## Main Ideas of CPython’s Garbage Collector

1. Maintain reference count. For every object, there is a count of the total number of references to that object. If that count ever falls to 0, then you can immediately deallocate that object because it is no longer live.
2. Periodically detect reference cycles. Deallocating when the reference count falls to 0 doesn’t work for all cases. Consider two objects A and B, where A holds a reference to B and B holds a reference to A. This is called a reference cycle. It could be the case that these are no longer live and so that both A and B should be garbage collected. However, the reference count on both objects are not zero, so they remain alive. To get around this, CPython uses an algorithm for detecting reference cycles and deallocating objects in the cycle.
3. Performance is enhanced with heuristics. Objects that have been created recently are more likely to need to be garbage collected. CPython introduces the concept of a generation to account for the relative age of an object. Younger generations have objects that have more recently been created and older generations hold objects that are less recent. Each object belongs to exactly one generation. When garbage collection is performed, CPython tries to garbage collect younger generations. Periodically, CPython will perform garbage collection on older generations (the rate at which this happens is determined by a heuristic).

**Q.2. What is a docstring?**

A docstring is a documentation string that we use to explain what a construct does. We place it as the first thing under a function, class, or a method, to describe what it does. We declare a docstring using three sets of single or double quotes.

1. >>> **def** sayhi():
2. """

This function prints Hi

1. """
2. print("Hi")
3. >>> sayhi()

Hi

To get a function’s docstring, we use its \_\_doc\_\_ attribute.

1. >>> sayhi.\_\_doc\_\_

**Q.3. What is the PYTHONPATH variable?**

PYTHONPATH is the variable that tells the interpreter where to locate the module files imported into a program. Hence, it must include the Python source library directory and the directories containing Python source code. You can manually set PYTHONPATH, but usually, the Python installer will preset it.

**Q.4 Differentiate between deep and shallow copy.**

A deep copy copies an object into another. This means that if you make a change to a copy of an object, it won’t affect the original object. In Python, we use the function deepcopy() for this, and we import the module copy. We use it like:

1. >>> **import** copy
2. >>> b=copy.deepcopy(a)

A shallow copy, however, copies one object’s reference to another. So, if we make a change in the copy, it will affect the original object. For this, we have the function copy(). We use it like:

1. >>> b=copy.copy(a)

**Q.5. Differentiate between lists and tuples.**

The major difference is that a list is mutable, but a tuple is immutable. Examples:

1. >>> mylist=[1,3,3]
2. >>> mylist[1]=2
3. >>> mytuple=(1,3,3)
4. >>> mytuple[1]=2
5. Traceback (most recent call **last**):
6. File "<pyshell#97>", line 1, **in** <module>
7. mytuple[1]=2

TypeError: ‘tuple’ object does not support item assignment

**Q.6. How would you work with numbers other than those in the decimal number system?**

With Python, it is possible to type numbers in binary, octal, and hexadecimal. Binary numbers are made of 0 and 1. To type in binary, we use the prefix 0b or 0B.

1. >>> **int**(0b1010)

10

To convert a number into its binary form, we use bin().

1. >>> bin(0xf)

‘0b1111’

Octal numbers may have digits from 0 to 7. We use the prefix 0o or 0O.

1. >>> oct(8)

‘0o10’

Hexadecimal numbers may have digits from 0 to 15. We use the prefix 0x or 0X.

1. >>> hex(16)

‘0x10’

1. >>> hex(15)

‘0xf’

**Q.7. Explain the //, %, and \*\* operators in Python.**

The // operator performs floor division. It will return the integer part of the result on division.

1. >>> 7//2

3

Normal division would return 3.5 here.

Similarly, \*\* performs exponentiation. a\*\*b returns the value of a raised to the power b.

1. >>> 2\*\*10

1024

Finally, % is for modulus. This gives us the value left after the highest achievable division.

1. >>> 13%7

6

1. >>> 3.5%1.5

0.5

**Q.8. What do you mean by \*args and \*\*kwargs?**

In cases when we don’t know how many arguments will be passed to a function, like when we want to pass a list or a tuple of values, we use \*args.

1. >>> **def** func(\*args):
2. **for** i **in** args:
3. **print**(i)
4. >>> func(3,2,1,4,7)

3

2

1

4

7

\*\*kwargs takes keyword arguments when we don’t know how many there will be.

1. >>> **def** func(\*\*kwargs):
2. **for** i **in** kwargs:
3. **print**(i,kwargs[i])
4. >>> func(a=1,b=2,c=7)

a.1

b.2

c.7

The words args and kwargs are convention, and we can use anything in their place.

**Q.9. What are negative indices?**

Let’s take a list for this.

1. >>> mylist=[0,1,2,3,4,5,6,7,8]

A negative index, unlike a positive one, begins searching from the right.

1. >>> mylist[-3]

6

This also helps with slicing from the back:

1. >>> mylist[-6:-1]

[3, 4, 5, 6, 7]

**Q.10. How do you get a list of all the keys in a dictionary?**

For this, we use the function keys().

1. >>> mydict={'a':1,'b':2,'c':3,'e':5}
2. >>> mydict.keys()
3. dict\_keys(['a', 'b', 'c', 'e'])

**1. What is Python?**

Python language is used in scripting and also it is a dynamic, interpreted [programming language](http://besttoppers.com/top-10-programming-languages/). It allows the programmer to express concepts in fewer lines of code than possible in languages such as Java or C++.

**2. List the features of Python?**

* Python provides very high-level dynamic data types
* Python supports automatic garbage collection.
* Python can be easily integrated with C, C++, Java, ActiveX, COM, and CORBA.
* Python supports functional and structured programming methods as well as OOP.
* Python can be compiled to byte-code for building large applications.

**3. Explain Python dictionaries?**

It is a kind of hash table type. They consist of key-value pairs and a dictionary key can be of any Python type but usually strings or numbers.

**4. How to create a dictionary in Python?**

The dictionary is enclosed by {} -curly braces and Square brackets are used to assign and access the values.

**5. How to convert string to float in Python?**

To convert string to float in a python float is used before the string.

**Ex: float (x) – X** will be the string which will be converted into a floating point variable.

**6. How will you reverse the list in Python?**

To Reverse object of list in place**list. reverses ()** statement is used.

**7. What are the rules for global and local variable in python?**

If a variable is assigned to a new value inside the function, then it is local and if a variable is defined outside a function then it is globally implicit.

**8. What is a module in Python?**

The module is simply a runnable code in python. It is a Python object with names, attributes that can be used for the purpose of bind and reference. The module can define variables, functions and classes.

**9. What is scope in Python?**

Scope is a region in Python program where it could be used without any qualification. i.e when the unqualified referenced with a name, then it can be looked out in the namespace to find the object.

**10. What are Tuple in Python?**

Tuples are enclosed within the parentheses. It is another sequence of data type that is similar to the list. The values of the tuples are separated by commas.

**11. What is lambda in Python?**

It is an anonymous function often used as an inline function. General form of lambda is

1

lambda arg1, arg2 arg3 arg4... :

**12. What is a package in Python?**

A packages in Python can have sub folders and modules. The package is imported by using import package Statement.

**Ex: import folder2.subfolder3.module2**

**13. Given the subclass of dictionary:**

1

2

3

4

class DefaultDict(dict):

def \_\_missing\_\_(owned, key):

return []

**Will the code below work? Explain why or why not?**

1

2

3

d = DefaultDict()

d['plora'] = 156:

Yes, it works. Because whenever the key is missing the dictionary instance, will be automatically be instantiated with a list.

**14. How to perform unit testing in Python?**

A unit testing framework called unittest is provided by Python. A unittest module in Python supports automation testing, shutdown code for tests and sharing of setup, independence of the tests from the reporting framework. and aggregation of tests into collections.

Q 1: You obviously know the detailed definition for python. But my question is compare python with other technologies>

* Python is an Interpreted language. Unlike other languages like C and it’s variant, it doesn’t needed to be compiled before run. Ruby and PHP are also Interpreted language.
* Python have first class object.that means it can be assigned to variables returned to other function and passed to other functions.
* Writing a Python language is quick.But the problem is the compilation time is slower compared to other languages.

Q 2: What about python and multi threading? Is it a good idea? List some methods to run python code in a parellel way.

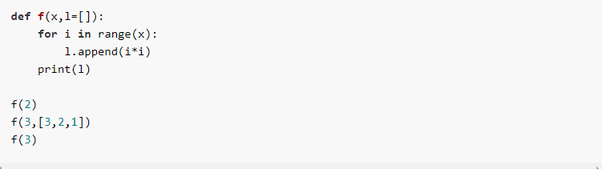
* Python doesn’t allow multi threading .It do have a multi threading package if you want to speed up the code.but it’s not a pretty good idea to do so.
* Python has a construct called [Global interpret lock](https://www.probytes.net/blog/python-global-interpreter-lock/). GIL is responsible for sending only one thread at a time for execution.A thread acquires a thread,do some work and passes,the GIL acquires the next thread and the process goes on and on. This process happens with in a blink of an eye that we feels the execution is happening in a parallel way.

Also read : [Python development trends 2018](https://www.probytes.net/blog/python-development-trends-2018/)

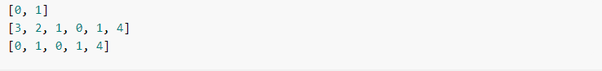
Q 3: How do you keep track on different versions of your code?

* Version control. At this point You should be act excited and tell them how you use GIL(or your favorite version). Because code without version control is like coffee without a cup. Sometimes we need to write once-off throw away scripts and that's ok, but if you are dealing with any significant amount of code, a version control system will be a benefit.

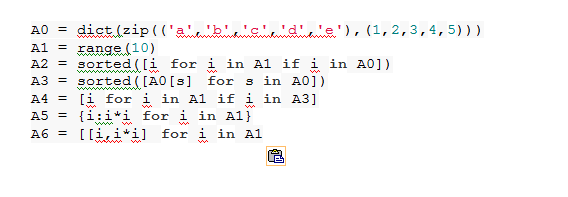
Q 4:What does this code output?



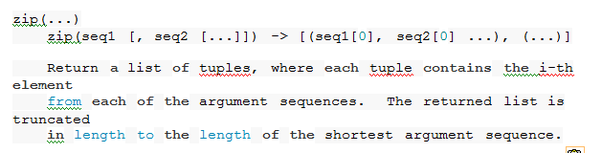
ans:



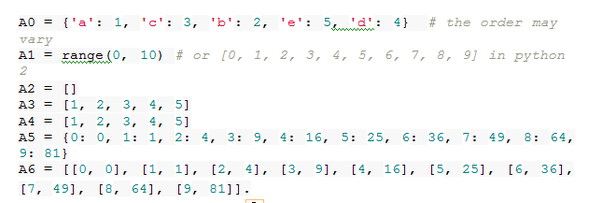
Q5: Looking at the below code, write down the final values of A0, A1, ...An.



If you dont know what zip is don't stress out. No sane employer will expect you to memorize the standard library. Here is the output of help(zip).



Answer:



Q6: What is monkey patching? Is it a good idea?

Ans: Monkey patching change the behavior of a function or a class which is already defined.

https://qph.fs.quoracdn.net/main-qimg-079fa9dbbdf8c55693a30c8d78e893ff

I hope the detailing I gave you about the Interview questions related to python will help you on your Upcoming Interviews. One thing You guys always have to keep in mind is that nobody is perfect. It is Ok to skip the answers that you don’t know in the Interview. Work Hard, Have belief in yourself and Go,Get it Done

1. Write a python script to count frequency of words, what data structure would you use?

2. In two lines write a code to find max and min in a list of numbers, what functions would you use?

3. What are current versions of Python? Explain their differences

4. What libraries do you use?

5. How do you isolate different environment with Python?

6. Where should you not use Python?

7. What are some of standard modules you typically use?

**Medium**

1. What is collections module, what are some of real world applications?

2. Can you write a script that extracts phone numbers {US format} from collection of text files?

3. Can you write a script that iterates on values of dictionary in descending manner?

4. What are list comprehensions, when should you avoid using them?

5. What are different implementations of Python?

6. Can you write a simple crawler in Python?

7. What modules do you use for interacting with DB in Python?

**Difficult**

1. How would you deal with dataset that doesn't fit onto memory? {E.g. Wordcounting over a really big text file}

2. How would you make your scripts run on multiple machines using Python?

3. How do you optimize your Python code?

4. How do you typically organize your code?

5. How would you extend Python with C/C++?

6. How does Python's garbage collection work?

7. What is GIL?

[CareerCup](http://www.careercup.com/) is a good resource it has forums where people post their interview questions inline with your list of companies that you want to get into. Also I'd recommend reading Cracking Coding Interview from the same website. The least common denominator for interviews at these companies would include

* Knowledge about algorithms
* Ability to optimize code
* Ability to express/communicate your solution (I think this would be more important

These are questions are asked by the interviewer on Python Programming language.

1. What is Python?

2. Is python a case-sensitive language?

3. What is PEP 8?

4. How are the functions help() and dir() different?

5. What are the tools that help to find bugs or perform the static analysis?

6. How are arguments passed by value or by reference?

7. What is the built-in type does python provides?

8. What is lambda in Python?

9. In Python what are iterators?

10. What is doctrine in Python

11. What is the difference between X range and range?

12. Explain how Python does Compile-time and Run-time code checking?

13. As Everything in Python is an Object, Explain the characteristics of Python's Objects.

14. Explain how can you make a Python Script executable on Unix?

There is a large number of elementary algorithmic interview questions you can use with any language (memory addressing, networking, trees, hash tables, sorts, concurrency, etc), but if you're looking for Python-specific questions, here are a few, in no particular order of difficulty or relevancy:

* Talk to me about the GIL. How does it impact concurrency in Python? What kinds of applications does it impact more than others?
* How does Python's garbage collection work?
* What is the difference between range and xrange? How has this changed over time?
* Here's a function. Optimize it for me.
* How do you iterate over a list and pull element indices at the same time?
* I'm getting a maximum recursion depth error for a function. What does this mean? How can I mitigate the problem?
* How do you enforce ordering for a dictionary-style object?
* How does Python's list.sort work at a high level? Is it stable? What's the runtime?
* What's the difference between a list, dictionary, and array in Python?
* What does this list comprehension do?
* Here's a class hierarchy with some methods defined. When I call this function, what gets printed?
* What is monkeypatching? How can you do it in Python?
* What are some caveats to pickling? Marshaling?
* How many ways can you append or concatenate strings? Which of these ways is fastest? Easiest to read?
* Print me the full pathname of every file in this directory tree.
* What's wrong with this function?
* What's your preferred editor? (vim, of course - anything else and they fail.)

What is Python?

Name some of the features of Python.

What is the purpose of PYTHONPATH environment variable?

What is the purpose of PYTHONSTARTUP environment variable?

What is the purpose of PYTHONCASEOK environment variable?

What is the purpose of PYTHONHOME environment variable?

Is python a case sensitive language?

What are the supported data types in Python?

What is the output of print str if str = 'Hello World!'?

What is the output of print str[0] if str = 'Hello World!'?

What is the output of print str[2:5] if str = 'Hello World!'?

What is the output of print str[2:] if str = 'Hello World!'?

What is the output of print str \* 2 if str = 'Hello World!'?

What is the output of print str + "TEST" if str = 'Hello World!'?

What is the output of print list if list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]?

What is the output of print list[0] if list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]?

What is the output of print list[1:3] if list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]?

What is the output of print list[2:] if list = [ 'abcd', 786 , 2.23, 'john', 70.2 ]?

What is the output of print tinylist \* 2 if tinylist = [123, 'john']?

What is the output of print list + tinylist \* 2 if list = [ 'abcd', 786 , 2.23, 'john', 70.2 ] and tinylist = [123, 'john']?

What are tuples in Python?

What is the difference between tuples and lists in Python?

What is the output of print tuple if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )?

It will print complete tuple. Output would be ('abcd', 786, 2.23, 'john', 70.200000000000003).

What is the output of print tuple[0] if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )?

What is the output of print tuple[1:3] if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )?

What is the output of print tuple[2:] if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )?

What is the output of print tinytuple \* 2 if tinytuple = (123, 'john')?

What is the output of print tuple + tinytuple if tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 ) and tinytuple = (123, 'john')?

What are Python's dictionaries?

How will you create a dictionary in python?

How will you get all the keys from the dictionary?

How will you get all the values from the dictionary?

How will you convert a string to an int in python?

How will you convert a string to a long in python?

How will you convert a string to a float in python?

How will you convert a object to a string in python?

How will you convert a object to a regular expression in python?

How will you convert a String to an object in python?

How will you convert a string to a tuple in python?

How will you convert a string to a list in python?

How will you convert a string to a set in python?

How will you create a dictionary using tuples in python?

How will you convert a string to a frozen set in python?

How will you convert an integer to a character in python?

How will you convert an integer to an unicode character in python?

How will you convert a single character to its integer value in python?

How will you convert an integer to hexadecimal string in python?

How will you convert an integer to octal string in python?

What is the purpose of \*\* operator?

What is the purpose of // operator?

What is the purpose of is operator?

What is the purpose of not in operator?

What is the purpose break statement in python?

What is the purpose continue statement in python?

What is the purpose pass statement in python?

How can you pick a random item from a list or tuple?

How can you pick a random item from a range?

**2. Know about the company:**

Spend time to know about the company’s background and various activities of the company. Knowledge about the company will make you look serious for the job. Also if

**Q1).What is**[**Python**](https://www.mytectra.com/interview-question/python-real-time-interview-questions-and-answers/)[[1]](https://www.quora.com/What-are-good-Python-interview-questions#VXmxx)**?**  
**Ans1:** [Python](https://www.mytectra.com/interview-question/python-real-time-interview-questions-and-answers/) is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it h  
as fewer syntactical constructions than other languages.

**Q2).Name some of the features of**[**Python**](https://www.mytectra.com/interview-question/python-real-time-interview-questions-and-answers/)**?**  
**Ans2:** Following are some of the salient features of python

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**Q3).Do you have any personal projects?**  
**Really?**  
**Ans3:**This shows that you are willing to do more than the bare minimum in terms of keeping your skillset up to date. If you work on personal projects and code outside of the workplace then employers are more likely to see you as an asset that will grow. Even if they don’t ask this question I find it’s useful to broach the subject.

**Q4).Is**[**python**](https://www.mytectra.com/interview-question/python-real-time-interview-questions-and-answers/)**a case sensitive language?**  
**Ans4:** Yes! Python is a case sensitive programming language.

What are the supported data types in Python?  
Python has five standard data types −

* Numbers
* String
* List
* Tuple
* Dictionary

**Q5).What is the output of print str if str = ‘Hello World!’?**  
**Ans5:** It will print complete string. Output would be Hello World!.

**Q6).What is the output of print str[0] if str = ‘Hello World!’?**  
**Ans6**: It will print first character of the string. Output would be H.

**Q7).What is the output of print str[2:5] if str = ‘Hello World!’?**  
**Ans7:** It will print characters starting from 3rd to 5th. Output would be llo.  
**Q8).What is the output of print str[2:] if str = ‘Hello World!’?**  
**Ans8:** It will print characters starting from 3rd character. Output would be llo World!.  
**Q9).What is the output of print str \* 2 if str = ‘Hello World!’?**  
**Ans9:** It will print string two times. Output would be Hello World!Hello World!.  
**Q10).What is the output of print str + “TEST” if str = ‘Hello World!’?**  
**Ans10:** It will print concatenated string. Output would be Hello World!TEST.

**Q11).What is the output of print list if list = [ ‘abcd’, 786 , 2.23, ‘john’, 70.2 ]?**  
**Ans11:** It will print concatenated lists. Output would be [ ‘abcd’, 786 , 2.23, ‘john’, 70.2 ].

**Q12).What is the output of print list[0] if list = [ ‘abcd’, 786 , 2.23, ‘john’, 70.2 ]?**  
**Ans12:** It will print first element of the list. Output would be abcd.

**Q13).What is the output of print list[1:3] if list = [ ‘abcd’, 786 , 2.23, ‘john’, 70.2 ]?**  
**Ans13:** It will print elements starting from 2nd till 3rd. Output would be [786, 2.23].

**Q14).What is the output of print list[2:] if list = [ ‘abcd’, 786 , 2.23, ‘john’, 70.2 ]?**  
**Ans14:** It will print elements starting from 3rd element. Output would be [2.23, ‘john’, 70.200000000000003].

**Q15).What is the output of print tinylist \* 2 if tinylist = [123, ‘john’]?**  
**Ans15:** It will print list two times. Output would be [123, ‘john’, 123, ‘john’].

**Q16).What is the output of print list + tinylist \* 2 if list = [ ‘abcd’, 786 , 2.23, ‘john’, 70.2 ] and tinylist = [123, ‘john’]?**  
**Ans16:** It will print concatenated lists. Output would be [‘abcd’, 786, 2.23, ‘john’, 70.2, 123, ‘john’, 123, ‘john’].

**Q17).What is tuples in Python?**  
**Ans17:** A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

**Q18).What is the difference between tuples and lists in Python?**  
**Ans18:**The main differences between lists and tuples are − Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated. Tuples can be thought of as read-only lists.

**Q19).What is the output of print tuple if tuple = ( ‘abcd’, 786 , 2.23, ‘john’, 70.2 )?**  
**Ans19:**It will print complete tuple. Output would be (‘abcd’, 786, 2.23, ‘john’, 70.200000000000003).

**Q20).What is the output of print tuple[0] if tuple = ( ‘abcd’, 786 , 2.23, ‘john’, 70.2 )?**  
**Ans20:** It will print first element of the tuple. Output would be abcd.

1. What is Python really? You can (and are encouraged) make comparisons to other technologies in your answer

2. Python and multi-threading. Is it a good idea? List some ways to get some Python code to run in a parallel way.

3. How do you keep track of different versions of your code?

4. Describe Python's garbage collection mechanism in brief.

5. What is monkey patching and is it ever a good idea?

6 . What does this stuff mean: \*args, \*\*kwargs? And why would we use it?

7. What do these mean to you: @classmethod, @staticmethod, @property?

8. What are tuples in Python?

9. What is the difference between tuples and lists in Python?

10. What are Python's dictionaries?

11. how will you convert a object to a string in python?

12. What is the purpose continue statement in python?

13. How can you get a random number in python?

14. What is the difference between del() and remove() methods of list?

15. How will you get the min valued item of a list?

16. How can you pick a random item from a range?

17. What is the purpose pass statement in python?

18. How will you merge elements in a sequence?

19. How will you get the length of the string?

20. How will you remove all leading whitespace in string?

* [What are some good embedded interview questions?](https://www.quora.com/What-are-some-good-embedded-interview-questions)
* [What are some good interview question-related financial derivatives?](https://www.quora.com/What-are-some-good-interview-question-related-financial-derivatives)
* [Is python a good choice for coding interview?](https://www.quora.com/Is-python-a-good-choice-for-coding-interview)
* [What is the best interview question ever?](https://www.quora.com/What-is-the-best-interview-question-ever)
* [What are some good python scripting based questions for an IP Network Engineer interview?](https://www.quora.com/What-are-some-good-python-scripting-based-questions-for-an-IP-Network-Engineer-interview)
* [What is the best question you can ask the interviewer at the end of the interview?](https://www.quora.com/What-is-the-best-question-you-can-ask-the-interviewer-at-the-end-of-the-interview)