Lecture-06: Python Operators (continue…)

Python Membership Operators:

Python’s membership operators test for membership in a sequence, such as strings, lists, or tuples. There are two membership operators as explained below –

Operator	Description	Example
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise.	x in y, here in results in a 1 if x is a member of sequence y.
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.	x not in y, here not in results in a 1 if x is not a member of sequence y.

Example:

```
a = 10
b = 20
list = [1, 2, 3, 4, 5 ];
if ( a in list ):
  print "Line 1 - a is available in the given list“
else:
  print "Line 1 - a is not available in the given list“
 if ( b not in list ):
 print "Line 2 - b is not available in the given list“
else:
  print "Line 2 - b is available in the given list“
a = 2
if ( a in list ):
  print "Line 3 - a is available in the given list“
else:
 print "Line 3 - a is not available in the given list"
```

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

```
Line 1 - a is not available in the given list

Line 2 - b is not available in the given list

Line 3 - a is available in the given list
```

Python Identity Operators:

Identity operators compare the memory locations of two objects. There are two Identity operators as explained below –

Operator	Description	Example
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.	x is y, here is results in 1 if id(x) equals id(y).
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise.	x is not y, here is not results in 1 if id(x) is not equal to id(y).

Example:

```
a = 20
b = 20
if ( a is b ):
   print "Line 1 - a and b have same identity“
else:
    print "Line 1 - a and b do not have same identity“
 if ( id(a) == id(b) ):
      print "Line 2 - a and b have same identity“
 else:
    print "Line 2 - a and b do not have same identity"
b = 30
if ( a is b ):
    print "Line 3 - a and b have same identity“
else:
     print "Line 3 - a and b do not have same identity“
if ( a is not b ):
    print "Line 4 - a and b do not have same identity“
else:
    print "Line 4 - a and b have same identity"
```

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

```
Line 1 - a and b have same identity
Line 2 - a and b have same identity
Line 3 - a and b do not have same identity
Line 4 - a and b do not have same identity
```

Python Operators Precedence:

The following table lists all operators from highest precedence to lowest.

Operator	Description
**	Exponentiation (raise to the power)
~ + -	Complement, unary plus and minus (method names for the last two are +@ and -@)
* / % //	Multiply, divide, modulo and floor division
+ -	Addition and subtraction
>> <<	Right and left bitwise shift

Operator	Description
&	Bitwise 'AND'td>
^ |	Bitwise exclusive `OR' and regular `OR'
<= < > >=	Comparison operators
<> == !=	Equality operators
= %= /= //= -= += *= **=	Assignment operators
is is not	Identity operators
in not in	Membership operators
not or and	Logical operators

Example:

```
a = 20
b = 10
c = 15
d = 5
e = 0
 e = (a + b) * c / d       #( 30 * 15 ) / 5
print "Value of (a + b) * c / d is ",  e
 e = ((a + b) * c) / d     # (30 * 15 ) / 5
print "Value of ((a + b) * c) / d is ",  e
 e = (a + b) * (c / d);    # (30) * (15/5)
print "Value of (a + b) * (c / d) is ",  e
 e = a + (b * c) / d;      #  20 + (150/5)
print "Value of a + (b * c) / d is ",  e
```

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

```
Value of (a + b) * c / d is 90
Value of ((a + b) * c) / d is 90
Value of (a + b) * (c / d) is 90
Value of a + (b * c) / d is 50
```

Python Logical Operators

There are following logical operators supported by Python language. Assume variable a holds 10 and variable b holds 20 then –

Operator	Description	Example
and Logical AND	If both the operands are true then condition becomes true.	(a and b) is true.
or Logical OR	If any of the two operands are non-zero then condition becomes true.	(a or b) is true.
not Logical NOT	Used to reverse the logical state of its operand.	Not(a and b) is false.

Example:

```
# logical and operator
 a = 10
b = 10
c = -10

if a > 0 and b > 0:
    print("The numbers are greater than 0")

if a > 0 and b > 0 and c > 0:
    print("The numbers are greater than 0")
else:
    print("Atleast one number is not greater than 0")
```

Output:

```
The numbers are greater than 0

Atleast one number is not greater than 0
```

Example:

Boolean and operator returns true if both operands return true.

```
>>> a=50
>>> b=25
>>> a>40 and b>40
False
>>> a>100 and b<50
False
>>> a==0 and b==0
False
>>> a>0 and b>0
True
```

Example:

Boolean or operator returns true if any one operand is true

```
>>> a=50
>>> b=25
>>> a>40 or b>40
True
>>> a>100 or b<50
True
>>> a==0 or b==0
False
>>> a>0 or b>0
True
```

Example:

The not operator returns true if its operand is a false expression and returns false if it is true.

```
>>> a=10
>>> a>10
False
>>> not(a>10)
True
```

References:

1. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher
2. Fundamentals of Python first Programmes by Kenneth A Lambert, Copyrighted material Course Technology Inc. 1 st edition (6th February 2009)
3. T. Budd, Exploring Python, TMH, 1st Ed, 2011
4. https://www.tutorialspoint.com/python/index.htm
5. https://www.geeksforgeeks.org/python-programming-language

****END OF THE LECTURE***

THANK YOU