Reading: Operators

Bookmark this page

Operators are special symbols that allow us to carry out operations. We can perform comparative, arithmetic or logical operations. The value that the operator operates on is called the operand.
As an example, let's take the arithmetic operation:

```
10+20
```

```
30
```

Here 10 and 20 are operands and '+' is the operator. 30 is the output of the arithmetic operation.

Let's look at the operators we have in Python in detail.

1. **Arithmetic Operators:** Arithmetic operators are used for performing mathematical operations like addition, subtraction, multiplication, division etc. The table below lists all the possible arithmetic operators we can possibly have to perform an arithmetic operation.

Operator	Meaning	Example
+	Add two or more operands with the + operators	3+5 will be 8
-	Subtract the second operand from the first	5-3 will be 2
*	Multiply two or more operands	6*2 will be 12
/	Divide the first operand by the second one. The result is always a float	4/2 will be 2.0
%	Modulus operation. The remainder of the division of the first operand by the second one	5%3 will be 2
//	Floor division - Result is rounded off to the whole number equal to or immediately before the resultant float	7//2 will be 3
**	The first operand raised to the power of the second operand	4**3 will be 64

Let's take a look at examples that demonstrate how each of the arithmetic operations is performed.

```
x = 20
y = 10

print('x + y =',x+y)
```

```
x + y = 30
```

```
print('x - y =',x-y)
```

```
x - y = 10
```

```
print('x * y =',x*y)
```

```
x * y = 200
```

```
print('x / y =',x/y)
```

```
x / y = 2.0
```

```
print('x // y =',x//y)
```

```
x // y = 2
```

```
print('x ** y =',x**y)
```

```
x ** y = 10240000000000
```

2. **Comparison Operators:** Comparison Operators are used for comparing values. It returns either **True** or **False** according to the condition. The table below lists all the possible comparison operators we can possibly have to perform a comparison operation.

Operator	Meaning	Example

>	Checks if the left operand is greater than the right operand and returns true or false	5>3 is True 3>8 is False
<	Checks if the left operand is lesser than the right operand and returns true or false	5<3 is False 3<8 is True
==	Checks if the two operands are equal and returns true or false	3 == 3 is True 3 == 5 is False
>=	Checks if the left operand is greater than or equal to the right operand and returns true or false	4 >= 2 will be True 4 >= 4 will be True 4 >= 3 will be False

<=	Checks if the left operand is lesser than or equal to the right operand and returns true or false	2 <= 4 will be True 2 <= 2 will be True 2 <= 3 will be False
!=	Checks if the left operand is not equal to the right operand and returns true or false	3 != 2 will be True 3 != 3 will be False

Let's take a look at a few examples of the same.

```
x = 10
y = 12

print('x > y is',x>y)
```

```
x > y is False
```

```
print('x < y is',x<y)
```

```
x < y is True
```

```
print('x == y is',x==y)
```

```
x == y is False
```

```
print('x != y is',x!=y)
```

```
x != y is True
```

```
print('x >= y is',x>=y)
```

```
x >= y is False
```

```
print('x <= y is',x<=y)
```

```
x <= y is True
```

3. **Logical Operators:** Logical Operators are used in combination with comparison operations to check for a combination of conditions. It returns either **True** or **False** according to the combination. The table below lists all the possible logical operators we can possibly have to perform a logical operation.

Operator	Meaning	Example
and	True only if both operands are true. The operands for this is usually a comparison operation which returns a boolean.	5<3 or 3<8 is False, as only the second operand is an operation that returns true 5>3 or 3<8 is True, as both operations in both operands return True
or	True if one of the two operands is true. The operands for this is usually a comparison operation which returns a boolean.	5<3 or 3<8 is True, as the second operand is an operation that returns true
not	True if the operand is false and False if the operand is true	3 == 3 is True 3 == 5 is False

Let's take a look at the following example to understand the logical operators.

```
x = True
y = False

print('x and y is',x and y)
```

```
x and y is False
```

```
print('x or y is',x or y)
```

```
x or y is True
```

```
print('not x is',not x)
```

```
not x is False
```

The comparison and logical operators are used for conditional branching and loops which we will take a look at, in the lessons to come.

- About Us
- Enterprise
- Blog
- Press

Support

- Contact us
- Terms of Service
- Privacy Policy