

Python Operators

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

Python Arithmetic Operators

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor Division	$x // y$

Python Assignment Operators

Operator

Example

Same As

=

`x = 4`

`x = 4`

`+=`

`x += 2`

`x = x + 2`

`-=`

`x -= 1`

`x = x - 1`

`*=`

`x *= 2`

`x = x * 2`

`/=`

`x /= 2`

`x = x / 2`

`%=`

`x %= 2`

`x = x % 2`

`//=`

`x //= 2`

`x = x // 2`

Python Comparison Operators

Operator

Name

Example

==

Equal

`x == y`

!=

Not equal

`x != y`

>

Greater than

`x > y`

<

Less than

`x < y`

>=

Greater than or equal

`x >= y`

<=

Less than or equal

`x <= y`

• Python Logical Operators

Operator

Description

Example

and

Returns True if both statements are true

`x == y and x < z`

or

Returns True if one of the statements is true

`x == y or x < z`

not

Reverse the result, returns False if the result is true

`not(x == y and x < z)`

Python Identity Operators

Operator

is

is not

Description

Returns True if both variables are the same object

Returns True if both variables are not the same object

Example

x is y

x is not y

Python Membership Operators

Operator

in

Returns True if a sequence with the specified value is present in the object

Example

x in y

not in

Returns True if a sequence with the specified value is not present in the object

x not in y

Python Bitwise Operators

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 left by pushing zeros in from the right and let the leftmost bits fall off