subtraction	Subtracts one value from another	x - y	Try it »
*	Multiplication	Multiplies two values	x * y	Try it »
/	Division	Divides one value by another	x / y	Try it »
%	Modulus	Returns the division remainder	x % y	Try it »
++	Increment	Increases the value of a variable by 1	++x	Try it »
--	Decrement	Decreases the value of a variable by 1	--x	Try it »

int x = 5

ADVERTISEMENT

Java Assignment Operators

Assignment operators are used to assign values to variables.
In the example below, we use the **assignment** operator (=) to assign the value **10** to a variable called **x**:

Example

```
intx =10;
```

Try it Yourself »

The **addition assignment** operator (+=) adds a value to a variable:

Example

```
intx =10;x +=5;
```

Try it Yourself »

A list of all assignment operators:

Operator	Example	Same As	Try it
=	x = 5	x = 5	Try it »
+=	x += 3	x = x + 3	Try it »
-=	x -= 3	x = x - 3	Try it »
*=	x *= 3	x = x * 3	Try it »
/=	x /= 3	x = x / 3	Try it »
%=	x %= 3	x = x % 3	Try it »
&=	x &= 3	x = x & 3	Try it »
|=	x |= 3	x = x | 3	Try it »
^=	x ^= 3	x = x ^ 3	Try it »
>>=	x >>= 3	x = x >> 3	Try it »
<<=	x <<= 3	x = x << 3	Try it »

x += 5 → x = x+5

i = i+1

i += i

x =+5

x += 5 → x = x+5

x == y

x != y

x >= y

x <= y

x > y

x < y

Java Comparison Operators

Comparison operators are used to compare two values:

Operator	Name	Example	Try it
==	Equal to	x == y	Try it »
!=	Not equal	x != y	Try it »
>	Greater than	x > y	Try it »
<	Less than	x < y	Try it »
>=	Greater than or equal to	x >= y	Try it »
<=	Less than or equal to	x <= y	Try it »

Java Logical Operators

Logical operators are used to determine the logic between variables or values:

Operator	Name	Description	Example	Try it
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10	Try it »
||	Logical or	Returns true if one of the statements is true	x < 5 || x < 4	Try it »
!	Logical not	Reverse the result, returns false if the result is true		

x=5 x=Ram x=7-8

{int x}

&& → logical

|| → logical OR

a	b	a||b
T	T	T
T	F	T
F	T	T
F	F	F

a	b	a&&b
T	F	F
F	T	F
F	F	F
T	T	T

! → logical not

!(T) = F

!(F) = T

P=1600
R=5
T=1

1. 30 - 50 → Bike
   - 30 to 40 → Splendor
   - 41 to 50 → Hero
2. 100 - 110 → Kurkure
   - 100 to 105 → Kurkure
   - 106 to 110 → Red chill kurkure
3. 321 to 421 → Mac
   - 321 to 389 → M1 mac
   - 390 to 421 → M2 mac
4. 500 to 721 → car
   - 500 to 600 Thar
   - 601 to 721 → Creta
5. Happy birthday!!

int num = 77

```java
public static void main(String[] args) {
        int num = 77;
        if (num >= 30 && num <= 50) {
            System.out.println("Bike");
            if(num>=30 && num<=40) {
                System.out.println("Splendor");
            }
            else {
                System.out.println("Hero");
            }
        }
        else if (num >= 100 && num <= 110) {
            System.out.println("KurKure");
        }
        else if (num >= 321 && num <= 421) {
            System.out.println("Mac");
        }
        else if (num >= 500 && num <= 721) {
            System.out.println("Car");
        }
        else {
            System.out.println("Happy Birthday!!");
        }

    }
```

x = 8 9 8 7 8

```java
public static void main(String[] args) {
        // TODO Auto-generated method stub
        int x = 8;
//      System.out.println(x++);// x = x +1;
//      System.out.println(x);
        int c = x++ + 7 + x-- - x-- - x++;
        System.out.println(c);

    }
```

8 + 7 + 9 - 8 - 7

x = 8 9 8 7 8 5 8 9

```java
public static void main(String[] args) {
        // TODO Auto-generated method stub
        int x = 8;
//      System.out.println(--x);// x = x +1;
//      System.out.println(x);
        int c = x++ + 7 + --x + x-- - x-- - --x - x++ + ++x;
        System.out.println(c);

    }
```

8 + 7 + 8 + 8 - 7 - 5 - 5 + 7

31 - 10 = 21