**Python Technical Interview Prep (4 Pages)**

[**https://www.linkedin.com/feed/update/urn:li:activity:6543841130511945728/**](https://www.linkedin.com/feed/update/urn:li:activity:6543841130511945728/)

* Concepts
* Abstract class
* Collections
* OOPS
* when to use notify and when to use notify all
* Hash Tables/Maps
* Databases
* Concurrency
* Data structures
* Binary trees
* Queries
* Initializing
* Thread synchronization.
* Daemon threads
* Static methods
* SQL Queries
* Multi-threading
* Distributed Computing
* Performance, Data Structuring etc.
* Here are the questions they asked (please note some questions where asked to java candidates so if it doesn’t match your background, disregard):
* Probability
* Synchronize HashMap
* Garbage Collection
* Definition of key word in synchronization
* Arraylist vs Linkedlist
* Networks: what's the difference between TCP and UDP?
* How does a HashMap work?
  + How to make sure a hashmap is threadsafe if multiple readers and writers have access to it?

1. **What are the key features of Python?**
   1. Easy to learn and use
   2. Dynamic typing
   3. Dynamic memory allocation
   4. Provides backend support in full stack development
   5. GUI support
   6. Object oriented programming, abstraction, etc.
   7. Integrated language, lines execute one-by-one. No need to compile like C and C++ so it’s portable as well
2. **Mention five benefits of using Python?**
   1. Extensive list of libraries for data analysis like pandas, numpy, scikit learn, matplotlib, seaborn
   2. Machine learning libraries like keras, tensorflow, etc.
   3. Web scraping
   4. Web applications
   5. Memory management, and typing
   6. GUI support – tkinter
3. **Mention the use of the split function in Python?**
   1. It takes a delimiter and splits a string into a list of substrings breaking on that delimiter. Default is space.
4. **Compare Java & Python**
   1. Python:
      1. Easy to use and read
      2. Slower as it’s dynamically typed
      3. Used for scripting, large-scale data processing, data science, machine learning
      4. Good for agile development
   2. Java
      1. Syntax is a bit strict, Requiring parenthesis for conditions and semicolons
      2. Faster than python
      3. Support for multi-threading
      4. Mostly used for android development
      5. Self-memory management
5. **What are the supported data types in Python?**
   1. Int
   2. Float
   3. complex
   4. str
   5. bool
   6. dict
   7. tuple
   8. list
   9. None
6. **How is memory managed in Python?**
   1. Stack memory
      1. All function calls, variable initializations are stored in a stack temporarily and are cleared when the function returns
   2. Heap memory
      1. Shared variables especially global ones are stored in this memory
   3. Garbage collector
      1. Python uses reference count to keep a track of all objects that are no longer used and removes them and frees the memory
7. **Whenever Python exits, why isn’t all the memory de-allocated?**
   1. CPython is the program that executes python, and sometimes that allocates some memory which the garbage collector cannot free.
   2. It’s because some variables which are shared and accessed by multiple objects. Cache could also be in use when python is exiting so it’s not clearer.
   3. Atexit
8. **What is dictionary in Python?**
   1. It’s a data structure type in python where data is stored as keys and values, and they are mutable. Keys should be unique and are not mutable, values are.
9. **What is a Python module?**
   1. It’s a .py file that has python functions, classes which can be imported to a different file, and easy to use across multiple files.
10. **How do you perform pattern matching in Python**
    1. You can use the keyword in to see if a substring is in a string. Or a value is in a list, etc. a string variable also has functions like startswith and endswith for pattern matching the first and last n characters
    2. You can use the regex library ‘re’ for pattern matching
11. **What is the difference between list and tuples?**
    1. Lists and tuples are both arrays, but lists are mutable so they are slow, and tuples are more efficient and fast since they aren’t. tuples should be used in cases where values are static.
12. **What is the difference between deep and shallow copy?**
    1. Shallow copy makes a new copy of that object but doesn’t copy the values. It only copies the reference to the value so when the new copy is changed, the old changes as well. In a nested-lists, the objects of the inner lists share references, so the change in one impacts another.
    2. Deep copy is when 2 objects are completely independent of each other and share no references at all.
13. **How is Multithreading achieved in Python?**
    1. Multithreading is when we want multiple processes to run simultaneously. Python has an inbuilt library called threading. You can define a function, and pass it to a thread. You can use thread.start() to run the function, and thread.join() to wait for the function to finish.
14. **How can the ternary operators be used in python?**
    1. Ternary operators mean conditionals. In python, we use if, else statements to define conditions.
    2. If condition is met: return true else return false
    3. Bool = true if condition is met else false
15. **Explain Inheritance in Python with an example.**
    1. Inheritance is an object oriented programming mechanism, that allows us to inherit properties of a class into a different class, so reduce redundant code.
    2. The most basic example would be types of animals. We define an interface class called Animal, and add functions like name, sound, legs.
    3. Now, we can create classes like Cat, dog, cow, bunny and all of them should inherit from the class animals because they are considered animals, and all the functions in the animal class would be available to use in these subclasses.
16. **Explain what Flask is and its benefits?**
    1. Flask is a web framework that can be used with python to create web applications. It’s simple to use, so it can be used by beginners as well as experienced. We can add a lot of different components to flask so it can handle complex applications as well.
17. **What is the usage of help() and dir() function in Python?**
    1. The help function can be called on python functions, and it prints out an explanation of how to use that function.
    2. The dir function prints out a list of attributes that can be called on an object without the explanations.
18. **What is monkey patching in Python?**
    1. Monkey patching is when we try to fix a bug in the code in runtime. This is used for dynamic changing of the code without modifying the source code so the code can run well in production for time-being. Later the monkey-patch is fixed and tested before deploying.
19. **What does this mean: \*args, \*\*kwargs? And why would someone use it?**
    1. This allows passing of n number of parameters.
    2. When you pass in parameters as \*args, they are stored in a tuple. One \* = tuple
    3. When you pass in parameters as \*\*kwargs, they are stored as dictionary. Two \*\* = dict
    4. Someone would use this when they want to pass in many parameters, and when we don’t know how many parameters we need. Example would be returning a sum of all variables but don’t know how many.
20. **Write a one-liner that will count the number of capital letters in a file. Your code should work even if the file is too big to fit in memory.**
    1. sum(1 for line in open(filename, 'r') for char in line if char.isupper())
21. **What are negative indexes and why are they used?**
    1. Negative indices are used to access the last n values of a list, tuple, string
    2. We can use them to access the last n values without calling the length of the data structure, making it efficient.
22. **How can you randomize the items of a list in place in Python?**
    1. Using the random library, random.shuffle(list)
23. **What is the process of compilation and linking in python?**
    1. The source code of python is compiled into a format known as byte-code, and byte-code is then converted into machine code. That is stored in a .pyc file, which is also called CPython.
    2. Dynamic linking occurs at runtime, because python is a interpreter language and it loads the bytecode line, by line and executes it.
24. Write a sorting algorithm for a numerical dataset in Python.

def merge\_sort(arr):

arr\_len = len(arr)

# base case

if arr\_len <=1:

return

mid = arr\_len//2

left\_arr = arr[:mid]

right\_arr = arr[mid:]

merge\_sort(left\_arr)

merge\_sort(right\_arr)

print(f"Merging left {left\_arr} and right {right\_arr}")

return merge(left\_arr, right\_arr, arr)

def merge(left, right, arr):

sorted\_arr = arr

i = j = k = 0

while i < len(left) and j < len(right):

if left[i] < right[j]:

sorted\_arr[k] = left[i]

i += 1

else:

sorted\_arr[k] = right[j]

j += 1

k += 1

while i < len(left):

sorted\_arr[k] = left[i]

i+=1

k+=1

while j < len(right):

sorted\_arr[k] = right[j]

j+=1

k+=1

return sorted\_arr

arr = [1,2,3,4,5,6,7,8]

merge\_sort(arr)

1. Explain split(), sub(), subn() methods of “re” module in Python.
2. How can you generate random numbers in Python?
3. What is the difference between range & xrange?
4. What is pickling and unpickling?
5. Mention the differences between Django, Pyramid and Flask.
6. Discuss the Django architecture.
7. Explain how you can set up the Database in Django.
8. Give an example how you can write a VIEW in Django?
9. Mention what the Django templates consists of.
10. Explain the use of session in Django framework?
11. List out the inheritance styles in Django.
12. What is map function in Python?
13. How to get indices of N maximum values in a NumPy array?
14. How do you calculate percentiles with Python/ NumPy?
15. What advantages do NumPy arrays offer over (nested) Python lists?
16. Explain the use of decorators.
17. What is the difference between NumPy and SciPy?
18. How do you make 3D plots/visualizations using NumPy/SciPy?
19. Which of the following statements create a dictionary? (Multiple Correct Answers Possible)
20. Which one of these is floor division?
21. What is the maximum possible length of an identifier?
22. Why are local variable names beginning with an underscore discouraged?
23. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1] ?
24. To open a file c:\scores.txt for writing, we use
25. What is the output of the following?
26. When will the else part of try-except-else be executed?
27. Write a sorting algorithm for a numerical dataset in Python.
28. How will you reverse a list?
29. How will you remove last object from a list?
30. What are the built-in type does python provides
31. What is namespace in Python?
32. How you can convert a number to a string?
33. You are having multiple Memcache servers running Python, in which one of the memcacher server fails, and it has your data, will it ever try to get key data from that one failed server?
34. Explain how you can minimize the Memcached server outages in your Python Development?
35. Explain how Memcached should not be used in your Python project?
36. Explain what is Dogpile effect? How can you prevent this effect
37. What is an algorithm?
38. What is time complexity of Binary Search?
39. Can Binary Search be used for linked lists?
40. How to find if two given rectangles overlap?
41. How to find angle between hour and minute hands at a given time?
42. When does the worst case of QuickSort occur?
43. Given a big string of characters, how to efficiently find the first unique character in it?
44. How to count inversions in a sorted array?
45. Given a big array, how to efficiently find k’th largest element in it?
46. Given an array of size n with range of numbers from 1 to n+1. The array doesn’t contain any duplicate, one number is missing, find the missing number.
47. How to write an efficient method to calculate x raise to the power n?
48. Given an input string and a dictionary of words, find out if the input string can be segmented into a space-separated sequence of dictionary words.
49. You are given an array of sorted words in an arbitrary language, you need to find order (or precedence of characters) in the language.
50. What is a Data Structure?
    1. What are linear and non linear data Structures?
51. What are the various operations that can be performed on different Data Structures?
52. How is an Array different from Linked List?
53. What is Stack and where it can be used?
54. What is a Queue, how it is different from stack and how is it implemented?
55. What are Infix, prefix, Postfix notations?
56. What is a Linked List and What are its types?
57. Which data structures are used for BFS and DFS of a graph?
58. Can doubly linked be implemented using a single pointer variable in every node?
59. How to implement a stack using queue?
60. How to implement a queue using stack?
61. Which Data Structure Should be used for implementiong LRU cache?
62. How to check if a given Binary Tree is BST or not?
63. Linked List Questions
64. Tree Traversal Questions
65. Convert a DLL to Binary Tree in-place
66. Convert Binary Tree to DLL in-place
67. Delete a given node in a singly linked list
68. Reverse a Linked List
69. Detect Loop in a Linked List
70. Which data structure is used for dictionary and spell checker?
71. Largest Number formed in the array
72. Find the length of maximum numbers of consecutive numbers jumped up in an array

How is python executed?

All code compiles down to a bytecode, which is the .pyc file, and run line by line inside a python VM