Lecture-05: Python Operators

Basic Operators:

Operators are the constructs which can manipulate the value of operands.

Consider the expression 4 + 5 = 9. Here, 4 and 5 are called operands and + is called operator.

Types of Operator:

Python language supports the following types of operators.

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Python Arithmetic Operators:

Assume variable a holds 10 and variable b holds 20, then –

Operator	Description	Example
+ Addition	Adds values on either side of the operator.	a + b = 30
- Subtraction	Subtracts right hand operand from left hand operand.	a – b = -10
* Multiplication	Multiplies values on either side of the operator	a * b = 200
/ Division	Divides left hand operand by right hand operand	b / a = 2
% Modulus	Divides left hand operand by right hand operand and returns remainder	b % a = 0

** Exponent	Performs exponential (power) calculation on operators	a**b =10 to the power 20
//Floor Division	Floor Division - The division of operands where the result is the quotient in which the digits after the decimal point are removed. But if one of the operands is negative, the result is floored, i.e., rounded away from zero (towards negative infinity) −	9//2 = 4 and 9.0//2.0 = 4.0, -11//3 = -4, -11.0//3 = -4.0

Example:

Assume variable a holds 21 and variable b holds 10, then –

```
a = 21
b = 10
c = 0
 c = a + b
print "Line 1 - Value of c is ", c
 c = a – b
print "Line 2 - Value of c is ", c
c = a * b
print "Line 3 - Value of c is ", c
```

```
c = a / b
print "Line 4 - Value of c is ", c
  c = a % b
print "Line 5 - Value of c is ", c
a = 2
b = 3
c = a**b
 print "Line 6 - Value of c is ", c
 a = 10
b = 5
c = a//b
 print "Line 7 - Value of c is ", c
```

When you execute the above program, it produces the following result –

Line 1 - Value of c is 31

Line 2 - Value of c is 11

Line 3 - Value of c is 210

Line 4 - Value of c is 2

Line 5 - Value of c is 1

Line 6 - Value of c is 8

Line 7 - Value of c is 2

Python Comparison Operators:

These operators compare the values on either sides of them and decide the relation among them. They are also called Relational operators.

Assume variable a holds 10 and variable b holds 20, then –

Operator	Description	Example
==	If the values of two operands are equal, then the condition becomes true.	(a == b) is not true.
!=	If values of two operands are not equal, then condition becomes true.	(a != b) is true.
<>	If values of two operands are not equal, then condition becomes true.	(a <> b) is true. This is similar to != operator.
>	If the value of left operand is greater than the value of right operand, then condition becomes true.	(a > b) is not true.

<	If the value of left operand is less than the value of right operand, then condition becomes true.	(a < b) is true.
>=	If the value of left operand is greater than or equal to the value of right operand, then condition becomes true.	(a >= b) is not true.
<=	If the value of left operand is less than or equal to the value of right operand, then condition becomes true.	(a <= b) is true.

Example:

Assume variable a holds 10 and variable b holds 20, then –

```
a = 21
b = 10
c = 0
 if ( a == b ):
    print "Line 1 - a is equal to b“
else:
 print "Line 1 - a is not equal to b"
if ( a != b ):
   print "Line 2 - a is not equal to b“
else:
      print "Line 2 - a is equal to b"
```

```
if ( a <> b ):
   print "Line 3 - a is not equal to b“
else:
   print "Line 3 - a is equal to b“
 if ( a < b ):
    print "Line 4 - a is less than b“
 else:
   print "Line 4 - a is not less than b"
if ( a > b ):
   print "Line 5 - a is greater than b“
else:
   print "Line 5 - a is not greater than b"
```

```
a = 5;
b = 20;
if ( a <= b ):
    print "Line 6 - a is either less than or equal to  b“
else:
print "Line 6 - a is neither less than nor equal to  b"
if ( b >= a ):
print "Line 7 - b is either greater than  or equal to b“
else:
 print "Line 7 - b is neither greater than  nor equal to b"
```

When you execute the above program it produces the following result –

Line 1 - a is not equal to b

Line 2 - a is not equal to b

Line 3 - a is not equal to b

Line 4 - a is not less than b

Line 5 - a is greater than b

Line 6 - a is either less than or equal to b

Line 7 - b is either greater than or equal to b

Python Assignment Operators:

Assume variable a holds 10 and variable b holds 20, then –

Operator	Description	Example
=	Assigns values from right side operands to left side operand	c = a + b assigns value of a + b into c
+= Add AND	It adds right operand to the left operand and assign the result to left operand	c += a is equivalent to c = c + a
-= Subtract AND	It subtracts right operand from the left operand and assign the result to left operand	c -= a is equivalent to c = c - a
*= Multiply AND	It multiplies right operand with the left operand and assign the result to left operand	c *= a is equivalent to c = c * a

/= Divide AND	It divides left operand with the right operand and assign the result to left operand	c /= a is equivalent to c = c / a
%= Modulus AND	It takes modulus using two operands and assign the result to left operand	c %= a is equivalent to c = c % a
**= Exponent AND	Performs exponential (power) calculation on operators and assign value to the left operand	c **= a is equivalent to c = c ** a
//= Floor Division	It performs floor division on operators and assign value to the left operand	c //= a is equivalent to c = c // a

Example:

```
a = 21
b = 10
c = 0
c = a + b
print "Line 1 - Value of c is ", c
 c += a
print "Line 2 - Value of c is ", c
 c *= a
print "Line 3 - Value of c is ", c
c /= a
 print "Line 4 - Value of c is ", c
```

```
c  = 2
c %= a
print "Line 5 - Value of c is ", c
 c **= a
print "Line 6 - Value of c is ", c
 c //= a
print "Line 7 - Value of c is ", c
```

When you execute the above program, it produces the following result –

Line 1 - Value of c is 31

Line 2 - Value of c is 52

Line 3 - Value of c is 1092

Line 4 - Value of c is 52

Line 5 - Value of c is 2

Line 6 - Value of c is 2097152

Line 7 - Value of c is 99864

Python Bitwise Operators:

Bitwise operator works on bits and performs bit by bit operation. Assume if a = 60; and b = 13; Now in the binary format their values will be 0011 1100 and 0000 1101 respectively. Following table lists out the bitwise operators supported by Python language with an example each in those, we use the above two variables (a and b) as operands −

a = 0011 1100

b = 0000 1101

\-----------------

a&b = 0000 1100

a|b = 0011 1101

a^b = 0011 0001

~a = 1100 0011

There are following Bitwise operators supported by Python language.

Operator	Description	Example
& Binary AND	Operator copies a bit to the result if it exists in both operands	(a & b) (means 0000 1100)
| Binary OR	It copies a bit if it exists in either operand.	(a | b) = 61 (means 0011 1101)
^ Binary XOR	It copies the bit if it is set in one operand but not both.	(a ^ b) = 49 (means 0011 0001)
~ Binary Ones Complement	It is unary and has the effect of 'flipping' bits.	(~a) = -61 (means 1100 0011 in 2's complement form due to a signed binary number.
<< Binary Left Shift	The left operands value is moved left by the number of bits specified by the right operand.	a << 2 = 240 (means 1111 0000)
>> Binary Right Shift	The left operands value is moved right by the number of bits specified by the right operand.	a >> 2 = 15 (means 0000 1111)

Example:

```
a = 60          # 60 = 0011 1100
b = 13          # 13 = 0000 1101
 c = 0
c = a & b;      # 12 = 0000 1100
print "Line 1 - Value of c is ", c
 c = a | b;     # 61 = 0011 1101
print "Line 2 - Value of c is ", c
c = a ^ b;      # 49 = 0011 0001
print "Line 3 - Value of c is ", c
 c = ~a;        # -61 = 1100 0011
print "Line 4 - Value of c is ", c
```

```
a = 60              # 60 = 0011 1100
 c = a << 2;        # 240 = 1111 0000
print "Line 5 - Value of c is ", c
 a = 60             # 60 = 0011 1100
 c = a >> 2;        # 15 = 0000 1111
print "Line 6 - Value of c is ", c
```

When you execute the above program it produces the following result –

Line 1 - Value of c is 12

Line 2 - Value of c is 61

Line 3 - Value of c is 49

Line 4 - Value of c is -61

Line 5 - Value of c is 240

Line 6 - Value of c is 15

References:

1. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher

2. T. Budd, Exploring Python, TMH, 1st Ed, 2011 3. Fundamentals of Python first Programmes by Kenneth A Lambert, Copyrighted material Course Technology Inc. 1 st edition (6th February 2009)

3. https://www.tutorialspoint.com/python/index.htm

4. https://www.geeksforgeeks.org/python-programming-language

****END OF THE LECTURE***

THANK YOU