ED5340 - Data Science: Theory and Practise

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Python identifiers and keywords

- P.I. is a name to identify variable, function, class, module etc.
- rules for identifiers
  - starts with an alphabet or _
  - followed by zero or more letters, _ or digits
  - keywords cannot be used

Datatypes in Python

Recall Datatypes in C / C++ (if you know)

- int
- float
- complex
- bool
- string
- bytes

Some differences with C

- int is of arbitrary precision (as you type, it will allocate memory!)
- floats are 64-bit double precision values (max. value?) - fractional or exponential form
- Only strings and no chars
- complex contains real and imag.
- bool takes only true or false
- bytes, containers and user-defined ones (classes) - we will see later.

Demo using L2_demodatatypes.py

Identifiers / Variables

variable names follow similar rules in C

- No need to define a variable (dynamic typing / allocation)
- `type’ function is used to find out an object type
- multiple assignments can be done in different ways
  - a = b = 3
  - a = 3; b = 3 # semicolon as a separator
  - a, b = 3, 4.45 # This is a very interesting way.

Arithmetic Operators

- Unary operators (+, -, ????)
- Binary operators (+, - , *, / , //, **, %)
- Ternary operator (remember the ? :)

Unary Operators

- Unary + and - are same as in C/C++
- No unary increment in decrement operators
- Typecasting (more of C++ style)

Binary operators, precedence, assignment expression

- + , - and * are as in C
- / is True division (No integer division as in C). 3/5 will not yield 0.
- ** is the exponentiation i.e. a ** b is equal to a^b
- // return quotient (without the fractional part, like a ‘floor’ operation)
- % gives remainder - could get unexpected result!
- Operator precedence - PEMDAS
- +=, -=, *=, /=, //=, **=, %=

PEMDAS

Operator precedence

Demo using L2_Operatordemo1.py

What will be the answer to the following?

- 10 // 3
- -10 // 3
- 10 // -3
- 10 % 3
- -10 % 3
- 10 % -3

Conversions between datatypes

- Mixed mode
    - op. between int and float will yield float
    - op. between int and complex will yield complex
    - op. between complex and float will yield complex
- Convert one to another using built-in functions int( ), float( ), complex( ), bool( )

Built-in math functions

- E.g. abs(x), pow(x,y) etc.
- Qn3: Find the min and max values of a given set of values
- Qn5: Given x and y, find the quotient and remainder. Round them to TWO digits after decimal point.

Object - data and member functions

`complex’ object

```
a = 1+ 2j

print(a.real) # Data

print(a.imag) # Data

print(a.conjugate()) # Function
```

Demo using L2_Complexdemo.py

Library functions (called as modules)

E.g. for more sophisticated mathematical functions

- Look at modules such as math, cmath, random, decimal etc.
- Use `import math' for using math module
- math module has mathematical functions, trigonometric functions etc.

Module - How to put into use?

math module as example

```
import math

pi = math.pi; r = 5

area = pi * r ** 2

print(area)

area1 = math.pi * r ** 2

print(area1)
```

importing function from a module

cos function in math

```
from math import cos

x = math.pi / 3

y = cos(x)  # No need to use math.cos
```

Demo using L2_import_example.py

Some questions to explore

- Qn6: Find out the regular math function (not in modules)
- Qn7: Find out the trigonometric functions available in math module.
- Qn8: Find out the various rounding functions for a number.
- Qn9: Find out the various styles of commenting
- Qn10: How do you write a `multi-line’ statement?