



Maratha Vidya Prasarak Samaj's
K.T.H.M. COLLEGE, NASHIK-2.
NAAC Accredited "A++" Grade (CGPA: 3.79)
College with potential for Excellence" By UGC
Pune University "Best College Award 2017-18"

Department of Electronic science Class: S. Y. B. Sc. (Comp. Sci.)	REMARK
Name: _____	
Roll no: _____ Batch: _____ Expt. No.: _____ Date: / /	

Title: Introduction to Python programming.

AIM: To write program for arithmetic operation using python language.

OBJECTIVES:

1. To understanding the basics of python programing.
2. To write python program for arithmetic operations.
3. To find equivalent resistance using python programming.

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python Syntax compared to other programming languages

Python was designed for readability, and has some similarities to the English language with influence from mathematics.

Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python fetchers:

Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of UNIX.

- Scalable: Python provides a better structure and support for large programs than shell scripting. Apart from the above-mentioned features, Python has a big list of good features, few are listed below:
- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Current Applications of Python

- A number of Linux distributions use installers written in Python example in Ubuntu we have the Ubiquity
- Python has seen extensive use in the information security industry, including in exploit development.
- Raspberry Pi– single board computer uses Python as its principal user-programming language.
- Python is now being used Game Development areas also.

Pros:

- Ease of use
- Multi-paradigm Approach

Cons:

- Slow speed of execution compared to C,C++
- Absence from mobile computing and browsers
- For the C, C++ programmers switching to python can be irritating as the language requires proper indentation of code. Certain variable names commonly used like sum are functions in python. So C, C++ programmers have to look out for these.

Programming:

1) Print Hello Word Program

```
print("Hello, world!")
```

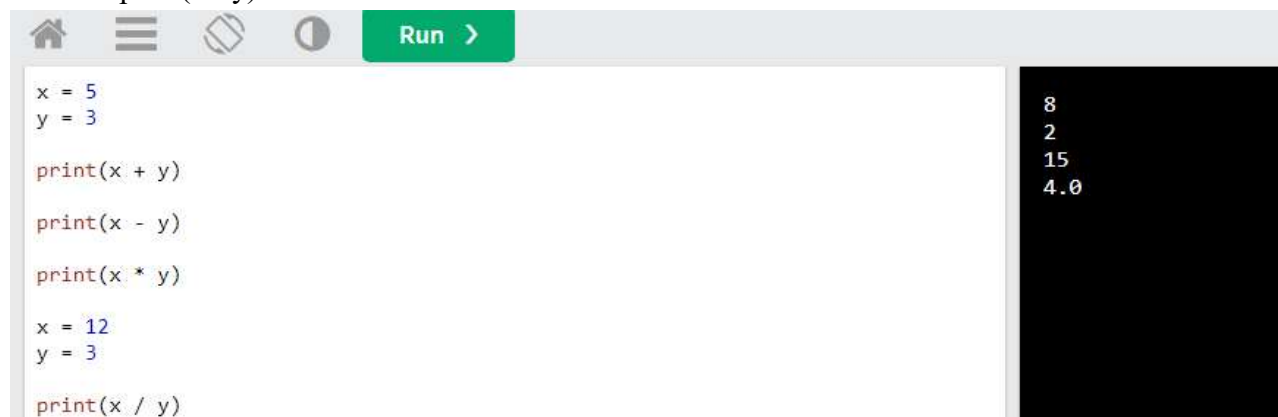
Output

A screenshot of a Python IDE interface. The top bar contains icons for home, menu, save, and a 'Run' button with a right arrow. The main editor area shows the code `print("Hello, World!")`. To the right of the editor, the output is displayed as 'Hello, World!' on a black background.

2) Add Two Variables

Program

```
x = 5
y = 3
print(x + y)
print(x - y)
print(x * y)
x = 12
y = 3
print(x / y)
```

A screenshot of a Python IDE interface. The top bar contains icons for home, menu, save, and a 'Run' button with a right arrow. The main editor area shows the code for adding and subtracting variables. To the right of the editor, the output is displayed on a black background, showing the results of the calculations: 8, 2, 15, and 4.0.

3) Add, Sub, Multi and Divide Two Numbers with User Input

Program

```
# Store input numbers
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')
# Add two numbers
sum = float(num1) + float(num2)
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')
# Sub two numbers
Sub = float(num1) - float(num2)
```

```

print('The Sub of {0} and {1} is {2}'.format(num1, num2, Sub))
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')
# Multiplication of two numbers
multi = float(num1) * float(num2)
print('The multi of {0} and {1} is {2}'.format(num1, num2, multi))
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')
# Divide two numbers
divi = float(num1) / float(num2)
print('The divi of {0} and {1} is {2}'.format(num1, num2, divi))

```

```

main_output - Notepad
File Edit Format View Help
Enter first number:
10
Enter second number:
10
The sum of 10 and 10 is 20.0
Enter first number:
10
Enter second number:
10
The Sub of 10 and 10 is 0.0
Enter first number:
10
Enter second number:
10
The multi of 10 and 10 is 100.0
Enter first number:
10
Enter second number:
10
The divi of 10 and 10 is 1.0

** Process exited - Return Code: 0 **
Press Enter to exit terminal
Ln 1, Col 1 100% Unix (LF) UTF-8

```

4) Find series and parallel equivalent of Resistor

```

# Store input value of resistors numbers
resistor1 = input('Enter first Value: ')
resistor2 = input('Enter second Value: ')
# Equivalent series resistor
Req = float(resistor1) + float(resistor2)
print('The Req of {0} and {1} is {2}'.format(resistor1, resistor2, Req))
resistor1 = input('Enter first Value: ')
resistor2 = input('Enter second Value: ')
# Equivalent Parallel resistor
RPeq = (float(resistor1) * float(resistor2)) / (float(resistor1) + float(resistor2))
print('The RPeq of {0} and {1} is {2}'.format(resistor1, resistor2, RPeq))

```

Output:

Enter first Value:

100

Enter second Value:

100

The RPeq of 100 and 100 is 50.0

Result: Basic of python programing has been studied and written program for arithmetic operations.