Python Operators

Python Operators

There are seven kinds of operators in Python. They are

- Arithmetic Operators
- Bitwise Operators
- Assignment Operators
- Comparison Operators / Relational Operators
- Identity Operators
- Membership Operators

In this tutorial, we will go through each of these operators with examples.

Arithmetic Operators

Arithmetic Operators are used to perform basic mathematical arithmetic operators like addition, subtraction, multiplication, etc. The following table lists out all the arithmetic operators in Python.

Operator Symbol	Description	Example
+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Modulus	x % y
**	Exponentiation	x ** y
//	Floor division	x // y

Python Arithmetic Operators

In the following program, we will take values for variables x and y, and perform arithmetic operations on these values using Python Arithmetic Operators.

Python Program

```
x = 5
y = 2

addition = x + y
subtraction = x - y
multiplication = x * y
division = x / y
modulus = x % y
exponentiation = x ** y
floor_division = x // y

print(f'x + y = {addition}')
print(f'x - y = {subtraction}')
print(f'x * y = {multiplication}')
print(f'x / y = {division}')
print(f'x % y = {modulus}')
print(f'x ** y = {exponentiation}')
print(f'x // y = {floor_division}')
```

Output

```
x + y = 7
x - y = 3
x * y = 10
x / y = 2.5
x % y = 1
x ** y = 25
x // y = 2
```

Bitwise Operators

Bitwise Operators are used to perform bit level operations. The following table lists out all the bitwise operators in Python.

Operator Symbol	Description	Example
&	AND	x & y
|	OR	x | y
^	XOR	x ^ y
~	NOT	~x
<<	Zero fill left shift	x << y
>>	Signed right shift	x >> y

Python Bitwise Operators

Python Program

```
# AND
x, y = 5, 2
print(x & y) # 0

# OR
x, y = 5, 2
print(x | y) # 7

# XOR
x, y = 5, 2
print(x ^ y) # 7

# NOT
x, y = 5, 2
print(~x) # -6

# Zero fill left shift
x, y = 5, 2
print(x << y) # 20

#Signed right shift
x, y = 5, 2
print(x >> y) # 1
```

Assignment Operators

Assignment Operators are used to assign or store a specific value in a variable. The following table lists out all the assignment operators in Python.

Operator Symbol	Description	Example	Equivalent to
=	Assignment	x = y	
+=	Addition Assignment	x += y	x = x + y
-=	Subtraction Assignment	x -= y	x = x – y
*=	Multiplication Assignment	x *= y	x = x * y
/=	Division Assignment	x /= y	x = x / y
%=	Modulus Assignment	x %= y	x = x % y
**=	Exponentiation Assignment	x **= y	x = x ** y
//=	Floor-division Assignment	x //= y	x = x // y
&=	AND Assignment	x &= y	x = x & y
|=	OR Assignment	x |= y	x = x | y
^=	XOR Assignment	x ^= y	x = x ^ y
<<=	Zero fill left shift Assignment	x <<= y	x = x << y
>>=	Signed right shift Assignment	x >>= y	x = x >> y

In the following program, we will take values for variables x and y, and perform assignment operations on these values using Python Assignment Operators.

Python Program

```
x, y = 5, 2
x += y
print(x) # 7

x, y = 5, 2
x -= y
print(x) # 3

x, y = 5, 2
x *= y
print(x) # 10

x, y = 5, 2
x /= y
print(x) # 2.5

x, y = 5, 2
x %= y
print(x) # 1

x, y = 5, 2
x **= y
print(x) # 25

x, y = 5, 2
x //= y
print(x) # 2

x, y = 5, 2
x &= y
print(x) # 0

x, y = 5, 2
x |= y
print(x) # 7

x, y = 5, 2
x ^= y
print(x) # 7

x, y = 5, 2
x <<= y
print(x) # 20

x, y = 5, 2
x >>= y
print(x) # 1
```

Comparison Operators

Comparison Operators are used to compare two operands. The following table lists out all the Comparison operators in Python.

Operator Symbol	Description	Example
==	Equal to	x == y
!=	Not Equal to	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

Python Comparison Operators

Python Program

```
# Equal to
x, y = 5, 2
print(x == y) # False

# Not equal to
x, y = 5, 2
print(x != y) # True

# Greater than
x, y = 5, 2
print(x > y) # True

# Less than
x, y = 5, 2
print(x < y) # False

# Greater than or equal to
x, y = 5, 2
print(x >= y) # True

# Less than or equal to
x, y = 5, 2
print(x <= y) # False
```

Logical Operators

Logical Operators are used to combine simple conditions and form compound conditions. The following table lists out all the Logical operators in Python.

Operator Symbol	Description	Example
and	Returns True if both operands are True.	x and y
or	Returns True if any of the operands is True.	x or y
not	Returns the complement of given boolean operand.	not x

Python Logical Operators

Python Program

```
# Logical AND
x, y = True, False
print(x and y) # False

# Logical OR
x, y = True, False
print(x or y) # True

# Logical NOT
x = True
print(not x) # False
```

Identity Operators

Identity Operators are used to check if two variables point to same reference of an object in Python. The following table lists out the two Identity operators in Python.

Operator Symbol	Description	Example
is	Returns True if both operands refer to same object.	x is y
is not	Returns True if two operands refer to different objects.	x is not y

Python Logical Operators

Two objects are said to have same reference, if they have same id. In the following program, we shall print the ids of x and y, along with the results of is and is not operators.

Python Program – x and y with same reference

```
x = [1, 2, 3]
y = x
print(x is y) # True
print(x is not y) # False
print(id(x))
print(id(y))
```

Output

```
True
False
2284566373000
2284566373000
```

We have assigned the value of x to y. Now, both x and y store the reference to same object in memory. Therefore, x is y.

Python Program – x and y with different reference

```
# is operator
x = [1, 2, 3]
y = [1, 2, 3]
print(x is y) # True
print(x is not y) # False
print(id(x))
print(id(y))
```

Output

```
False
True
1961841222280
1961841222344
```

Even though the list elements are same, x and y are assigned with two different list objects. Hence, the ids are different for x and y and therefore x is not y, in this case.

Membership Operators

Membership Operators are used to check if an element or item is present in the given collection or sequence. The following table lists out the two Membership operators in Python.

Operator Symbol	Description	Example
in	Returns True if element (x) is present in sequence (y).	x in y
not in	Returns True if element (x) is not present in sequence (y).	x not in y

Python Logical Operators

Python Program

```python
# in operator
x = 1
y = [1, 2, 3]
print(x in y) # True

# not in operator
x = 8
y = [1, 2, 3]
print(x not in y) # True
```

Conclusion

In this Python Tutorial, we learned about different kinds of Operators in Python: Arithmetic, Bitwise, Assignment, Comparison, Logical, Identity and Membership.

Python Programming

- Python Tutorial
- Install Python
- Install Anaconda Python
- Python HelloWorld Program
- Python Variables
- Python Datatype Conversion
- Python Comments
- Python If
- Python If Else
- Python While Loop
- Python For Loop
- Python Operators
- Python Functions
- Python Lambda Functions
- Python Builtin Functions

Python Collections

- Python Strings
- Python Lists
- Python Tuples
- Python Dictionary
- Python Sets

Libraries

- Python Numpy Tutorial
- Python SciPy Tutorial
- Python Pandas Tutorial
- Python Matplotlib Tutorial

Advanced Topics

- Python Multithreading

Useful Resources

- Python Interview Questions