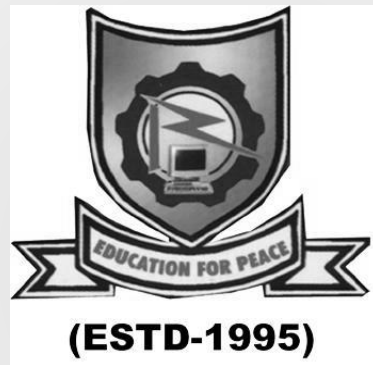


Python Programming



**RGM College of Engineering & Technology
(Autonomous)**

Department of Computer Science & Engineering

Academic Year : 2020-2021

PYTHON OPERATORS - 5



Guido Van Rossum

Dept. of CSE, RGM CET(Autonomous), Nandyal

Learning Mantra

**If you really strong in the basics, then
remaining things will become so easy.**

Agenda:

1. Special Operators

1. Identity Operators
2. Membership Operators

2. Operator Precedence

9. Special Operators

□ There are two types of special operators are there in Python:

1. Identity Operators

2. Membership Operators

1. Identity Operators

□ **We can use identity operators for address comparison.** There are two identity operators used in Python:

i) is

ii) is not

□ r1 is r2 returns True if both r1 and r2 are pointing to the same object.

□ r1 is not r2 returns True if both r1 and r2 are not pointing to the same object.

Eg:

a=10

b=10

print(a is b) ➔ True

x=True

y=True

print(x is y) ➔ True

Eg:

```
a="rgmcet"
```

```
b="rgmcet"
```

```
print(id(a))      ➔ 2730506434688
```

```
print(id(b))      ➔ 2730506434688
```

```
print(a is b)     ➔ True
```


Eg:

```
list1=["one","two","three"]
```

```
list2=["one","two","three"]
```

```
print(id(list1))      ➔ 2730505561800
```

```
print(id(list2))      ➔ 2730505562120
```

```
print(list1 is list2) ➔ False
```

```
print(list1 is not list2) # reference comparison (is & is not) ➔ True
```

```
print(list1 == list2)    # content comparison (==) ➔ True
```

Note:

- ❑ We can use `is` operator for address comparison where as `==` operator for content comparison.

2. Membership Operators

- ❑ We can use Membership operators to check whether the given object present in the given collection.(It may be String, List, Set, Tuple or Dict)

There are two types of membership operators used in Python:

i) in

ii) not in

- ❑ in returns True if the given object present in the specified Collection.
- ❑ not in returns True if the given object not present in the specified Collection.

Eg: Program to demonstrate membership operators

```
x="hello learning Python is very easy!!!"
```

```
print('h' in x)           → True
```

```
print('d' in x)           → False
```

```
print('d' not in x)       → True
```

```
print('python' in x)      # case sensitivity      → False
```

```
print('Python' in x)      → True
```

Eg:

```
list1=["sachin","sourav","yuvi","bevan"]
```

```
print("sachin" in list1)           ➔ True
```

```
print("Boon" in list1)            ➔ False
```

```
print("Lara" not in list1)        ➔ True
```

10. Operator Precedence

- If multiple operators present then which operator will be evaluated first is decided by operator precedence.

Eg:

`print(3+10*2)` $\rightarrow 23$

`print((3+10)*2)` $\rightarrow 26$

The following list describes operator precedence in Python:

()	→ Parenthesis
**	→ exponential operator
~, -	→ Bitwise complement operator, unary minus operator
*, /, %, //	→ multiplication, division, modulo, floor division
+, -	→ addition, subtraction
<<, >>	→ Left and Right Shift
&	→ bitwise And
^	→ Bitwise X-OR
	→ Bitwise OR
<, <=, >, >=, ==, !=	→ Relational or Comparison operators
=, +=, -=, *=...	→ Assignment operators
is, is not	→ Identity Operators
in, not in	→ Membership operators
not	→ Logical not
and	→ Logical and
or	→ Logical or

Eg:

a=30

b=20

c=10

d=5

print((a+b)*c/d) # division operator in Python always going to provide float value as
it's result ➔ 100.0

print((a+b)*(c/d)) ➔ 100.0

print(a+(b*c)/d) ➔ 70.0

Exercise:

`print(3/2*4+3+(10/5)**3-2)`



`print(3/2*4+3+2.0**3-2)`



`print(3/2*4+3+8.0-2)`



`print(1.5*4+3+8.0-2)`



`print(6.0+3+8.0-2)`



Any question?



If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thank You