

Python Operators

[< Previous](#)[Next >](#)

Python Operators

Operators are used to perform operations on variables and values.

Python divides the operators in the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

Python Arithmetic Operators

Arithmetic operators are used with numeric values to perform common mathematical operations:

Operator	Name	Example	Try it
+	Addition	x + y	Try it »
-	Subtraction	x - y	Try it »
*	Multiplication	x * y	Try it »
/	Division	x / y	Try it »
%	Modulus	x % y	Try it »
**	Exponentiation	x ** y	
//	Floor division	x // y	

Python Assignment Operators

Assignment operators are used to assign values to variables:

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 >>= 3	x = x >> 3	Try it »
<<=	x <<= 3	x = x << 3	Try it »

Python Comparison Operators

Comparison operators are used to compare two values:

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

<	Less than	<code>x < y</code>	Try it »
>=	Greater than or equal to	<code>x >= y</code>	Try it »
<=	Less than or equal to	<code>x <= y</code>	Try it »

Python Logical Operators

Logical operators are used to combine conditional statements:

Operator	Description	Example	Try it
and	Returns True if both statements are true	<code>x < 5 and x < 10</code>	Try it »
or	Returns True if one of the statements is true	<code>x < 5 or x < 4</code>	Try it »
not	Reverse the result, returns False if the result is true	<code>not(x < 5 and x < 10)</code>	Try it »

Python Identity Operators

Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location:

Operator	Description	Example	Try it
is	Returns true if both variables are the same object	<code>x is y</code>	Try it »
is not	Returns true if both variables are not the same object	<code>x is not y</code>	Try it »

Python Membership Operators

Membership operators are used to test if a sequence is presented in an object:

Operator	Description	Example	Try it
----------	-------------	---------	--------

in	Returns True if a sequence with the specified value is present in the object	x in y	Try it »
not in	Returns True if a sequence with the specified value is not present in the object	x not in y	Try it »

Python Bitwise Operators

Bitwise operators are used to compare (binary) numbers:

Operator	Name	Description
&	AND	Sets each bit to 1 if both bits are 1
	OR	Sets each bit to 1 if one of two bits is 1
^	XOR	Sets each bit to 1 if only one of two bits is 1
~	NOT	Inverts all the bits
<<	Zero fill left shift	Shift left by pushing zeros in from the right and let the leftmost bits fall off
>>	Signed right shift	Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off

Test Yourself With Exercises

Exercise:

Multiply **10** with **5** , and print the result.

```
print(10 * 5)
```

[Submit Answer »](#)

[Start the Exercise](#)

[◀ Previous](#)

[Next ▶](#)

[Copyright 1999-2018](#) by Refsnes Data. All Rights Reserved.