Python Q&A

Baboo J. Cui, Yangang Cao

June 20, 2019

Contents

1 Python Itself

1. How to install python?
   Use conda directly or use brew etc
2. How to run a python code in cosole?
   $ python file_name.py
3. Types of python execution mode?
   Interactive mode (>>>) and command line mode (directly run python file by interpretor)
4. Strong type vs weak typrs?
   If type needs to be specify when declaring the var
5. Python3 default coding?
   UTF-8 for code and unicode in memory
6. Legal variable naming?
   Letter, number and underscore, can't started with a number
7. How to declare a constant?
   Use all capital letters, still fake constant
8. Camel-case vs underscore naming?
   Just choose the one you like
9. Basic I/O control?
   2 func, input() to get str and print() for output
10. What is duck typing?
    Way of object inference, it is a type if it looks like it

2 Pythob Basics

1. "//" operator?
   For getting the whole part of the division
2. "%" operator?
   For getting the residue
3. How to calculate a to the power of b?
   2 ways: a**b or pow(a, b), tip pow() is built-in
4. Multiplication between int and str?
   Just repeat the str for int times
5. Type coercion?
   Type(o), change o to Type, may cause error
6. Type of value from input()?
   Must be str, even number from this way will be str
7. How to get the type of an object?
   Function type() will return the type

8. How to get keywords of python?
   >>> from keyword import kwlist, and print it

9. How many kw in python?
   33

10. While else structure?
    Else part will be execute if break statement in while is not executed

11. Which is faster while 1 or while True?
    while 1 because bool is a subclass of int, similarly if x is faster than if x=True

12. How bool can be coerced to int?
    True to 1 and False to 0

13. How number can be coerced to bool?
    0 to False and others are True

14. Logic AND OR and XOR?
    Corresponds to &, | and ^ , in terms of bit

15. Python print() format control?
    Placeholder: %, string: %s, number: %d, %f, %x(hex), string canonical: %r, tip r is for repr. Then follow %(var1, var2)

16. Python print() format control with format()?
    Use {} as place holder and follow .format(var1, var2)

17. Common coding?
    UTF-8, ASCII, GBK, unicode

18. Relationship between bit and Byte
    1 Byte = 8 bit, bit is the smallest unit in computer

19. x or y?
    If x is true, return x, otherwise return y

20. x and y?
    If x is true, return y, otherwise return x, easy for bool var, may be complicate for numbers

21. Priority of NOT, AND, OR?
    NOT > AND > OR

22. How many bits are there if decimal -> binary?
    Use int.bit_length() to get

23. How str can be coerced to bool?
    Empty str to false and others to True

24. How bool can be coerced to str?
    True to 'True' and False to 'False'

25. Does comma has effects in print()?
    There will be a white space, print("a", "b") -> "a b", there is a comma

26. Usual input and output function of python?
    I: input(), O: print()

27. Is python case-sensitive?
    Yes

28. How indent is achieved in python?
    4 white space, not TAB! And don't mix them together

29. How to comment code in python?
    Multiline: """xxx""", single line: # xxx

30. How to build up a block?
    Start with colon (:) and block code with indent

31. Data type that python can directly deal with?
    int, float, str, bool, None(not 0), list, dict, etc

32. Notion of number in binary, oct and hex?
    0b, 0o and 0x

33. Essential of assignment to a var?
    Let the var point to a certain value(be clear about the mechanism)

34. Coding of str in python?
    Unicode in memory, note that python code is in utf-8 for storage

35. Size of ASCII?
    I Byte = 8bit, 256 cases

36. Size of Unicode?
    2 Bytes = 16bit, 65536 cases

37. Convert char to unicode number?
    ord()

38. Convert unicode number to char?
    chr()

39. How to declare str in byte?
    Prefix: b'xxx', each one occupy 1 byte

40. How to get size of an object in memory?
    Use sys.getsizeof() function, note this will return the size of an object, which may be quite big! Str is very complicate

41. How to use python to run bash command?
    Use os.system(COMMAND)

42. How to decoding a binary to string?
    Use str.decode() method, arg could be "ascii", "utf-8", "gb2312", to ignore errors, add errors='ignore'

43. How to encode a string to binary?
    Use str.encode() method

44. How to format a decimal?
    Use {num:a.b f} where num is the position, a is num of integer bit, b is for decimal, f represent float

45. How to find length of a list?
    Use len() function, len(LIST) will return

46. How to find the last element of a list?
    Use LIST[-1], index of -1 represent the last element, -2 for the second last one, etc

47. Difference between list and tuple? List is mutable, declared by [ ] and tuple is immutable and declared by ()

48. List.append(o)?
    Append obj o to the end of list

49. List.insert(idx, o)?
    Insert obj o to index idx

50. List.pop() or list.pop(i)?
    Pop out the last element, and pop(i) pop the element with index i

51. Acquire the idx element of list?
    List[idx]

52. How to init a tuple with one element?
    T = (obj1,), the key is to add a comma after obj1

53. How to understand immutability of tuple?
    Each element that points to won't be mutable, but the content that each points may be mutable

54. Condition control structure?
    If else, or if elif else

55. How to loop an iterable object by for?
    Use for loop, for item in iterable_obj

56. How to loop by while?
    Use while structure: while condition: block_exe

57. How to generate a range?
    Use range() function, range(a) will generate [0...a-1] and range(a, b) will generate [a, a+1...b-1]

58. Difference between break and continue?
    Break will stop the loop and continue will simply jump to next loop, usually both are related to if condition

59. How to terminate a python program directly?
    Use ctrl + c to kill the process

60. What is dict?
    A key-value mechanism, also known as map in other languages

61. How to declare a dict?
    Format: {key1: val1, key2: val2, ... }, curly braces, comma separation, colon separation, key must be immutable

62. Why dict can find a key value so fast?
    Use binary tree mechanism, which lead to log(n) time scale

63. How to determine if a dict has a key?
    Use code: if key_val in dict, or: dict.get(key_val)

64. How to delete a key-value in dict?
    Use code: dict.pop(key)

65. Trade off between list and dict?
    List is slower but occupy smaller memory, dict is faster but requires more memory

66. How to create a set?
    Use curly braces:{item1, item2, ... } or use code set(list), which will turn a list to set

67. Properties of set?
    It has no order and there is no repeats

68. Add and remove key of a set?
    Corresponding to set.add(key) and set.remove(key)

69. Common operation on sets?
    Union "|" and intersection "&"

70. Difference between dict and set?
    Share the same mechanism but set doesnt have value, can't put mutable obj in

71. Is str mutable? What about list?
    Str is immutable and list is mutable

72. Immutable objects?
    Number, string and tuple. Contents in certain add can't be changed

73. How to find the memory address of an object?
    Use function id(), id(o) will return the memory add

74. How to get console args?
    By import sys, and then use sys.argv[idx]

75. What is configurative programming?
    A framework is created such that coding is like setting configuration

76. Mechanism of importing a module?
    Python interpreter just go through line by line(interview question)

3 Functions

1. How to define a function?
   Easy...

2. How to specify the return type of a function?
   Use syntax def FUN_NAME () -> RETURN_TYPE:

3. Disadvantages of declaration like def func(a=[1,2])?
   Args are mutable, may cause unpredicted errors!

4 OOP-Object Oriented Programming

1. What is OOP?
   Take object as the basic unit for programming, OOP vs procedure-oriented-programming(set of instruction)

2. Why OOP?
   Everything are objects(a set of objects) and execution becomes the interaction between instances

3. What does an object include?
   2 parts, properties(data) and methods(functions)

4. Class vs instance?
   Class is template, instance is the specified class, instance is based on class

5. 3 characteristics of OOP?
   Encapsulation, inheritance, polymorphism

6. How to add properties to an instance dynamically?
   Use form instance.var_name = val to dynamically add a property, this won't work for other instances

7. How to add method to class dynamically?
   Use form class.method_name = func_name, this may arise warning from the IDE

8. What is ___slots___ for?
   A var in class which control the possible property names

9. What is the range of ___slots___?
   Only work for current class, won't work for subclass

10. What is the type of ___slots___?
    It is a tuple

11. What if both a class and its base have ___slots___?
    The possible property will be the union of the two ___slots___

12. What is @property for?
    A decorator which may help us to take a method like property

13. How many ways are there to define class methods?
    3 ways, regular definition(related to self), decorated by @classmethod(related to cls) and @staticmethod

14. Difference between self and cls?
    Self is bound to instance of class and cls is bound to class

15. Call of a regular method in a class?
    Can be called by object but not class, or by class with first arg to be an instance of that class

16. Call of a class method in a class?
    Can be called by both instance and class directly

17. Call of a static method in a class?
    Can be called by both instance and class directly

18. Why static method in class instead of an independent func?
    To indicate that the method belongs to the class and by inheritance, code can be managed better

19. What is MRO?
    Method resolve order, a mechanism for inheritance

5 Metaclass

1. Biggest difference between static and dynamic language?
   Static: definition is done during compiling process. Dynamic: definition are created during runtime

2. How class can be defined?
   Two ways, by general declaration and by type() method

3. How to define a class by type()?
   Use form CLASS_NAME = type('NAME', (BASE_CLASS,), dic(METHOD1=FUNC1,...)), dynamic way to define a class

4. What does type() do?
   It can show the type of an object. It also can be used to define a class

5. Difference between ___new___() and ___init___()?
   ___new___() create the obj and ___init___() initialize the obj. Initialization comes after creation

6. How class is created(not asking how defined)?
   During running time, essentially by function type(), tip: not from declaration!!!

7. Difference between general class definition and type()?
   First one is in static way, the second one can be used in dynamic process, essentially, both share the same purpose

8. Dynamically create a class by static or dynamic language?
   Easier for dynamic language(itself support), static language requires constructing the source code in the beginning.

9. What is the type of a class?
   All class name itself has type: "type"

10. Relationship between type and object?
    Type is subclass of object, metaclass of object is type. Both are created during the execution of interpreter

11. What is metaclass?
    An class that controls how another class is defined, can be considered as template for other classes

12. What is the parent class of metaclass?
    Must be type, cannot be object

13. How class derived from metaclass is created?
    By calling type.___new___(mcs, name, bases, attrs), it is the return of ___new___() function in the metaclass!

14. How metaclass is defined?
    Name end with Metaclass by convention(not necessary), inherit from type, define ___new___() to control how other classes are created

15. What type is name in ___new___(mcs, name, bases, attrs)?
    It is str, who has value of the name of the class that take it as template

16. What type is bases in ___new___(mcs, name, bases, attrs)?
    It is tuple, a tuple that contains the parent classes in the target class

17. What type is attrs in ___new___(mcs, name, bases, attrs)?
    It is dict, has form var_name: value, var could be either function or properties

18. How are args in ___new___(mcs, name, bases, attrs) passed?
    When interpreter reading a class, it will use type() to create a class, and args are passed in the conventional way

19. Does any class has a corresponding metaclass?
    Yes, and the metaclass is usually implicitly inherited

20. What is abstract class?(Not that important in python)
    Class abstraction from many classes with certain similarities, it has a higher abstraction, a template for other classes

21. How to declare an abstract class in python?
    First from abc import abstractmethod, ABCMeta, then declare a class with arg metaclass=ABCMeta and decorate method with @abstractmethod

22. Characteristics of abstract class?
    Methods only have declaration, no implementation. Cannot be instantiated. Must have abstract method and must be overridden latter.

23. What is interface class?
    Like header, cant be instantiated, only contains method declaration, contains methods, properties, event etc... Doesnt contain constants etc...

6 Enum

1. What is enum class?
   A enumeration, just list everything, like a key value system linked by equality, usually for constants

2. How to create a enum class?
   First from enum import Enum, and then create a class inherit from Enum

3. How to get key of enum?
   Directly use dot "." or by enum_name['KEY_VAL']

4. How to get key value in a enum?
   By enum.KEY_VAL.value

5. What is @unique for?
   Make sure that both key and value won't repeat! (bi-jective)

6. How to import unique?
   Use statement from enum import unique

7. What is enum generally for?
   For finding key by value

7 Error, Debug and Test

1. What is bug?
   Any unexpected thing, bug must be repaired

2. What causes bug?
   Programming error, wrong input, unexpected condition(disk is full...)

3. What is python pdb?
   A way of debug, python debug

4. What is error code(value)?
   When something go wrong, there will be a specific return value like return -1

5. Disadvantages of error code?
   Mix the error code and general return value together

6. Try, except, else, finally?
   Try to execute whats in try, if any error, jump to except and else part will be execute if no error in try, then goes to finally(optional)

7. What if there might be more than a type of error?
   One try can contain more than one expect block

8. How to write except part?
   Write form except: or except ERROR_TYPE as e:

9. Except ZeroDivisionError as e:, what type are they?
   ZeroDivisionError has type: type since it is a class, e has type:<class 'ZeroDivisionError'> since it is an instance of the error class

10. What is the base class of all error classes?
    All are inherited from BaseException

11. Range of try... except works?
    Function contains it and any outer part that contains them, catch the error at nice position will be okay, don't need to put it everywhere

12. What if error does't caught by any except?
    It will be thrown upper until caught by Python interpreter to print error and stop the program

13. How to read Traceback (most recent call last):
    It shows that there are error and goes from top to bottom, the last line show the real reason

14. How to record error?
    Import logging and add logging.exception(e) to output error and finish the program

15. What essentially an error is?
    It is an instance of a class

16. How to throw an instance of error?
    Use raise, need to raise an instance of class that is well designed that inherit from some error class

17. What could raise do if put into the block of except?
    Convert one error to other type, should be logically reasonable

18. Easiest way of debugging?
    Use print(), have to delete it after debugging... sad

19. Syntax of assert?
    Have form assert CONDITION, ERROR_MESSAGE, condition should be true, otherwise will be error

20. How to stop assert statement during execution?
    Execute .py file with form $ python -O xxx.py

21. How to log info?
    Use form logging.info(STRING)

22. How to config the level of logging?
    Add code logging.basicConfig(level=logging.INFO) after import logging

23. Level of logging?
    There are 4: debug, info, warning, error, the higher level you set, the lower cases will be ignored

24. How to start a program by pdb?
    Use form $ python -m pdb xxx.py

25. How to see code in pdb?
    Use command: “1”

26. How to execute code one line by another?
    Use command: “n”

27. How to get value of a var in pdb?
    Use command: “p VAR_NAME”

28. How to quit pdb?
    Use command: “q”

29. How to use pdb.set_trace()?
    Put it at position where might have error, it will pause the program there , use “p Var” to debug and press “c” to continue

30. What is the best way of debugging?
    Ultimately. . . logging

31. What is TDD?
    Short for Test-Driven-Development

32. What is unit test?
    Check if a module, function or class work correctly, put all test conditions in a module, after revision, check if all conditions could pass the test

33. Advantage of unit test?
    It can almost guarantee that the behavior of code is correct

34. What does a test unit class inherit from?
    From unites.TestCase

35. Purpose of methods in unit test?
    By convention, methods start with “test” are test methods, otherwise not test methods which won’t be executed during test

36. What does assertEqual() do?
    Has form self.assertEqual(abs(-1), 1), check if expected output equals to target output

37. What does assertRaises() do? Has form with self.assertRaises(ERROR_TYPE):
    BLOCK, if do anything in BLOCK, there will be ERROR_TYPE will be thrown

38. How to run unit test by coding?
    Directly use statement: unites.main()

39. How to run unit test in console?
    By command: python -m unittest xxx.py

40. What are setUp() and tearDown() functionality?
    They will be executed before and after a test

41. Which module is used for doc test?
    Use import doctest

42. How to run doc test?
    Use statement doctest.testmod()

43. Where the doc test code should be?
    Within the triple comments: """xxx"""

44. How to write test doc?
    All statement start with >>> STATEMENT(could be more than one) and then the next line follow the output

8 Others

8.1 Virtual Environment

1. What is virtual environment for?
   To build isolated environment for different programs. Different programs may depend on different python version and packages

2. What command is used to build virtual env?
   Use virtualenv

3. How to install virtualenv?
   Use command $ pip3 install virtualenv by default, or $ conda install virtualenv(not recommend for conda, use create)

4. How to build up a virtual env?
   Use command $ virtualenv no-site-packages ENV_NAME, where the ENV_NAME will generate a new folder that contains everything

5. What is no-site-packages for?
   Avoid the copy of the third party package, make the env very clean

6. How to activate virenv?
   Use command $ source ENV_PATH/bin/activate to activate

7. Where the packages will be installed in virenv?
   At path: ./lib/pythonx.x/site-packages/

8. How to deactivate a virtual env?
   Use command $ deactivate

9. How to show all installed packages?
   Use command $ pip list

10. How to check packages installed under conda?
    Use command $ conda list

11. How to check existing env?
    Use command $ conda env list or $ conda info -e

12. How to check info of conda?
    Use command $ conda info

13. How to create virtual env by conda?
    Use command $ conda create -n ENV_NAME python=3.6, python version must be added!

14. How to activate virenv by conda?
    Use command $ conda activate ENV_NAME

15. How to quit virenv by conda?
    Use command $ conda deactivate

16. How to delete virenv by conda?
    Use command $ conda remove -n ENV_NAME all (2 - before all)

17. How to install a package?
    Use command $ conda install PACK_NAME

18. How to delete a package?
    Use command $ conda remove PACK_NAME, this is intent to delete package under that environment

19. How to update conda?
    Use command $ conda update -n base -c defaults conda

20. How to install requirement.txt?
    Use command $ pip install -r requirements.txt or by $ conda install -file requirements.txt

21. How to generate requirement.txt for virenv?
    Use command $ pip freeze > requirement.txt

22. How to add Tsinghua source?
    Use command $ conda config - -add channels
    https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
    $ conda config - -add channels
    https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
    $ conda config - -set show_channel_urls yes

23. How to reset source?
    Use command $ conda config - -remove-key channels

8.2 GUI

1. Any packages for GUI?
   Tk, wxWidgets, QT, GTK...

2. Advantage of TK?
   It can be used directly

3. Which language is TK based on?
   Based on TCL

4. How to import TK?
   Add statement from tkinter import *, MUST be *

5. Which class is Application inherit from?
   Class Frame

6. What args should ___init___() of Application contains?
   Two args: self and master=None

7. What is done in ___init___()?
   3 things: Frame.___init___(self, master), self.pack(), self.createWidgets(), (declaration and pack)

8. What is widget?
   Any GUI object in TK is known as widget

9. What is pack() for?
   Add widget into GUI container and achieve layout, after declare a component, pack() function is mandatory

10. Functions for layout?
    2 functions, pack() for easy layout and grid() for more complex ones

11. What is label API?
    Use statement Label(self, text="TEXT"), and then pack()

12. What is button API?
    Use statement Button(self, text="TEXT", command=self.COMMAND), and then pack()

13. What is entry API?
    Use statement Entry(self), and then pack()

14. How to use messagebox?
    First import tkinter.messagebox as messagebox, and follow messagebox.showinfo('TITLE', 'MESS_DISP')

15. How to instantiate an application?
    3 steps, create instance, set instance.master.title("TEXT") and start instance.mainloop()

9 PyCharm

1. How to search?
   Use shortcut cmd+O, then easy

2. Meaning of set a folder as resources root?
   For searching file, will add the path to search path