Python – Basics via Examples

Saurish Chakrabarty

Acharya Prafulla Chandra College

The topics discussed are taken from the WBSU B.Sc. (NEP) syllabus (First Semester, Paper: PHSDSC101P – Mathematical Methods I Lab)

References

- C. Hill, "Learning Scientific Programming with Python"
- A. K. Gupta, "Scientific Computing in Python"
- J. V. Guttag, "Introduction to Computation and Programming Using Python"

1 Using python as a simple calculator

Enter the following lines into a python prompt.

```
>>> 57*23
```

```
>>> 1000**0.5
```

2 Using variables

Calculating the volume of a cuboid.

```
>>> a=2
>>> b=3
>>> c=4
>>> vol=a*b*c
>>> print(vol)
```

Variables are names given to objects in python, using which they may be called. These names can have letters, numbers and underscores. A variable name must not start with a number. It must not be a reserved keyword (`import`, `as`, `for`, `lambda`, etc) or a builtin constant (`True`, `False`, `None`).

3 Program files or scripts

Python commands like the lines given above, may be written into a text file with an extension *py*, e.g., *volOfCuboid.py*. Such a file is called a python program or a python script. On a python prompt, a python program may be run using, e.g., `exec(open('volOfCuboid.py').read())`. Alternatively, on a linux terminal, you may enter `python volOfCuboid.py`. If you use an IDLE such as Spyder, there is a button in the editor which you may click, or a corresponding shortcut.

4 Types of numbers

Numbers in python are of three types, *integers*, *floats* and *complex*. Unlike most programming languages, integers in python are of arbitrary precision, i.e., the number 128989851589589153285105853218908539189532589108 is stored without any approximation. Floating point numbers are stored with a finite precision, usually 15 significant digits. For convenience, numbers may be grouped using the _ character which is ignored by the python interpreter, e.g., 299_792_458 is the same as 299792458. A complex number, $3.2 + 4.1i$, is written as `3.2+4.1j` or as `complex(3.2,4.1)` in python.

5 Operations with numbers

Usual mathematical operations can be done – addition(+), subtraction(-), multiplication(*), division(/), exponentiation(**), and, quotient(//) and remainder(%) calculations. Using parentheses (first brackets), different parts of an expression may be grouped together. Multiple exponentiations are done from right to left. Other multiple operations with same precedence levels are done from left to right. The precedence rule (like the BODMAS rule) which is followed is as follows.

exponentiation → multiplication, division, quotient and remainder → addition and subtraction.

6 Methods for python objects

Most python objects have available methods associated with them. For example try the following.

⋙ `0.625.as_integer_ratio()` (not too useful for real numbers with recurring parts).

⋙ `(1+3j).imag`

⋙ `(1+3j).conjugate()`

7 Some common functions

Try using the following functions.

⋙ `print(2**4)`

⋙ `abs(-5)`

⋙ `abs(2.3)`

⋙ `abs(2+3j)`
The `abs` function is overloaded. It does different things depending on what it is fed with.

⋙ `round(3.5)` The `round` function gives the nearest integer to a floating point number. For half-integers (1.5, -2.5, etc), it gives the nearest even numbers (so that the probability of giving more is the same as that of giving less, unlike marks in exams).

⋙ `input('Enter your Name: ')` or just `input()`
The interpreter waits for an input which, by default, is taken as a string (bunch of characters, to be discussed later). The input may be converter to a suitable type using the `eval` function. Functions like `int`, `float` etc may also be used.

8 Help and Dir Commands

- For any python function whose name you came to know, try the following.
  ⋙ `help(SOME_PYTHON_FUNCTION)`
  You may try the `print` function or the `abs` function.

- For any python object, try,
  ⋙ `dir(SOME_OBJECT)`
  to get a list of available *methods*, many of which are functions, on which you may run `help`. Methods whose names look like __SOMETHING__ are usually not useful for most programmers.

9 math module

The `math` module is a useful module from which various mathematical functions and constants may be used. It can be imported using one of the following ways.

⋙ `import math`
Objects from the math module are referred to as, e.g., `math.pi`, `math.exp` etc.

⋙ `import math as m`
This gives a shorter or more convenient name to the module.

⋙ `from math import sin,pi,exp`
Only what is needed may be imported.

⋙ `from math import *`
This is discouraged because of possible conflicts with other function names.

10 if statement

This can be used in two ways.

- Inline:
  Suppose `x` is a number which has been defined.
  `signx=-1 if x<0 else 1`
- Block:

```
x=23
if x%2==0:
    print(2)
elif x%3==0:
    print(3)
elif x%5==0:
    print(5)
elif x%7==0:
    print(7)
else:
    print('not a multiple of single digit primes')
```

11 Lists and Tuples, Mutability and Immutability