Python Operators: Arithmetic, Logical, Comparison, Assignment, Bitwise & Precedence

By Steve Campbell Updated May 3, 2022

What are Logical Operators in Python?

Logical Operators in Python are used to perform logical operations on the values of variables. The value is either true or false. We can figure out the conditions by the result of the truth values. There are mainly three types of logical operators in python : logical AND, logical OR and logical NOT. Operators are represented by keywords or special characters.

In this tutorial, we going to learn various operators

Arithmetic Operators

Arithmetic Operators perform various arithmetic calculations like addition, subtraction, multiplication, division, %modulus, exponent, etc. There are various methods for arithmetic calculation in Python like you can use the eval function, declare variable & calculate, or call functions.

Example: For arithmetic operators we will take simple example of addition where we will add two-digit 4+5=9

This code is editable. Click Run to Execute

```
x= 4
y= 5
print(x + y)
```

Run

Similarly, you can use other arithmetic operators like for multiplication(*), division (/), substraction (-), etc.

Comparison Operators

Comparison Operators In Python compares the values on either side of the operand and determines the relation between them. It is also referred to as relational operators. Various comparison operators in python are (==, != , <>, >,<=, etc.)

Example: For comparison operators we will compare the value of x to the value of y and print the result in true or false. Here in example, our value of x = 4 which is smaller than y = 5, so when we print the value as x>y, it actually compares the value of x to y and since it is not correct, it returns false.

This code is editable. Click Run to Execute

```
x = 4
y = 5
print(('x > y  is',x>y))
```

Run

Likewise, you can try other comparison operators (x < y, x==y, x!=y, etc.)

Python Assignment Operators

Assignment Operators in **Python** are used for assigning the value of the right operand to the left operand. Various assignment operators used in Python are (+=, – = , *=, /= , etc.).

Example: Python assignment operators is simply to assign the value, for example

This code is editable. Click Run to Execute

```
num1 = 4
num2 = 5
print(("Line 1 - Value of num1 : ", num1))
print(("Line 2 - Value of num2 : ", num2))
```

Run

Example of compound assignment operator

We can also use a compound assignment operator, where you can add, subtract, multiply right operand to left and assign addition (or any other arithmetic function) to the left operand.

- Step 1: Assign value to num1 and num2
- Step 2: Add value of num1 and num2 (4+5=9)
- Step 3: To this result add num1 to the output of Step 2 (9+4)
- Step 4: It will print the final result as 13

This code is editable. Click Run to Execute

```
num1 = 4
num2 = 5
res = num1 + num2
res += num1
print(("Line 1 - Result of + is ", res))
```

Run

Logical Operators or Bitwise Operators

Logical operators in Python are used for conditional statements are true or false. Logical operators in Python are AND, OR and NOT. For logical operators following condition are applied.

- For AND operator – It returns TRUE if both the operands (right side and left side) are true
- For OR operator- It returns TRUE if either of the operand (right side or left side) is true
- For NOT operator- returns TRUE if operand is false

Example: Here in example we get true or false based on the value of a and b

This code is editable. Click Run to Execute

```
a = True
b = False
print(('a and b is',a and b))
print(('a or b is',a or b))
print(('not a is',not a))
```

Run

Membership Operators

These operators test for membership in a sequence such as lists, strings or tuples. There are two membership operators that are used in Python. (in, not in). It gives the result based on the variable present in specified sequence or string

Example: For example here we check whether the value of x=4 and value of y=8 is available in list or not, by using **in** and **not in** operators.

This code is editable. Click Run to Execute

```
x = 4
y = 8
list = [1, 2, 3, 4, 5 ];
if ( x in list ):
   print("Line 1 - x is available in the given list")
else:
   print("Line 1 - x is not available in the given list")
if ( y not in list ):
   print("Line 2 - y is not available in the given list")
else:
   print("Line 2 - y is available in the given list")
```

Run

- Declare the value for x and y
- Declare the value of list
- Use the “in” operator in code with if statement to check the value of x existing in the list and print the result accordingly
- Use the “not in” operator in code with if statement to check the value of y exist in the list and print the result accordingly
- Run the code- When the code run it gives the desired output

Identity Operators

Identity Operators in Python are used to compare the memory location of two objects. The two identity operators used in Python are (is, is not).

- Operator is: It returns true if two variables point the same object and false otherwise
- Operator is not: It returns false if two variables point the same object and true otherwise

Following operands are in decreasing order of precedence.

Operators in the same box evaluate left to right

Operators (Decreasing order of precedence)	Meaning
**	Exponent
*, /, //, %	Multiplication, Division, Floor division, Modulus
+, –	Addition, Subtraction
<= < > >=	Comparison operators
= %= /= //= -= += *= **=	Assignment Operators
is is not	Identity operators
in not in	Membership operators
not or and	Logical operators

Example:

This code is editable. Click Run to Execute

```
x = 20
y = 20
if ( x is y ):
    print("x & y  SAME identity")
y=30
if ( x is not y ):
    print("x & y have DIFFERENT identity")
```

Run

- Declare the value for variable x and y
- Use the operator “is” in code to check if value of x is same as y
- Next we use the operator “is not” in code if value of x is not same as y
- Run the code- The output of the result is as expected

Operator precedence

The operator precedence determines which operators need to be evaluated first. To avoid ambiguity in values, precedence operators are necessary. Just like in normal multiplication method, multiplication has a higher precedence than addition. For example in 3+ 4*5, the answer is 23, to change the order of precedence we use a parentheses (3+4)*5, now the answer is 35. Precedence operator used in Python are (unary + – ~, **, * / %, + – , &) etc.

This code is editable. Click Run to Execute

```
v = 4
w = 5
x = 8
y = 2
z = 0
z = (v+w) * x / y;
print("Value of (v+w) * x/ y is ",  z)
```

Run

- Declare the value of variable v,w…z
- Now apply the formula and run the code
- The code will execute and calculate the variable with higher precedence and will give the output

Python 2 Example

Above examples are Python 3 codes, if you want to use Python 2, please consider following codes

This code is editable. Click Run to Execute

```
#Arithmetic Operators
x= 4
y= 5
print x + y

#Comparison Operators
x = 4
y = 5
print('x > y  is',x>y)

#Assignment Operators
num1 = 4
num2 = 5
print ("Line 1 - Value of num1 : ", num1)
print ("Line 2 - Value of num2 : ", num2)

#compound assignment operator
num1 = 4
num2 = 5
res = num1 + num2
res += num1
print ("Line 1 - Result of + is ", res)

#Logical Operators
a = True
b = False
print('a and b is',a and b)
print('a or b is',a or b)
print('not a is',not a)

#Membership Operators
x = 4
y = 8
```

```
list = [1, 2, 3, 4, 5 ];
if ( x in list ):
   print "Line 1 - x is available in the given list"
else:
   print "Line 1 - x is not available in the given list"
if ( y not in list ):
   print "Line 2 - y is not available in the given list"
else:
   print "Line 2 - y is available in the given list"

#Identity Operators
x = 20
y = 20
if ( x is y ):
    print "x & y  SAME identity"
y=30
if ( x is not y ):
    print "x & y have DIFFERENT identity"

#Operator precedence
v = 4
w = 5
x = 8
y = 2
z = 0
z = (v+w) * x / y;
print "Value of (v+w) * x/ y is ",  z
```

Run

Summary:

Operators in a programming language are used to perform various operations on values and variables. In Python, you can use operators like

- There are various methods for arithmetic calculation in Python as you can use the eval function, declare variable & calculate, or call functions
- Comparison operators often referred as relational operators are used to compare the values on either side of them and determine the relation between them
- Python assignment operators are simply to assign the value to variable
- Python also allows you to use a compound assignment operator, in a complicated arithmetic calculation, where you can assign the result of one operand to the other
- For AND operator – It returns TRUE if both the operands (right side and left side) are true
- For OR operator- It returns TRUE if either of the operand (right side or left side) is true
- For NOT operator- returns TRUE if operand is false
- There are two membership operators that are used in Python. (in, not in).
- It gives the result based on the variable present in specified sequence or string
- The two identify operators used in Python are (is, is not)
- It returns true if two variables point the same object and false otherwise
- Precedence operator can be useful when you have to set priority for which calculation need to be done first in a complex calculation.