

Python Frequently Asked Interview Questions

- **Naren Allam**, Trainer & Consultant

Introduction

1. Python 2 vs Python 3
2. Compiled and interpreted
3. Python distributions
4. Can you run python on JVM and .Net CLR
5. What is .pyc and .pyb files ?
6. Default Character encoding in python 2 and python 3
7. What is slicing?
8. output ?

```
x = raw_input()
y = raw_input()
print x + y
```

Strings

9. Reverse a string in Python
10. #output
"Hello" * 3
11. #output
s = "Hello world! 123\$"
s.upper()
12. x = int("2345.6") ?
13. Explain indexing and slicing?
14. S = "Hello world!"

Find the outputs of
S[1:9:2]?

`S[::-2]`?

`S[-4:-1:1]`?

`S[::-1]`?

`S[3::-1]` ?

15. #output

`date = '12/02/1984'`

`date [: 5: -1][: -1]`

`date.rsplit('/', 1)`

16. How do you convert string to list?

Operators

17. #output

`x = raw_input("Enter x value")`

`y = raw_input("enter y value")`

`print x+y`

18. Difference between `" / "` and `" // "` ?

19. #output

`bool(0), bool(' '), bool(None)`

20. Swap two variables x, y in Python

21. #output

`s1 = "Hello"`

`s2 ="Polo"`

`s3 ="Hello"`

`s1 is s3 ?`

22. `x = 99`

`y =100`

`x = 200`

Print `"{1}, {0}, {2}, {0}".format(x,y,z)`

23. Is there switch in Python?

24. What is the purpose of `"else"` in `"for"` loop?

Control Structures

25. ternary operator in python?
26. how **for-else** construct works?
27. is there switch case in python ?
28. range() vs xrange()
29. What is iterator?
30. What is generator?

Programs:

1. Write a program to find the given number is prime or not in efficient way, using for-else construct?
2. How do you check the given number is palindrome or not?
3. print the following triangle using for loop

```
5
4 4
3 3 3
2 2 2 2
1 1 1 1 1
```

Data-structures (list, tuple, set, dict)

31. Difference between list and tuple?
32. Reverse the list?
33. Multiply a list with scalar
34. Multiple types stored in lists?
35. List append Vs extend
36. How do you remove duplicates in a list?
37. List of characters to string?
38. How do you sort a list? What is the internal algorithm it use?
39. What are the uses of tuple?
40. t = (3,) what is this ?
41. x =(3,) what is the type of x ?
42. When do you use tuple?

43. `t = (8,2,5,4,9,1, 3, 7, 10, 6)`
44. #output
- `t[2: 7: 2]`
- `t[: : -1]`
45. What is the internal data structure of set in Python?
46. Explain how `dict()` works internally?.
47. Give two use cases for a `dict()`?
48. When does a dictionary is used instead of a list?
49. What is purpose of `Counter()` data structure?
50. How do you create a dictionary which can preserve the order of pairs?
51. How do we check a key is existing in dictionary?
52. Mutable vs immutable
53. What is list comprehension?
54. What is tuple comprehension?
55. What is `map()`, `filter()`, `reduce()`?
56. difference between `remove()` and `pop()` in list?
57. What is associative container in Python?
58. **Program:** Find the word with maximum number of occurrences in a large text file.

Functions (decorators, generators)

59. What are the iterables in Python?
60. What is iterator and what is generator?
61. How do you write custom generators?
62. What does the `yield` statement do?
63. How do you write custom iterator?
64. What is Recursion? Give Example?
65. What is decorator, usage?
66. Write a function decorator in Python?
67. How are arguments passed by value or by reference?
68. Mention what are the rules for local and global variables in Python?
69. How to use `*args` and `**kwargs` in python?

70. What are positional arguments?
71. What are default arguments?
72. What are keyword arguments?
73. What are variable arguments?
74. What are variable keyword arguments?
75. What is a closer in Python?
76. When do you use variable keyword arguments?
77. How can you get all global variables and local variables in a scope?
78. How “call by value” and “call by reference” works in Python?
79. Difference between “cmp()” function, “==” and “is”?
80. Mention the use of the split() function in Python?
81. What is the purpose of ZIP, enumerate()? How to unzip list of tuples to multiple lists?
82. How can you implement functional programming and why would you?
83. What is lambda and how it works in Python?
84. How method overloading works in Python?
85. Difference between “deepcopy” and “shallow copy”
86. What is **docstring** in Python?
87. What is pure function?
88. What is the use of next function?
89. **Program:** Flatten the below list using recursion

```
L = [34, 5, [ 4, 5, 3, [ 3, 19, 9, 1, 2 ] , 5], 13, 4, [ 6, 14] ]
```

Files

90. How do you read line by line from a file in python ?
91. with open('abc.tx') as f:

```
s = f.read()
```


What happens if 'abc.txt' doesn't exist?

Oops

92. Is Python object oriented? what is object oriented programming?

- 93. What is a Class? How do you create it in Python?
- 94. What is abstract class and how to implement in Python?
- 95. Difference between static and class methods?
- 96. How to create private variable in Python? Can you access private variables?
- 97. Creating inline object and class objects?
- 98. How to create callable objects in Python?
- 99. How do you sort class objects in a list?
- 100. What is the difference between “`__str__`” and “`__repr__`”?
- 101. What is `exec()` and `eval()`?
- 102. What is **with**, and what is a context manager?
- 103. Create context manager and decorator using a class.
- 104. What are the magic methods in Python?
- 105. What is the difference between a class attribute and an instance attribute?
- 106. What is **descriptor**?
- 107. What is a **property**?
- 108. How to implement operator overloading in Python?
- 109. What is a **metaclass** in python?
- 110. Explain Inheritance in Python with an example.
- 111. ‘Super’ in python?
- 112. What is MRO?
- 113. What is namespace in Python?
- 114. What is polymorphism? When do you use polymorphism?
- 115. How do you store object in a `set()` or `dict()` as keys and how do you provide hashing criteria for a class?

Multi-threading

- 116. What is the difference between concurrency and parallelism?
- 117. Why threading is not recommended in Python?
- 118. Difference between threading and multiprocessing?.
- 119. What is Global interpreter lock(GIL)?
- 120. How do you prevent dead-lock?

- 121. What is the benefit of python multi-threading in jython?
- 122. Explain below terms,
 - a. Critical section?
 - b. What is dead-lock ?
 - c. What is live-lock?

Modules –Packages

- 123. What is a Python module?
- 124. How do you import all functions/classes in a module?
- 125. How do u implement import*
- 126. When do you get recursive import error, and how do you resolve it?
- 127. Explain how can you access a module written in Python from C?
- 128. Name few Python modules for Statistical, Numerical and scientific computations?
- 129. What is package in Python?
- 130. What is the difference between package and directory?
- 131. What is the purpose of __init__.py in Python?
- 132. What is __all__?
- 133. What is the purpose of help, dir functions?
- 134. What is the purpose of __name__?
- 135. Private variables/ functions in a module ?

Exception Handling

- 136. What is the purpose of Exception Handling? How do you achieve it in Python?
- 137. What are the different types of exceptions generated in python?
- 138. Why does the exception handling have a finally block?
- 139. How **finally** and **else** works ?
- 140. Can you have multiple except blocks? In which order we have to arrange **except** blocks?

Unit test

- 141. What is the purpose of unit testing?
- 142. What you should not test?
- 143. What is mocking?
- 144. How do you mock exceptions?
- 145. How do you write an unit test for a function which has infinity loop?
- 146. What are the python modules required for unit testing?
- 147. How do you mock a function?

Miscellaneous

- 148. What are the various logging levels in python ?
- 149. How do you create rotating file handler in python?
- 150. What is ORM?
- 151. What is virtualenv?
- 152. What is serialization ? difference between pickle.dumps() vs pickle.dump?
- 153. Difference between **search()**, **match()**, **findall()** in **re** module
- 154. Explain **greedy-matching** in **re** module?
- 155. Explain, broadcasting in **numpy** arrays ?