

# JAVA COURSE

---

OPERATOR

# OPERATOR

---

Operator ជានិមិត្តសញ្ញាដែលធ្វើប្រតិបត្តិការទៅលើ variable និង value។

ឧទាហរណ៍ + ជា operator ប្រើសម្រាប់ការបូក ហើយ \* ជា operator សម្រាប់ប្រមាណវិធីគុណ ។

តើបែងចែក operator ជា៦ប្រភេទ៖

## ១. ARITHMETIC OPERATOR

Arithmetic Operator ធ្វើប្រមាណវិធីនព្វន្ឋទៅលើ variable និង value។

ប្រមាណវិធី	អត្ថន័យ
+	បូក
-	ដក
*	គុណ
/	ចែកយកលទ្ធផល
%	ចែកយកសំណល់

## 9. ARITHMETIC OPERATOR (ଟି)

---

```
int a = 12, b = 5;
```

```
System.out.println("a + b = " + (a + b));
```

```
System.out.println("a - b = " + (a - b));
```

```
System.out.println("a * b = " + (a * b));
```

```
System.out.println("a / b = " + (a / b));
```

```
System.out.println("a % b = " + (a % b));
```

## ១. ARITHMETIC OPERATOR (ត)

---

បង្ហាញលទ្ធផល៖

$$a + b = 17$$

$$a - b = 7$$

$$a * b = 60$$

$$a / b = 2$$

$$a \% b = 2$$



## ១. ARITHMETIC OPERATOR (ត)

---

ការចែកយកលទ្ធផល ( / )

បើតួចែក និងតំណាងចែក សុទ្ធតែជាចំនួនគត់ នោះផលចែកក៏ចេញជាចំនួនគត់ដែរ។ តែបើមានមួយណាជាចំនួនទសភាគ នោះផលចែកនឹងចេញជាចំនួនទសភាគដែរ។

(9 / 2) បាន 4

(9.0 / 2) បាន 4.5

(9 / 2.0) បាន 4.5

(9.0 / 2.0) បាន 4.5

## ១. ARITHMETIC OPERATOR (ត)

---

ចែកយកសំណល់ ( % )

គេច្រើនប្រើទៅលើចំនួនគត់។

យក9ទៅចែកនឹង2 យើងបានសំណល់1

$9\%2$  បាន 1

## ២. ASSIGNMENT OPERATOR

---

Assignment Operator ធ្វើការផ្តល់តម្លៃទៅអោយ variable។

ឧ. int age;

age = 31;

= គឺជា Assignment Operator វាធ្វើការផ្តល់តម្លៃពីខាងស្តាំទៅអោយ variable ដែលនៅខាងឆ្វេង។



## ២. ASSIGNMENT OPERATOR (ត)

ប្រមាណវិធី	ឧទាហរណ៍	សមមូល
=	a = b;	a = b;
+=	a += b;	a = a + b;
-=	a -= b;	a = a - b;
*=	a *= b;	a = a * b;
/=	a /= b;	a = a / b;

## ໒. ASSIGNMENT OPERATOR (ຮູບ)

---

```
int a = 4; int result;
```

```
result = a;
```

```
System.out.println("result using =: " + result);
```

```
result += a;
```

```
System.out.println("result using +=: " + result);
```

```
result *= a;
```

```
System.out.println("result using *=: " + result);
```

## ២. ASSIGNMENT OPERATOR (ត)

---

បង្ហាញលទ្ធផល

result using `=`: 4

result using `+=`: 8

result using `*=`: 32

## ៣. RELATIONAL OPERATOR

---

Relational Operators ប្រើដើម្បីត្រួតពិនិត្យទៅលើ operand ពីរ។

ឧ.  $a < b$ ;

$<$  គឺជា Relational Operator ពិនិត្យថាតើ  $a$  តូចជាង  $b$  ឬទេ។

វាផ្តល់តម្លៃត្រលប់មកវិញ អាចជា true ឬ false។

### ៣. RELATIONAL OPERATOR (ត)

ប្រមាណវិធី	អត្ថន័យ	ឧទាហរណ៍
==	ស្មើ	3 == 5 លទ្ធផល false
!=	មិនស្មើ	3 != 5 លទ្ធផល true
>	ធំជាង	3 > 5 លទ្ធផល false
<	តូចជាង	3 < 5 លទ្ធផល true
>=	ធំជាងឬស្មើ	3 >= 5 លទ្ធផល false
<=	តូចជាងឬស្មើ	3 <= 5 លទ្ធផល false



## ៤. LOGICAL OPERATOR

Logical Operator ប្រើសម្រាប់ពិនិត្យមើល expression ថាតើ ពិតឬមិនពិត។  
លទ្ធផលរបស់វា គេយកដើម្បីធ្វើ decision making។

ប្រមាណវិធី	ឧទាហរណ៍	អត្ថន័យ
&& (Logical AND)	exp1 && exp2	true ប្រសិនបើ exp1 និង exp2 ពិត
(Logical OR)	expr1    exp2	true បើ exp1 ឬ exp2 ពិត
! (Logical NOT)	!expression	true បើ expression មិនពិត

## 4. LOGICAL OPERATOR (६)

---

```
System.out.println((5 > 3) && (8 > 5)); // true
```

```
System.out.println((5 > 3) && (8 < 5)); // false
```

```
System.out.println((5 < 3) || (8 > 5)); // true
```

```
System.out.println((5 > 3) || (8 < 5)); // true
```

```
System.out.println((5 < 3) || (8 < 5)); // false
```

```
System.out.println(!(5 == 3)); // true
```

```
System.out.println(!(5 > 3)); // false
```

## ៥. UNARY OPERATOR

---

Unary Operators ត្រូវបានប្រើចំពោះ operand តែមួយ។

ឧ. ++ ជា Unary Operator ដែលតម្លើងតម្លៃលេខម្តងមួយៗ។

មានន័យថា ++4 យើងបាន 5។

## ៥. UNARY OPERATOR (ត)

ប្រភេទវិធី	អត្ថន័យ
+	Unary plus: មិនចាំបាច់ប្រើទេ ព្រោះលេខវិជ្ជមានស្រាប់
-	Unary minus: ត្រឡប់សញ្ញារបស់ express
++	Increment operator: កើនម្តង១
--	Decrement operator: ថយម្តង១
!	Logical complement operator: ត្រឡប់សញ្ញា boolean

## ๕. UNARY OPERATOR (๓)

---

```
int a = 12, b = 12; int result1, result2;
```

```
System.out.println("Value of a: " + a);
```

```
result1 = ++a;
```

```
System.out.println("After increment: " + result1);
```

```
System.out.println("Value of b: " + b);
```

```
result2 = --b;
```

```
System.out.println("After decrement: " + result2);
```



## ៥. UNARY OPERATOR (ត)

---

បង្ហាញ

Value of a: 12

After increment: 13

Value of b: 12

After decrement: 11

## ៦. BITWISE OPERATOR

---

Bitwise Operator ប្រើសម្រាប់ធ្វើប្រតិបត្តិការលើ Individual bits។

ខ. Bitwise complement Operation of 35

35 = 00100011 (In Binary)

~ 00100011

---

11011100 = 220 (In decimal)

~ គឺជា Bitwise Operator។ វាត្រូវបានប្រើប្រាស់លើ bit នីមួយៗ (0 ទៅ 1 ហើយ 1 ទៅ 0)

## ៦. BITWISE OPERATOR (ត)

ប្រមាណវិធី	អត្ថន័យ
$\sim$	Bitwise Complement
$\ll$	Left Shift
$\gg$	Right Shift
$\ggg$	Unsigned Right Shift
$\&$	Bitwise AND
$\wedge$	Bitwise exclusive OR

# INSTANCEOF OPERATOR (OTHER)

---

instanceof ពិនិត្យមើលថាតើ object មួយជារបស់ Class ជាក់លាក់ណាមួយឬទេ។

```
១. String str = "Piseth Java School";
```

```
boolean result = str instanceof String;
```

```
System.out.println("Is str an object of String? " + result);
```

បង្ហាញ

```
Is str an object of String? true
```

# TERNARY OPERATOR (OTHER)

---

ternary operator (conditional operator) គឺជាទំរង់កាត់នៃ if-then-else statement

ឧ. `variable = Expression ? expression1 : expression2`

បើ Expression ពិត នោះ variable ចាប់យកតម្លៃ expression1

បើ Expression មិនពិត នោះ variable ចាប់យកតម្លៃ expression2



# TERNARY OPERATOR (ꣳ)

---

```
int februaryDays = 29;
```

```
String result;
```

```
result = (februaryDays == 28) ? "Not a leap year" : "Leap year";
```

```
System.out.println(result); // Leap year
```