Python Programming

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

Academic Year : 2020-2021

PYTHON OPERATORS

Guido Van Rossum

Learning Mantra

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

Agenda:

1. Introduction
2. Arithmetic Operators
3. Examples on various Arithmetic Operations

INTRODUCTION

- In general, the person who is doing some operation is known as operator, such as Telephone operator, Camera operator etc.,. In same way, the Python symbol, which is used to perform certain activity is known as operator.

The following Python Operators, we are going to discuss as part of our discussion:

1. Arithmetic Operators
2. Relational or Comparison Operators
3. Equality Operators
4. Logical Operators
5. Bitwise Operators
6. Shift Operators
7. Assignment Operator
8. Ternary Operator (or) Conditional Operator
9. Special Operators
    i) Identity Operators
    ii) Membership Operators
10. Operator Precedence
11. Mathematical functions using math module

1. Arithmetic Operators

Following are the arithmetic operators (7) used in Python:

1. Addition ➔ +
2. Subtraction ➔ -
3. Multiplication ➔ *
4. Normal Division ➔ /
5. Modulo Division ➔ %

In addition to these common arithmetic operators, Python supports two more special arithmetic operators:

6. Floor Division ➔ //
7. Exponential Operator (or) Power Operator ➔ **

Eg:

a = 10

b = 3

print(a+b) ➔13

print(a-b) ➔7

print(a*b) ➔30

print(a%b) ➔1

Floor Division

suppose 10.3 is there, What is the floor value of 10.3?

- Answer is 10

What is the ceil value of 10.3?

- Answer is 11

Eg:

print(10/2) ➔5.0

Note:

- In Python division operation always meant for floating point arithmetic and gives floating point value as it's result.
- This is Python 3 specific behaviour, In Python 2 you are going to get 5 as result.

Eg:

print(10/3) ➔3.3333333333333335

- If you want to get integer value as result of division operation, you need to make use of **floor division(//)** operator.
- floor division(//) operator meant for integral arithmetic operations as well as floating point arithmetic operations.
- The result of floor division(//) operator can be always floor value of either integer value or float value based on your arguments.
- If both arguments are '**int**' type, then the result is '**int**' type.
- If at least one of the argument is **floa**t type, then the result is also **float** type.

Eg:

print(10//2) ➔5

print(10/3) ➔3.3333333333333335

print(10.0/3) ➔3.3333333333333335

print(10.0//3) ➔3.0

print(10//3) ➔3

print(10.0//3.0) ➔3.0

NOTE:

- Floor integer value of 3.33333 is 3
- Floor float value of 3.33333 is 3.0
- Floor integer value of 5.9997777 is 5
- Floor float value of 5.9997777 is 5.0

Eg:

print(20/2) ➔10.0

print(20.5/2) ➔10.25

print(20//2) ➔10

print(20.5//2) ➔10.0

print(30//2) ➔15

print(30.0//2) ➔15.0

Power Operator or Exponential Operator (**):

Eg:

print(10**2) # meaning of this is 10 to the power 2 ➔100

print(3**3) ➔27

Note:

- We can use +,* operators for string type also.
- If we want to use + operator for string type then compulsory both arguments should be string type only otherwise we will get error.

Eg:

print(10 + 20) ➔30

print("karthi" + "sahasra") ➔karthisahasra

print("karthi" + "10") ➔karthi10

print("sahasra" + 10) ➔**TypeError:** can only concatenate str (not "int") to str

- If we use * operator for str type then compulsory one argument should be 'int' and other argument should be 'str' type.

Eg:

print("karthi" * 3) ➔karthikarthikarthi

print(3 * "karthi") ➔karthikarthikarthi

print(2.5 * 'karthi') ➔ **TypeError:** can't multiply sequence by non-int of type 'float'

Eg:

print('karthi' * 'sahasra')

➔**TypeError:** can't multiply sequence by non-int of type 'str'

print('karthi' * int('3')) ➔karthikarthikarthi

Note :

➔ '+' operator for String (Concatenation Operator)

➔ '*' operator for String (String Multiplication Operator (or) String Repetition Operator)

ZeroDivisionError:

For any number x,

x/0 or x//0 or x%0 always raises "ZeroDivisionError“

Eg:

10/0 ➔**ZeroDivisionError:** division by zero

10.0%0 ➔ **ZeroDivisionError:** float modulo

10.0//0 ➔**ZeroDivisionError:** float divmod()

10//0 ➔**ZeroDivisionError:** Integer division or modulo by zero

Arithmetic Operators with bool type

- Internally Boolean values are represented as integer values only.

Eg:

print("karthi" * True) #True ➔ 1

Output: Karthi

print("karthi" * False) #False ➔ 0, Output is an empty string

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