```
 Python Interview Questions and Answers by Pythonlife



1. What is Python? What are the benefits of using Python?
Ans: Python is a programming language with objects, modules, threads, exceptions and automatic




2. What is PEP 8?
Ans: PEP 8 is a coding convention, a set of recommendation, about how to write your Python code




3. What is pickling and unpickling?
Ans: Pickle module accepts any Python object and converts it into a string representation and c




4. How Python is interpreted?
Ans: Python language is an interpreted language. Python program runs directly from the source c
```

5. How memory is managed in Python?
Ans: Python memory is managed by Python private heap space. All Python objects and data structu
The allocation of Python heap space for Python objects is done by Python memory manager. The co
Python also have an inbuilt garbage collector, which recycle all the unused memory and frees th

6. What are the tools that help to find bugs or perform static analysis?
Ans: PyChecker is a static analysis tool that detects the bugs in Python source code and warns

7. What are Python decorators?
Ans: A Python decorator is a specific change that we make in Python syntax to alter functions e

8. What is the difference between list and tuple?
Ans: The difference between list and tuple is that list is mutable while tuple is not. Tuple ca

9. How are arguments passed by value or by reference?
Ans: Everything in Python is an object and all variables hold references to the objects. The re

10. What is Dict and List comprehensions are?
Ans: They are syntax constructions to ease the creation of a Dictionary or List based on existi

11. What are the built-in type does python provides?
Ans: There are mutable and Immutable types of Pythons built in types Mutable built-in types
List Sets
Dictionaries Immutable built-in types
Strings Tuples Numbers

12. What is namespace in Python?
Ans: In Python, every name introduced has a place where it lives and can be hooked for. This is

13. What is lambda in Python?
Ans: It is a single expression anonymous function often used as In-line function.

14. Why lambda forms in python does not have statements?
Ans: A lambda form in python does not have statements as it is used to make new function object

15. What is pass in Python?
Ans: Pass means, no-operation Python statement, or in other words it is a place holder in compo

16. In Python what are iterators?
Ans: In Python, iterators are used to iterate a group of elements, containers like list.

17. What is unit test in Python?
Ans: A unit testing framework in Python is known as unittest. It supports sharing of setups, au

18. In Python what is slicing?
Ans: A mechanism to select a range of items from sequence types like list, tuple, strings etc.

19. What are generators in Python?
Ans: The way of implementing iterators are known as generators. It is a normal function except

20. What is docstring in Python?
Ans: A Python documentation string is known as docstring, it is a way of documenting Python fur

21. How can you copy an object in Python?
Ans: To copy an object in Python, you can try copy.copy () or copy.deepcopy() for the general c

22. What is negative index in Python?
Ans: Python sequences can be index in positive and negative numbers. For positive index, 0 is t

23. How you can convert a number to a string?
Ans: In order to convert a number into a string, use the inbuilt function str(). If you want a

24. What is the difference between Xrange and range?
Ans: Xrange returns the xrange object while range returns the list, and uses the same memory ar

25. What is module and package in Python?
Ans: In Python, module is the way to structure program. Each Python program file is a module, w
The folder of Python program is a package of modules. A package can have modules or subfolders

26. Mention what are the rules for local and global variables in Python?
Ans: Local variables: If a variable is assigned a new value anywhere within the function’s body

Global variables: Those variables that are only referenced inside a function are implicitly glo

27. How can you share global variables across modules?
Ans: To share global variables across modules within a single program, create a special module

28. Explain how can you make a Python Script executable on Unix?To make a Python Script executa
Ans: Script file’s mode must be executable and
the first line must begin with # (#!/usr/local/bin/python)

29. Explain how to delete a file in Python?
Ans: By using a command os.remove (filename) or os.unlink(filename)

30. Explain how can you generate random numbers in Python?
Ans: To generate random numbers in Python, you need to import command as import random
random.random()
This returns a random floating point number in the range [0,1)

31. Explain how can you access a module written in Python from C?
Ans: You can access a module written in Python from C by following method, Module = =PyImport_I

32. Mention the use of // operator in Python?
Ans: It is a Floor Divisionoperator , which is used for dividing two operands with the result a

33. Mention five benefits of using Python?
Ans: Python comprises of a huge standard library for most Internet platforms like Email, HTML,
Python does not require explicit memory management as the interpreter itself allocates the memo
Provide easy readability due to use of square brackets Easy-to-learn for beginners
Having the built-in data types saves programming time and effort from declaring variables

34. Mention the use of the split function in Python?
Ans: The use of the split function in Python is that it breaks a string into shorter strings us

35. Explain what is Flask & its benefits?
Ans: Flask is a web micro framework for Python based on “Werkzeug, Jinja 2 and good intentions”

Flask is part of the micro-framework. Which means it will have little to no dependencies on ext

36. Mention what is the difference between Django, Pyramid, and Flask?
Ans: Flask is a “micro framework” primarily build for a small application with simpler requirem
Pyramid are build for larger applications. It provides flexibility and lets the developer use t

Like Pyramid, Django can also used for larger applications. It includes an ORM.

37. Mention what is Flask-WTF and what are their features?
Ans: Flask-WTF offers simple integration with WTForms. Features include for Flask WTF are
Integration with wtforms Secure form with csrf token Global csrf protection Internationalizatio
File upload that works with Flask Uploads

38. Explain what is the common way for the Flask script to work?
Ans: The common way for the flask script to work is…
Either it should be the import path for your application Or the path to a Python file

39. Explain how you can access sessions in Flask?
Ans: A session basically allows you to remember information from one request to another. In a f

40. Is Flask an MVC model and if yes give an example showing MVC pattern for your application?
Ans: Basically, Flask is a minimalistic framework which behaves same as MVC framework. So MVC i

```
from flask import Flaskapp = Flask(_name_)
@app.route(“/”)

Def hello():

return “Hello World”

app.run(debug = True)
```

In this code your,
Configuration part will be

```
from flask import Flask

app = Flask(_name_)
```

View part will be

```
@app.route(“/”)

Def hello():

return “Hello World”
```

While you model or main part will be

```
app.run(debug = True)
```

41. What type of a language is python? Interpreted or Compiled?
Ans: Beginner’s Answer:
Python is an interpreted, interactive, objectoriented programming language.
Expert Answer:
Python is an interpreted language, as opposed to a compiled one, though the
distinction can be blurry because of the presence of the bytecode compiler. This means
that source files can be run directly without explicitly creating an executable which is
then run.

42. What do you mean by python being an “interpreted language”? (Continues from previous questi
Ans: An interpreted languageis a programming languagefor which most of its
implementations execute instructions directly, without previously compiling a program
into machinelanguageinstructions. In context of Python, it means that Python program
runs directly from the source code.

43. What is python’s standard way of identifying a block of code?
Ans: Indentation.

44. Please provide an example implementation of a function called “my_func” that returns the sq
Ans:
defmy_func(x):
returnx**2

45. Is python statically typed or dynamically typed?

Ans: Dynamic.
In a statically typed language, the type of variables must be known (and usually declared) at the point at which it is used. Attempting to use it will be an error. In a dynamically typed language, objects still have a type, but it is determined at runtime. You are free to bind names (variables) to different objects with a different type. So long as you only perform operations valid for the type the interpreter doesn’t care what type they actually are.

46. Is python strongly typed or weakly typed language?
Ans: Strong.
In a weakly typed language a compiler / interpreter will sometimes change the type of a variable. For example, in some languages (like JavaScript) you can add strings to numbers ‘x’ + 3 becomes ‘x3’. This can be a problem because if you have made a mistake in your program, instead of raising an exception execution will continue but your variables now have wrong and unexpected values. In a strongly typed language (like Python) you can’t perform operations inappropriate to the type of the object attempting to add numbers to strings will fail. Problems like these are easier to diagnose because the exception is raised at the point where the error occurs rather than at some other, potentially far removed, place.

47. Create a unicode string in python with the string “This is a test string”?
Ans: some_variable=u’Thisisateststring’
Or
some_variable=u”Thisisateststring”

48. What is the python syntax for switch case statements?
Ans: Python doesn’t support switchcase statements. You can use ifelse statements for this purpose.

49. What is a lambda statement? Provide an example.
Ans: A lambda statement is used to create new function objects and then return them at runtime. Example:
my_func=lambdax:x**2
creates a function called my_func that returns the square of the argument passed.

50.What are the rules for local and global variables in Python?
Ans: If a variable is defined outside function then it is implicitly global. If variable is assigned new value inside the function means it is local. If we want to make it global we

need to explicitly define it as global. Variable referenced inside the function are implicit global

51.What is the output of the following program?
Ans:

```
#!/usr/bin/python
deffun1(a):
print’a:’,a
a=33;
print’locala:’,a
a=100
fun1(a)
print’aoutsidefun1:’,a
```

Ans. Output:

```
a:100
locala:33
aoutsidefun1:100
```

```
52.What is the output of the following program?
Ans:

#!/usr/bin/python
deffun2():
globalb
print’b:’,b
b=33
print’globalb:’,b
b=100
fun2()
print’boutsidefun2’,b
Ans. Output:
b:100
globalb:33
boutsidefun2:33




53. What is the output of the following program?
Ans:

#!/usr/bin/python

deffoo(x,y):
globala
a=42
x,y=y,x
b=33
b=17
c=100
print(a,b,x,y)
a,b,x,y=1,15,3,4
foo(17,4)
print(a,b,x,y)

Ans.Output:
4217417
421534

54.What is the output of the following program?
Ans:

#!/usr/bin/python
deffoo(x=[]):
x.append(1)
```

```
returnx
foo()
foo()

Output:
[1]
[1,1]

55. What is the purpose of #!/usr/bin/pythonon the first line in the above
code? Is there any advantage?
Ans: By specifying #!/usr/bin/pythonyou specify exactly which interpreter will be
used to run the script on a particular system. This is the hardcoded path to the python
interpreter for that particular system. The advantage of this line is that you can use a
specific python version to run your code.

56.What is the output of the following program?
Ans:

list=[‘a’,’b’,’c’,’d’,’e’]
printlist[10]
Ans. Output:
IndexError.Or Error.

57.What is the output of the following program?
Ans:

list=[‘a’,’b’,’c’,’d’,’e’]
printlist[10:]
Ans. Output:
[]
Theabovecodewilloutput[],andwillnotresultinanIndexError.
As one would expect, attempting to access a member of a list using an index that
exceeds the number of members results in an IndexError.

58. What does this list comprehension do:
Ans:

[x**2forxinrange(10)ifx%2==0]
Ans. Creates the following list:
[0,4,16,36,64]

59. Do sets, dictionaries and tuples also support comprehensions?
Ans: Sets and dictionaries support it. However tuples are immutable and have
generators but not comprehensions.
Set Comprehension:
r={xforxinrange(2,101)
ifnotany(x%y==0foryinrange(2,x))}
Dictionary Comprehension:
{i:jfori,jin{1:’a’,2:’b’}.items()}
since
{1:’a’,2:’b’}.items()returnsalistof2-Tuple.iisthefirstelement
oftuplejisthesecond.
```

```
605
606 60.What are some mutable and immutable datatypes/datastructures in
607 python?
608 Ans:
609 Mutable Types Immutable Types
610
611 Dictionary number
612 List boolean
613 string
614 tuple
615
616 61.What are generators in Python?
617 Ans: Generators are functions that return an iterable collection of items, one at a time, in a
618 Let’s try and build a generator for fibonacci numbers –
619
620 ## generate fibonacci numbers upto n
621 def fib(n):
622     p, q = 0, 1
623     while(p < n):
624         yield p
625         p, q = q, p + q
626
627 x = fib(10)    # create generator object
628
629 ## iterating using __next__(), for Python2, use next()
630 x.__next__()    # output => 0
631 x.__next__()    # output => 1
632 x.__next__()    # output => 1
633 x.__next__()    # output => 2
634 x.__next__()    # output => 3
635 x.__next__()    # output => 5
636 x.__next__()    # output => 8
637 x.__next__()    # error
638
639 ## iterating using loop
640 for i in fib(10):
641     print(i)    # output => 0 1 1 2 3 5 8
642
643
644
645
646 62.What can you use Python generator functions for?
647 Ans: One of the reasons to use generator is to make the solution clearer for some kind
648 of solutions.
649 The other is to treat results one at a time, avoiding building huge lists of results that you
650 would process separated anyway.
651
652 63.When is not a good time to use python generators?
653 Ans: Use list instead of generator when:
654 1 You need to access the data multiple times (i.e. cache the results instead of
655 recomputing them)
656 2 You need random access (or any access other than forward sequential order):
```

```
3 You need to join strings (which requires two passes over the data)
4 You are using PyPy which sometimes can’t optimize generator code as much
as it can with normal function calls and list manipulations.

64.What’s your preferred text editor?
Ans: Emacs. Any alternate answer leads to instant disqualification of the applicant

65.When should you use generator expressions vs. list comprehensions in Python and vice-versa?
Ans: Iterating over the generator expression or the list comprehension will do the same
thing. However, the list comp will create the entire list in memory first while the
generator expression will create the items on the fly, so you are able to use it for very
large (and also infinite!) sequences.

66. What is a negative index in Python?
Ans: Python arrays and list items can be accessed with positive or negative numbers. A
negative Index accesses the elements from the end of the list counting backwards.
Example:
a=[123]
printa[-3]
printa[-2]
Outputs:
1
2

67. What is the difference between range and xrange functions?
Ans: Range returns a list while xrange returns an xrange object which take the
same memory no matter of the range size. In the first case you have all items already
generated (this can take a lot of time and memory). In Python 3 however, range is
implemented with xrange and you have to explicitly call the list function if you want to
convert it to a list.



68. How can I find methods or attributes of an object in Python?
Ans: Builtin dir() function of Python ,on an instance shows the instance variables as
well as the methods and class attributes defined by the instance’s class and all its base
classes alphabetically. So by any object as argument to dir() we can find all the
methods & attributes of the object’s class

69. What is the statement that can be used in Python if a statement is required syntactically b
Ans:
pass

70. Do you know what is the difference between lists and tuples? Can you give me an example for
Ans:
First list are mutable while tuples are not, and second tuples can be hashed e.g.
to be used as keys for dictionaries. As an example of their usage, tuples are used when
the order of the elements in the sequence matters e.g. a geographic coordinates, “list”
of points in a path or route, or set of actions that should be executed in specific order.
Don’t forget that you can use them a dictionary keys. For everything else use lists

71. What is the function of “self”?
```

```
Ans:
“Self” is a variable that represents the instance of the object to itself. In most of
the object oriented programming languages, this is passed to the methods as a hidden
parameter that is defined by an object. But, in python it is passed explicitly. It refers to
separate instance of the variable for individual objects. The variables are referred as
“self.xxx”.

72. How is memory managed in Python?
Ans:
Memory management in Python involves a private heap containing all
Python objects and data structures. Interpreter takes care of Python heap and
the programmer has no access to it. The allocation of heap space for Python
objects is done by Python memory manager. The core API of Python provides
some tools for the programmer to code reliable and more robust program. Python
also has a builtin garbage collector which recycles all the unused memory.
The gc module defines functions to enable /disable garbage collector:
gc.enable() Enables automatic garbage collection.
gc.disable()-Disables automatic garbage collection

73. What is __init__.py?
Ans:
It is used to import a module in a directory, which is called package import.

74. Print contents of a file ensuring proper error handling?
Ans:
try:
withopen(‘filename’,’r’)asf:
printf.read()
exceptIOError:
print”Nosuchfileexists”

75 How do we share global variables across modules in Python?
Ans:
We can create a config file and store the entire global variable to be
shared across modules in it. By simply importing config, the entire global variable
defined will be available for use in other modules.
For example I want a, b & c to share between modules.
config.py :
a=0
b=0
c=0
module1.py:
importconfig
config.a=1
config.b=2
config.c=3
print”a,b&resp.are:”,config.a,config.b,config.c

output of module1.py will be
123

76. Does Python support Multithreading?
```

Ans: Yes
Medium

77. How do I get a list of all files (and directories) in a given directory in Python?
Ans: Following is one possible solution there can be other similar ones:
import os
for dirname,dirnames,filenames in os.walk(‘.’):
#printpathtoallsubdirectoriesfirst.
forsubdirnameindirnames:
printos.path.join(dirname,subdirname)
#printpathtoallfilenames.
forfilenameinfilenames:
printos.path.join(dirname,filename)
#Advancedusage:
#editingthe’dirnames’listwillstopos.walk()fromrecursing
intothere.
if’.git’indirnames:
#don’tgointoany.gitdirectories.
dirnames.remove(‘.git’)

78. How to append to a string in Python?
Ans: The easiest way is to use the += operator. If the string is a list of character, join()
function can also be used.

79. How to convert a string to lowercase in Python?
Ans: use lower() function.
Example:
s=’MYSTRING’
prints.lower()

80. How to convert a string to lowercase in Python?
Ans: Similar to the above question. use upper() function instead.

81. How to check if string A is substring of string B?
Ans: The easiest way is to use the in operator.
>>> ‘abc’ in ‘abcdefg’
True

82. Find all occurrences of a substring in Python
Ans: There is no simple builtin string function that does what you’re looking for, but
you could use the more powerful regular expressions:
>>>[m.start()forminre.finditer(‘test’,’testtesttesttest’)]
[0,5,10,15]//thesearestartingindicesforthestring

83. What is GIL? What does it do?Talk to me about the GIL. How does it impact concurrency in Py
Ans: Python’s GIL is intended to serialize access to interpreter internals from different
threads. On multicore systems, it means that multiple threads can’t effectively make
use of multiple cores. (If the GIL didn’t lead to this problem, most people wouldn’t care
about the GIL it’s only being raised as an issue because of the increasing prevalence
of multicore systems.)
Note that Python’s GIL is only really an issue for CPython, the reference
implementation. Jython and IronPython don’t have a GIL. As a Python developer, you

don’t generally come across the GIL unless you’re writing a C extension. C extension writers need to release the GIL when their extensions do blocking I/O, so that other threads in the Python process get a chance to run.

84. Print the index of a specific item in a list?
Ans: use the index() function
>>>[“foo”,”bar”,”baz”].index(‘bar’)
1
.

85.How do you iterate over a list and pull element indices at the same time?
Ans: You are looking for the enumerate function. It takes each element in a sequence (like a list) and sticks it’s location right before it. For example:

>>>my_list=[‘a’,’b’,’c’]
>>>list(enumerate(my_list))
[(0,’a’),(1,’b’),(2,’c’)]
Note that enumerate() returns an object to be iterated over, so wrapping it in list() just helps us see what enumerate() produces.
An example that directly answers the question is given below
my_list=[‘a’,’b’,’c’]
fori,charinenumerate(my_list):
printi,char
The output is:
0a
1b
2c

86. How does Python’s list.sort work at a high level? Is it stable? What’s the runtime?
Ans: In early pythonversions, the sort function implemented a modified version of quicksort. However, it was deemed unstable and as of 2.3 they switched to using an adaptive mergesort algorithm.

87. What does the list comprehension do:
Ans:
my_list=[(x,y,z)forxinrange(1,30)foryinrange(x,30)forzin range(y,30)ifx**2+y**2==z**2]
It creates a list of tuples called my_list, where the first 2 elements are the perpendicular sides of right angle triangle and the third value ‘z’ is the hypotenuse.
[(3,4,5),(5,12,13),(6,8,10),(7,24,25),(8,15,17),(9,12,15),
(10,24,26),(12,16,20),(15,20,25),(20,21,29)]

88. How can we pass optional or keyword parameters from one function to another in Python?
Ans:
Gather the arguments using the * and ** specifiers in the function’s parameter list. This gives us positional arguments as a tuple and the keyword arguments as a dictionary.
Then we can pass these arguments while calling another function by using * and **:
deffun1(a,*tup,**keywordArg):
…
keywordArg[‘width’]=’23.3c’
…
Fun2(a,*tup,**keywordArg)

89. Python How do you make a higher order function in Python?
Ans:
A higherorder function accepts one or more functions as input and returns a new function. Sometimes it is required to use function as data To make high order function , we need to import functools module The functools.partial() function is used often for high order function.

90. What is map?
Ans:
The syntax of map is:
map(aFunction,aSequence)
The first argument is a function to be executed for all the elements of the iterable given as the second argument. If the function given takes in more than 1 arguments, then many iterables are given.

91. Tell me a very simple solution to print every other element of this list?
Ans:

```
L=[0,10,20,30,40,50,60,70,80,90]
L[::2]
```

92. Are Tuples immutable?
Ans: Yes.

93. Why is not all memory freed when python exits?
Ans: Objects referenced from the global namespaces of Python modules are not always deallocated when Python exits. This may happen if there are circular references. There are also certain bits of memory that are allocated by the C library that are impossible to free (e.g. a tool like the one Purify will complain about these). Python is, however, aggressive about cleaning up memory on exit and does try to destroy every single object. If you want to force Python to delete certain things on deallocation, you can use the at exit module to register one or more exit functions to handle those deletions.

94. What is Java implementation of Python popularly know?
Ans: Jython.

95. What is used to create unicode strings in Python?
Ans:
Add u before the string.
u ‘mystring’

96. What is a docstring?
Ans:
docstring is the documentation string for a function. It can be accessed by function_name.__doc__

97. Given the list below remove the repetition of an element.
Ans:

words=[‘one’,’one’,’two’,’three’,’three’,’two’]
A bad solution would be to iterate over the list and checking for copies somehow and then remove them!

A very good solution would be to use the set type. In a Python set, duplicates are not allowed.
So, list(set(words)) would remove the duplicates.

98. Print the length of each line in the file ‘file.txt’ not including any whitespaces at the end of the lines?
Ans:
withopen(“filename.txt”,”r”)asf1:
printlen(f1.readline().rstrip())
rstrip() is an inbuilt function which strips the string from the right end of spaces or tabs (whitespace characters).

99. What is wrong with the code?
Ans:

func([1,2,3])#explicitlypassinginalist
func() #usingadefaultemptylist
deffunc(n=[]):
#dosomethingwithn
printn

This would result in a NameError. The variable n is local to function func and can’t be accessesd outside. So, printing it won’t be possible.

100. What does the below mean?
Ans:

s = a + ‘[‘ + b + ‘:’ + c + ‘]’

seems like a string is being concatenated. Nothing much can be said without knowing types of variables a, b, c. Also, if all of the a, b, c are not of type string, TypeError would be raised. This is because of the string constants (‘[‘ , ‘]’) used in the statement.
101. What are Python decorators?
Ans:
A Python decorator is a specific change that we make in Python syntax to alter functions easily.

102. What is namespace in Python?
Ans:
In Python, every name introduced has a place where it lives and can be hooked for. This is known as namespace. It is like a box where a variable name is mapped to the object placed. Whenever the variable is searched out, this box will be searched, to get corresponding object.

103. Explain the role of repr function.
Ans:
Python can convert any value to a string by making use of two functions repr() or

```
str(). The str() function returns representations of values which are humanreadable,
while repr() generates representations which can be read by the interpreter. repr()
returns a machinereadable representation of values, suitable for an exec command.
Following code sniipets shows working of repr() & str() :
deffun():
y=2333.3
x=str(y)
z=repr(y)
print”y:”,y
print”str(y):”,x
print”repr(y):”,z
fun()
———-
output
y:2333.3
str(y):2333.3
repr(y):2333.3000000000002

104. What is LIST comprehensions features of Python used for?
Ans:
LIST comprehensions features were introduced in Python version 2.0, it creates
a new list based on existing list. It maps a list into another list by applying a function to

each of the elements of the existing list. List comprehensions creates lists without using
map() , filter() or lambda form.

105. Explain how to copy an object in Python.?
Ans:
There are two ways in which objects can be copied in python. Shallow copy &
Deep copy. Shallow copies duplicate as minute as possible whereas Deep copies
duplicate everything. If a is object to be copied then …
copy.copy(a) returns a shallow copy of a.
copy.deepcopy(a) returns a deep copy of a.

106. Describe how to send mail from a Python script?
Ans:
The smtplib module defines an SMTP client session object that can be used to
send mail to any Internet machine.
A sample email is demonstrated below.
import smtplib
SERVER = smtplib.SMTP(‘smtp.server.domain’)
FROM = sender@mail.com
TO = [“user@mail.com”] # must be a list
SUBJECT = “Hello!”
TEXT = “This message was sent with Python’s smtplib.”
# Main message
message = “””
From: Lincoln < sender@mail.com >
To: CarreerRide user@mail.com
Subject: SMTP email msg
This is a test email. Acknowledge the email by responding.
””” % (FROM, “, “.join(TO), SUBJECT, TEXT)
```

```
server = smtplib.SMTP(SERVER)
server.sendmail(FROM, TO, message)
server.quit()

107. Which of the languages does Python resemble in its class syntax?
Ans: c++.

108. Python  How to create a multidimensional list?
Ans: There are two ways in which Multidimensional list can be created:
By direct initializing the list as shown below to create myList below.
>>>myList=[[227,122,223],[222,321,192],[21,122,444]]
>>>printmyList[0]
>>>printmyList[1][2]
_____
Output
[227, 122, 223]
192
The second approach is to create a list of the desired length first and then fill in each
element with a newly created lists demonstrated below :
>>>list=[0]*3
>>>foriinrange(3):
>>>list[i]=[0]*2
>>>foriinrange(3):
>>>forjinrange(2):
>>>list[i][j]=i+j
>>>printlist
_____
Output
[[0,1],[1,2],[2,3]]

109. Explain the disadvantages of python
Ans: Disadvantages of Python are: Python isn’t the best for memory intensive tasks.
Python is interpreted language & is slow compared to C/C++ or Java.

110. Explain how to make Forms in python.
Ans. As python is scripting language forms processing is done by Python. We need to
import cgi module to access form fields using FieldStorage class.
Every instance of class FieldStorage (for ‘form’) has the following attributes:
form.name: The name of the field, if specified.

form.filename: If an FTP transaction, the clientside filename.
form.value: The value of the field as a string.
form.file: file object from which data can be read.
form.type: The content type, if applicable.
form.type_options: The options of the ‘contenttype’ line of the HTTP request, returned
as a dictionary.
form.disposition: The field ‘contentdisposition’; None if unspecified.
form.disposition_options: The options for ‘contentdisposition’.
form.headers: All of the HTTP headers returned as a dictionary.
A code snippet of form handling in python:
importcgi
form=cgi.FieldStorage()
```

```
ifnot(form.has_key(“name”)andform.has_key(“age”)):
print”<H1>Name&AgenotEntered</H1>”
print”FilltheName&Ageaccurately.”
return
print”<p>name:”,form[“name”].value
print”<p>Age:”,form[“age”].value

111. Explain how python is interpreted.
Ans: Python program runs directly from the source code. Each type Python programs
are executed code is required. Python converts source code written by the programmer
into intermediate language which is again translated it into the native language
machine language that is executed. So Python is an Interpreted language.

112. Explain how to overload constructors (or methods) in Python.?
Ans. _init__ () is a first

113.What is the difference between List & Tuple in Python.?
LIST vs TUPLES
LIST    TUPLES
Lists are mutable i.e they can be edited.       Tuples are immutable (tuples are lists which ca
Lists are slower than tuples.   Tuples are faster than list.
Syntax: list_1 = [10, ‘Chelsea’, 20]    Syntax: tup_1 = (10, ‘Chelsea’ , 20)


114.What are the key features of Python?
Ans:

Python is an interpreted language. That means that, unlike languages like C and its variants, P
Python is dynamically typed, this means that you don’t need to state the types of variables whe
Python is well suited to object orientated programming in that it allows the definition of clas
In Python, functions are first-class objects. This means that they can be assigned to variables
Writing Python code is quick but running it is often slower than compiled languages. Fortunatel
Python finds use in many spheres – web applications, automation, scientific modeling, big data


115.How is Python an interpreted language?
Ans: An interpreted language is any programming language which is not in machine level code bet

116.How is memory managed in Python?
Ans:

Memory management in python is managed by Python private heap space. All Python objects and dat
The allocation of heap space for Python objects is done by Python’s memory manager. The core AP
Python also has an inbuilt garbage collector, which recycles all the unused memory and so that


 117.What is PYTHONPATH?
Ans:It is an environment variable which is used when a module is imported. Whenever a module is

118. What are python modules? Name some commonly used built-in modules in Python?
Ans:Python modules are files containing Python code. This code can either be functions classes

```

Some of the commonly used built-in modules are:

os
sys
math
random
data time
JSON
119.What are local variables and global variables in Python?
Ans:

Global Variables:

Variables declared outside a function or in global space are called global variables. These var

Local Variables:

Any variable declared inside a function is known as a local variable. This variable is present

Example:

```
1
2
3
4
5
6
a=2
def add():
b=3
c=a+b
print(c)
add()
```

Output: 5

When you try to access the local variable outside the function add(), it will throw an error.

120. Is python case sensitive?
Ans:Yes. Python is a case sensitive language.

121.What is type conversion in Python?
Ans:Type conversion refers to the conversion of one data type iinto another.

int() – converts any data type into integer type

float() – converts any data type into float type

ord() – converts characters into integer

hex() – converts integers to hexadecimal

oct() – converts integer to octal

tuple() – This function is used to convert to a tuple.

set() – This function returns the type after converting to set.

list() – This function is used to convert any data type to a list type.

dict() – This function is used to convert a tuple of order (key,value) into a dictionary.

str() – Used to convert integer into a string.

complex(real,imag) – This functionconverts real numbers to complex(real,imag) number.

122. How to install Python on Windows and set path variable?
Ans:To install Python on Windows, follow the below steps:

Install python from this link: https://www.python.org/downloads/
After this, install it on your PC. Look for the location where PYTHON has been installed on you
Then go to advanced system settings and add a new variable and name it as PYTHON_NAME and paste
Look for the path variable, select its value and select ‘edit’.
Add a semicolon towards the end of the value if it’s not present and then type %PYTHON_HOME%

123. Is indentation required in python?
Ans:Indentation is necessary for Python. It specifies a block of code. All code within loops, c

124. What is the difference between Python Arrays and lists?
Ans:Arrays and lists, in Python, have the same way of storing data. But, arrays can hold only a

Example:

```
1
2
3
4
5
import array as arr
My_Array=arr.array('i',[1,2,3,4])
My_list=[1,'abc',1.20]
print(My_Array)
print(My_list)
```

Output:

array(‘i’, [1, 2, 3, 4]) [1, ‘abc’, 1.2]

125. What are functions in Python?
Ans:A function is a block of code which is executed only when it is called. To define a Python

Example:

```
1
2
3
def Newfunc():
print("Hi, Welcome to Edureka")
Newfunc(); #calling the function
Output: Hi, Welcome to Edureka

126.What is __init__?
Ans:__init__ is a method or constructor in Python. This method is automatically called to alloc

Here is an example of how to use it.

1
2
3
4
5
6
7
8
9
10
11
class Employee:
def __init__(self, name, age,salary):
self.name = name
self.age = age
self.salary = 20000
E1 = Employee("XYZ", 23, 20000)
# E1 is the instance of class Employee.
#__init__ allocates memory for E1.
print(E1.name)
print(E1.age)
print(E1.salary)
Output:



XYZ

23

20000



127.What is a lambda function?
Ans:An anonymous function is known as a lambda function. This function can have any number of p

Example:

```

```
1
2
a = lambda x,y : x+y
print(a(5, 6))
Output: 11

128. What is self in Python?
Ans:Self is an instance or an object of a class. In Python, this is explicitly included as the

The self variable in the init method refers to the newly created object while in other methods,

129. How does break, continue and pass work?
Break   Allows loop termination when some condition is met and the control is transferred to th
Continue        Allows skipping some part of a loop when some specific condition is met and the
Pass    Used when you need some block of code syntactically, but you want to skip its execution
130. What does [::-1} do?
Ans: [::-1] is used to reverse the order of an array or a sequence.
For example:
1
2
3
import array as arr
My_Array=arr.array('i',[1,2,3,4,5])
My_Array[::-1]
Output: array('i', [5, 4, 3, 2, 1])

[::-1] reprints a reversed copy of ordered data structures such as an array or a list. the orig


131. How can you randomize the items of a list in place in Python?
Ans: Consider the example shown below:

1
2
3
4
from random import shuffle
x = ['Keep', 'The', 'Blue', 'Flag', 'Flying', 'High']
shuffle(x)
print(x)
The output of the following code is as below.

['Flying', 'Keep', 'Blue', 'High', 'The', 'Flag']
132. What are python iterators?
Ans:Iterators are objects which can be traversed though or iterated upon.

133. How can you generate random numbers in Python?
Ans: Random module is the standard module that is used to generate a random number. The method

1
2
import random
```

```
random.random
The statement random.random() method return the floating point number that is in the range of

randrange(a, b): it chooses an integer and define the range in-between [a, b). It returns the e
uniform(a, b): it chooses a floating point number that is defined in the range of [a,b).Iyt ret
normalvariate(mean, sdev): it is used for the normal distribution where the mu is a mean and th
The Random class that is used and instantiated creates an independent multiple random number ge
134. What is the difference between range & xrange?
Ans: For the most part, xrange and range are the exact same in terms of functionality. They bot

This means that xrange doesn’t actually generate a static list at run-time like range does. It

This is especially true if you have a really memory sensitive system such as a cell phone that

135. How do you write comments in python?
Ans:Comments in Python start with a # character. However, alternatively at times, commenting is

Example:

#Comments in Python start like this
print("Comments in Python start with a #")
Output:  Comments in Python start with a #

136. What is pickling and unpickling?
Ans: Pickle module accepts any Python object and converts it into a string representation and d

137. What are the generators in python?
Ans: Functions that return an iterable set of items are called generators.

138. How will you capitalize the first letter of string?
Ans:In Python, the capitalize() method capitalizes the first letter of a string. If the string

139. How will you convert a string to all lowercase?
Ans:To convert a string to lowercase, lower() function can be used.

Example:

1
2
stg='ABCD'
print(stg.lower())
Output: abcd

140. How to comment multiple lines in python?
Ans: Multi-line comments appear in more than one line. All the lines to be commented are to be

141.What are docstrings in Python?
Ans:Docstrings are not actually comments, but, they are documentation strings. These docstrings

Example:

1
```

```
2
3
4
5
6
7
8
"""
Using docstring as a comment.
This code divides 2 numbers
"""
x=8
y=4
z=x/y
print(z)
Output: 2.0

141. What is the purpose of is, not and in operators?
Ans:Operators are special functions. They take one or more values and produce a corresponding r

is: returns true when 2 operands are true  (Example: “a” is ‘a’)

not: returns the inverse of the boolean value

in: checks if some element is present in some sequence

142. What is the usage of help() and dir() function in Python?
Ans: Help() and dir() both functions are accessible from the Python interpreter and used for vi

Help() function: The help() function is used to display the documentation string and also facil
Dir() function: The dir() function is used to display the defined symbols.
143. Whenever Python exits, why isn’t all the memory de-allocated?
Ans:

Whenever Python exits, especially those Python modules which are having circular references to
It is impossible to de-allocate those portions of memory that are reserved by the C library.
On exit, because of having its own efficient clean up mechanism, Python would try to de-allocat
143. What is a dictionary in Python?
Ans: The built-in datatypes in Python is called dictionary. It defines one-to-one relationship

Let’s take an example:

The following example contains some keys. Country, Capital & PM. Their corresponding values are

1
dict={'Country':'India','Capital':'Delhi','PM':'Modi'}
1
print dict[Country]
India
1
print dict[Capital]
Delhi
```

```
1
print dict[PM]
Modi
```

144. How can the ternary operators be used in python?

Ans: The Ternary operator is the operator that is used to show the conditional statements. This

Syntax:

The Ternary operator will be given as:

```
[on_true] if [expression] else [on_false]x, y = 25, 50big = x if x < y else y
```

Example:

The expression gets evaluated like if x<y else y, in this case if x<y is true then the value is

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

Ans: We use *args when we aren’t sure how many arguments are going to be passed to a function,

147. What does len() do?

Ans:It is used to determine the length of a string, a list, an array, etc.

Example:

```
1
2
stg='ABCD'
len(stg)
```

148. Explain split(), sub(), subn() methods of “re” module in Python.

Ans: To modify the strings, Python’s “re” module is providing 3 methods. They are:

split() – uses a regex pattern to “split” a given string into a list.

sub() – finds all substrings where the regex pattern matches and then replace them with a diffe

subn() – it is similar to sub() and also returns the new string along with the no. of replaceme

149. What are negative indexes and why are they used?

Ans: The sequences in Python are indexed and it consists of the positive as well as negative nu

The index for the negative number starts from ‘-1’ that represents the last index in the sequer

The negative index is used to remove any new-line spaces from the string and allow the string t

150. What are Python packages?

Ans: Python packages are namespaces containing multiple modules.

151.How can files be deleted in Python?

Ans: To delete a file in Python, you need to import the OS Module. After that, you need to use

Example:

```
1
2
import os
os.remove("xyz.txt")
```

152. What are the built-in types of python?

Ans: Built-in types in Python are as follows –

Integers
Floating-point
Complex numbers
Strings
Boolean
Built-in functions
153. What advantages do NumPy arrays offer over (nested) Python lists?
Ans:

Python’s lists are efficient general-purpose containers. They support (fairly) efficient insert
They have certain limitations: they don’t support “vectorized” operations like elementwise addi
NumPy is not just more efficient; it is also more convenient. You get a lot of vector and matri
NumPy array is faster and You get a lot built in with NumPy, FFTs, convolutions, fast searching
154. How to add values to a python array?
Ans: Elements can be added to an array using the append(), extend() and the insert (i,x) functi

Example:

1
2
3
4
5
6
7

```
a=arr.array('d', [1.1 , 2.1 ,3.1] )
a.append(3.4)
print(a)
a.extend([4.5,6.3,6.8])
print(a)
a.insert(2,3.8)
print(a)
```

Output:

array(‘d’, [1.1, 2.1, 3.1, 3.4])

array(‘d’, [1.1, 2.1, 3.1, 3.4, 4.5, 6.3, 6.8])

array(‘d’, [1.1, 2.1, 3.8, 3.1, 3.4, 4.5, 6.3, 6.8])

155. How to remove values to a python array?
Ans: Array elements can be removed using pop() or remove() method. The difference between these

Example:

```
a=arr.array('d', [1.1, 2.2, 3.8, 3.1, 3.7, 1.2, 4.6])
print(a.pop())
print(a.pop(3))
a.remove(1.1)
print(a)
```

```
Output:

4.6

3.1

array('d', [2.2, 3.8, 3.7, 1.2])



156. Does Python have OOps concepts?
Ans: Python is an object-oriented programming language. This means that any program can be solv

157. What is the difference between deep and shallow copy?
Ans: Shallow copy is used when a new instance type gets created and it keeps the values that ar

Deep copy is used to store the values that are already copied. Deep copy doesn't copy the refer

158. How is Multithreading achieved in Python?
Ans:

Python has a multi-threading package but if you want to multi-thread to speed your code up, the
Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only o
This happens very quickly so to the human eye it may seem like your threads are executing in pa
All this GIL passing adds overhead to execution. This means that if you want to make your code
159. What is the process of compilation and linking in python?
Ans: The compiling and linking allows the new extensions to be compiled properly without any er

The steps that are required in this as:

Create a file with any name and in any language that is supported by the compiler of your syste
Place this file in the Modules/ directory of the distribution which is getting used.
Add a line in the file Setup.local that is present in the Modules/ directory.
Run the file using spam file.o
After a successful run of this rebuild the interpreter by using the make command on the top-lev
If the file is changed then run rebuildMakefile by using the command as 'make Makefile'.
160. What are Python libraries? Name a few of them.
Ans: Python libraries are a collection of Python packages. Some of the majorly used python libr

161. What is split used for?
Ans: The split() method is used to separate a given string in Python.

Example:

1
2
a="KausalVikash python"
print(a.split())
Output:  ['KausalVikash', 'python']

162. How to import modules in python?
Ans: Modules can be imported using the import keyword.  You can import modules in three ways-
```

```
1593
1594  Example:
1595
1596  1
1597  2
1598  3
1599  import array           #importing using the original module name
1600  import array as arr    # importing using an alias name
1601  from array import *    #imports everything present in the array module
1602
1603
1604  163. Explain Inheritance in Python with an example.
1605  Ans: Inheritance allows One class to gain all the members(say attributes and methods) of anothe
1606
1607  They are different types of inheritance supported by Python:
1608
1609  Single Inheritance – where a derived class acquires the members of a single super class.
1610  Multi-level inheritance – a derived class d1 in inherited from base class base1, and d2 are inh
1611  Hierarchical inheritance – from one base class you can inherit any number of child classes
1612  Multiple inheritance – a derived class is inherited from more than one base class.
1613
1614
1615  164. How are classes created in Python?
1616  Ans: Class in Python is created using the class keyword.
1617
1618  Example:
1619
1620  1
1621  2
1622  3
1623  4
1624  5
1625  class Employee:
1626  def __init__(self, name):
1627  self.name = name
1628  E1=Employee("abc")
1629  print(E1.name)
1630  Output: abc
1631
1632  165. What is monkey patching in Python?
1633  Ans: In Python, the term monkey patch only refers to dynamic modifications of a class or module
1634
1635  Consider the below example:
1636
1637  1
1638  2
1639  3
1640  4
1641  # m.py
1642  class MyClass:
1643  def f(self):
1644  print "f()"
```

```
We can then run the monkey-patch testing like this:

1
2
3
4
5
6
7
import m
def monkey_f(self):
print "monkey_f()"
m.MyClass.f = monkey_f
obj = m.MyClass()
obj.f()
The output will be as below:

monkey_f()
As we can see, we did make some changes in the behavior of f() in MyClass using the function we

166. Does python support multiple inheritance?
Ans: Multiple inheritance means that a class can be derived from more than one parent classes.

167. What is Polymorphism in Python?
Ans: Polymorphism means the ability to take multiple forms. So, for instance, if the parent cla

168. Define encapsulation in Python?
Ans: Encapsulation means binding the code and the data together. A Python class in an example c

169. How do you do data abstraction in Python?
Ans: Data Abstraction is providing only the required details and hiding the implementation from

170.Does python make use of access specifiers?
Ans: Python does not deprive access to an instance variable or function. Python lays down the c
171. How to create an empty class in Python?
Ans: An empty class is a class that does not have any code defined within its block. It can be
For example-
1
2
3
4
5
class a:
    pass
obj=a()
obj.name="xyz"
print("Name = ",obj.name)
Output:

Name =  xyz


```

172.What’s The Process To Get The Home Directory Using ‘~’ In Python?
Ans: You need to import the os module, and then just a single line would do the rest.

```
import os
print (os.path.expanduser('~'))
```

Output:

```
/home/runner
```

173.How To Find Bugs Or Perform Static Analysis In A Python Application?
Ans:

You can use PyChecker, which is a static analyzer. It identifies the bugs in Python project and
Another tool is Pylint, which checks whether the Python module satisfies the coding standard.
174.When Is The Python Decorator Used?
Ans: Python decorator is a relative change that you do in Python syntax to adjust the functions

175.Can Python be used for web client and web server side programming? And which one is best su
Ans: Python is best suited for web server-side application development due to its vast set of

However, Python can be used as a web client-side application which needs some conversions for a

176. Mention at least 3-4 benefits of using Python over the other scripting languages such as J
Ans: Enlisted below are some of the benefits of using Python.

Application development is faster and easy.
Extensive support of modules for any kind of application development including data analytics/m
An excellent support community to get your answers.

177.What is the type () in Python?
Ans: The built-in method which decides the types of the variable at the program runtime is know

178.What are the key points of Python?
Ans:

Similar to PERL and PHP, Python is processed by the interpreter at runtime. Python supports Obj
Derived from other languages, such as ABC, C, C++, Modula-3, SmallTalk, Algol-68, Unix shell, a
Python is copyrighted, and its source code is available under the GNU General Public License (G
Supports the development of many applications, from text processing to games.
Works for scripting, embedded code and compiled the code.
Detailed

179.How is memory managed in Python?
Ans: Memory is managed by the private heap space. All objects and data structures are located i

180.What tools can help find bugs or perform the static analysis?
Ans: For performing Static Analysis, PyChecker is a tool that detects the bugs in source code a

181.How Does Python Handle Memory Management?
Ans:

Python uses private heaps to maintain its memory. So the heap holds all the Python objects and
And it’s the Python memory manager that handles the Private heap. It does the required allocati
Python employs a built-in garbage collector, which salvages all the unused memory and offloads

182.What Are The Principal Differences Between The Lambda And Def?
Ans:

Lambda Vs. Def.
Def can hold multiple expressions while lambda is a uni-expression function.
Def generates a function and designates a name to call it later. Lambda forms a function object
Def can have a return statement. Lambda can’t have return statements.
Lambda supports to get used inside a list and dictionary.

183.Write A Reg Expression That Confirms An Email Id Using The Python Reg Expression Module “Re
Ans: Python has a regular expression module “re.”

Check out the “re” expression that can check the email id for .com and .co.in subdomain.

```
import re
print(re.search(r"[0-9a-zA-Z.]+@[a-zA-Z]+\.(com|co\.in)$","micheal.pages@mp.com"))
```

184.What Do You Think Is The Output Of The Following Code Fragment? Is There Any Error In The C
Ans:

```
list = ['a', 'b', 'c', 'd', 'e']
print (list[10:])
```

The result of the above lines of code is []. There won’t be any error like an IndexError.

You should know that trying to fetch a member from the list using an index that exceeds the mem

185. Is There A Switch Or Case Statement In Python? If Not Then What Is The Reason For The Same
Ans: No, Python does not have a Switch statement, but you can write a Switch function and then

186.What Is A Built-In Function That Python Uses To Iterate Over A Number Sequence?
Ans: Range() generates a list of numbers, which is used to iterate over for loops.

```
for i in range(5):
```

```
    print(i)
The range() function accompanies two sets of parameters.

range(stop)
stop: It is the no. of integers to generate and starts from zero. eg. range(3) == [0, 1, 2].
range([start], stop[, step])
Start: It is the starting no. of the sequence.
Stop: It specifies the upper limit of the sequence.
Step: It is the incrementing factor for generating the sequence.
Points to note:
Only integer arguments are allowed.
Parameters can be positive or negative.
The range() function in Python starts from the zeroth index.


187.What Are The Optional Statements Possible Inside A Try-Except Block In Python?
Ans: There are two optional clauses you can use in the try-except block.

The “else” clause
It is useful if you want to run a piece of code when the try block doesn’t create an exception
The “finally” clause
It is useful when you want to execute some steps which run, irrespective of whether there occur


188.What Is A String In Python?
Ans: A string in Python is a sequence of alpha-numeric characters. They are immutable objects.

189. What Is Slicing In Python?
Ans: Slicing is a string operation for extracting a part of the string, or some part of a list

190. What Is %S In Python?
Ans: Python has support for formatting any value into a string. It may contain quite complex ex

One of the common usages is to push values into a string with the %s format specifier. The form

191.What Is The Index In Python?
Ans: An index is an integer data type which denotes a position within an ordered list or a stri

In Python, strings are also lists of characters. We can access them using the index which begin

For example, in the string “Program,” the indexing happens like this:

Program 0 1 2 3 4 5
192. What Is Docstring In Python?
Ans: A docstring is a unique text that happens to be the first statement in the following Pytho

Module, Function, Class, or Method definition.

A docstring gets added to the __doc__ attribute of the string object.


```

193.What Is A Function In Python Programming?
Ans: A function is an object which represents a block of code and is a reusable entity. It brir

Python has given us many built-in functions such as print() and provides the ability to create

194. How Many Basic Types Of Functions Are Available In Python?
Ans: Python gives us two basic types of functions.

1. Built-in, and

2. User-defined.

The built-in functions happen to be part of the Python language. Some of these are print(), dir

195. How Do We Write A Function In Python?
Ans: We can create a Python function in the following manner.

Step-1: to begin the function, start writing with the keyword def and then mention the functior

Step-2: We can now pass the arguments and enclose them using the parentheses. A colon, in the e

Step-3: After pressing an enter, we can add the desired Python statements for execution.

196. What Is A Function Call Or A Callable Object In Python?
Ans: A function in Python gets treated as a callable object. It can allow some arguments and al

197. What Is The Return Keyword Used For In Python?
Ans: The purpose of a function is to receive the inputs and return some output.

The return is a Python statement which we can use in a function for sending a value back to its

198. What Is “Call By Value” In Python?
Ans: In call-by-value, the argument whether an expression or a value gets bound to the respecti

Python will treat that variable as local in the function-level scope. Any changes made to that

199. What Is “Call By Reference” In Python?
Ans: We use both “call-by-reference” and “pass-by-reference” interchangeably. When we pass an a

This scheme also has the advantage of bringing more time and space efficiency because it leaves

On the contrary, the disadvantage could be that a variable can get changed accidentally during

200. What Is The Return Value Of The Trunc() Function?
Ans: The Python trunc() function performs a mathematical operation to remove the decimal values
201. Is It Mandatory For A Python Function To Return A Value?
Ans: It is not at all necessary for a function to return any value. However, if needed, we can

202. What Does The Continue Do In Python?
Ans: The continue is a jump statement in Python which moves the control to execute the next ite

The continue statement is applicable for both the “while” and “for” loops.

203. What Is The Purpose Of Id() Function In Python?
Ans: The id() is one of the built-in functions in Python.

Signature: id(object)
It accepts one parameter and returns a unique identifier associated with the input object.

204. What Does The *Args Do In Python?
Ans: We use *args as a parameter in the function header. It gives us the ability to pass N (var

Please note that this type of argument syntax doesn’t allow passing a named argument to the fur

Example of using the *args:

```
# Python code to demonstrate
# *args for dynamic arguments
def fn(*argList):
    for argx in argList:
        print (argx)

fn('I', 'am', 'Learning', 'Python')
```

The output:

```
I
am
Learning
Python
```

205.Does Python Have A Main() Method?
Ans: The main() is the entry point function which happens to be called first in most programmin

Since Python is interpreter-based, so it sequentially executes the lines of the code one-by-one

Python also does have a Main() method. But it gets executed whenever we run our Python script e

We can also override the Python default main() function using the Python if statement. Please s

```
print("Welcome")
print("__name__ contains: ", __name__)
def main():
    print("Testing the main function")
if __name__ == '__main__':
    main()
```

The output:

```
Welcome
__name__ contains:  __main__
Testing the main function
```

206. What Does The __ Name __ Do In Python?
Ans: The __name__ is a unique variable. Since Python doesn’t expose the main() function, so whe

To see whether the main() gets called, we can use the __name__ variable in an if clause compare

207. What Is The Purpose Of “End” In Python?
Ans: Python’s print() function always prints a newline in the end. The print() function accepts

```
# Example: Print a  instead of the new line in the end.
print("Let's learn" , end = ' ')
print("Python")

# Printing a dot in the end.
print("Learn to code from techbeamers" , end = '.')
print("com", end = ' ')
```

The output is:

```
Let's learn Python
Learn to code from techbeamers.com
```

208. When Should You Use The “Break” In Python?
Ans: Python provides a break statement to exit from a loop. Whenever the break hits in the code

The break statement in a nested loop causes the control to exit from the inner iterative block

209.What Is The Difference Between Pass And Continue In Python?
Ans: The continue statement makes the loop to resume from the next iteration.

On the contrary, the pass statement instructs to do nothing, and the remainder of the code exec

210. What Does The Len() Function Do In Python?
Ans: In Python, the len() is a primary string function. It determines the length of an input st

```
>>> some_string = 'techbeamers'
>>> len(some_string)
11
```

211. What Does The Chr() Function Do In Python?
Ans: The chr() function got re-added in Python 3.2. In version 3.0, it got removed.

It returns the string denoting a character whose Unicode code point is an integer.

For example, the chr(122) returns the string ‘z’ whereas the chr(1212) returns the string ‘ҽ’.

212. What Does The Ord() Function Do In Python?
Ans: The ord(char) in Python takes a string of size one and returns an integer denoting the Uni

```
>>> ord("z")
122
```

213. What Is Rstrip() In Python?
Ans: Python provides the rstrip() method which duplicates the string but leaves out the whitesp

The rstrip() escapes the characters from the right end based on the argument value, i.e., a str

The signature of the rstrip() is:

```
2009
2010  str.rstrip([char sequence/pre>
2011  #Example
2012  test_str = 'Programming    '
2013  # The trailing whitespaces are excluded
2014  print(test_str.rstrip())
2015
2016
2017  214.What Is Whitespace In Python?
2018  Ans: Whitespace represents the characters that we use for spacing and separation.
2019
2020  They possess an “empty” representation. In Python, it could be a tab or space.
2021
2022  215. What Is Isalpha() In Python?
2023  Ans: Python provides this built-in isalpha() function for the string handling purpose.
2024
2025  It returns True if all characters in the string are of alphabet type, else it returns False.
2026
2027  216. How Do You Use The Split() Function In Python?
2028  Python’s split() function works on strings to cut a large piece into smaller chunks, or sub-str
2029
2030  #Example
2031  str = 'pdf csv json'
2032  print(str.split(" "))
2033  print(str.split())
2034  The output:
2035
2036  ['pdf', 'csv', 'json']
2037  ['pdf', 'csv', 'json']
2038  217. What Does The Join Method Do In Python?
2039  Ans: Python provides the join() method which works on strings, lists, and tuples. It combines t
2040
2041  218. What Does The Title() Method Do In Python?
2042  Ans: Python provides the title() method to convert the first letter in each word to capital for
2043
2044  #Example
2045  str = 'lEaRn pYtHoN'
2046  print(str.title())
2047  The output:
2048
2049  Learn Python
2050  Now, check out some general purpose Python interview questions.
2051
2052  219. What Makes The CPython Different From Python?
2053  Ans: CPython has its core developed in C. The prefix ‘C’ represents this fact. It runs an inter
2054
2055  220. Which Package Is The Fastest Form Of Python?
2056  Ans: PyPy provides maximum compatibility while utilizing CPython implementation for improving i
2057
2058  The tests confirmed that PyPy is nearly five times faster than the CPython. It currently suppor
2059
2060  221. What Is GIL In Python Language?
```

```
Ans: Python supports GIL (the global interpreter lock) which is a mutex used to secure access t

222. How Is Python Thread Safe?
Ans: Python ensures safe access to threads. It uses the GIL mutex to set synchronization. If a

For example, many of the Python operations execute as atomic such as calling the sort() method

223. How Does Python Manage The Memory?
Ans: Python implements a heap manager internally which holds all of its objects and data struct

This heap manager does the allocation/de-allocation of heap space for objects.



224.What Is The Set Object In Python?
Ans: Sets are unordered collection objects in Python. They store unique and immutable objects.

225. What Is The Use Of The Dictionary In Python?
Ans: A dictionary has a group of objects (the keys) map to another group of objects (the values

They are mutable and hence will not change. The values associated with the keys can be of any P

226. Is Python List A Linked List?
Ans: A Python list is a variable-length array which is different from C-style linked lists.

Internally, it has a contiguous array for referencing to other objects and stores a pointer to

Here are some Python interview questions on classes and objects



227.What Is Class In Python?
Ans: Python supports object-oriented programming and provides almost all OOP features to use in

A Python class is a blueprint for creating the objects. It defines member variables and gets th

We can make it by using the keyword “class.” An object gets created from the constructor. This

In Python, we generate classes and instances in the following way.

>>>class Human:  # Create the class
...     pass
>>>man = Human()  # Create the instance
>>>print(man)
<__main__.Human object at 0x0000000003559E10>
228. What Are Attributes And Methods In A Python Class?
Ans: A class is useless if it has not defined any functionality. We can do so by adding attribu

>>> class Human:
...     profession = "programmer" # specify the attribute 'profession' of the class
>>> man = Human()
>>> print(man.profession)
```

```
programmer
After we added the attributes, we can go on to define the functions. Generally, we call them me

>>> class Human:
    profession = "programmer"
    def set_profession(self, new_profession):
        self.profession = new_profession
>>> man = Human()
>>> man.set_profession("Manager")
>>> print(man.profession)
Manager
229. How To Assign Values For The Class Attributes At Runtime?
Ans: We can specify the values for the attributes at runtime. We need to add an init method and

>>> class Human:
    def __init__(self, profession):
        self.profession = profession
    def set_profession(self, new_profession):
        self.profession = new_profession

>>> man = Human("Manager")
>>> print(man.profession)
Manager
 230.What Is Inheritance In Python Programming?
Ans: Inheritance is an OOP mechanism which allows an object to access its parent class features

Python Interview Questions - Inheritance

We do it intentionally to abstract away the similar code in different classes.

The common code rests with the base class, and the child class objects can access it via inheri

class PC: # Base class
    processor = "Xeon" # Common attribute
    def set_processor(self, new_processor):
        processor = new_processor

class Desktop(PC): # Derived class
    os = "Mac OS High Sierra" # Personalized attribute
    ram = "32 GB"

class Laptop(PC): # Derived class
    os = "Windows 10 Pro 64" # Personalized attribute
    ram = "16 GB"

desk = Desktop()
print(desk.processor, desk.os, desk.ram)

lap = Laptop()
print(lap.processor, lap.os, lap.ram)
The output:

```

```
Xeon Mac OS High Sierra 32 GB
Xeon Windows 10 Pro 64 16 GB


231.What Is Composition In Python?
Ans: The composition is also a type of inheritance in Python. It intends to inherit from the ba

See the below diagram.

Python Interview Questions - Composition

To demonstrate composition, we need to instantiate other objects in the class and then make use

class PC: # Base class
    processor = "Xeon" # Common attribute
    def __init__(self, processor, ram):
        self.processor = processor
        self.ram = ram

    def set_processor(self, new_processor):
        processor = new_processor

    def get_PC(self):
        return "%s cpu & %s ram" % (self.processor, self.ram)

class Tablet():
    make = "Intel"
    def __init__(self, processor, ram, make):
        self.PC = PC(processor, ram) # Composition
        self.make = make

    def get_Tablet(self):
        return "Tablet with %s CPU & %s ram by %s" % (self.PC.processor, self.PC.ram, self.make

if __name__ == "__main__":
    tab = Tablet("i7", "16 GB", "Intel")
    print(tab.get_Tablet())
The output is:

Tablet with i7 CPU & 16 GB ram by Intel



232.What Are Errors And Exceptions In Python Programs?
Ans: Errors are coding issues in a program which may cause it to exit abnormally.

On the contrary, exceptions happen due to the occurrence of an external event which interrupts

233. How Do You Handle Exceptions With Try/Except/Finally In Python?
Ans: Python lay down Try, Except, Finally constructs to handle errors as well as Exceptions. We

try:
```

```
    print("Executing code in the try block")
    print(exception)
except:
    print("Entering in the except block")
finally:
    print("Reached to the final block")
The output is:

Executing code in the try block
Entering in the except block
Reached to the final block
234. How Do You Raise Exceptions For A Predefined Condition In Python?
Ans: We can raise an exception based on some condition.

For example, if we want the user to enter only odd numbers, else will raise an exception.

# Example - Raise an exception
while True:
    try:
        value = int(input("Enter an odd number- "))
        if value%2 == 0:
            raise ValueError("Exited due to invalid input!!!")
        else:
            print("Value entered is : %s" % value)
    except ValueError as ex:
        print(ex)
        break
The output is:

Enter an odd number- 2
Exited due to invalid input!!!
Enter an odd number- 1
Value entered is : 1
Enter an odd number-
235. What Are Python Iterators?
Ans: Iterators in Python are array-like objects which allow moving on the next element. We use

Python library has a no. of iterators. For example, a list is also an iterator and we can start

236. What Is The Difference Between An Iterator And Iterable?
Ans: The collection type like a list, tuple, dictionary, and set are all iterable objects where

Here are some advanced-level Python interview questions.

237. What Are Python Generators?
Ans: A Generator is a kind of function which lets us specify a function that acts like an itera

In a generator function, the yield keyword substitutes the return statement.

# Simple Python function
def fn():
    return "Simple Python function."
```

```
2269
2270  # Python Generator function
2271  def generate():
2272      yield "Python Generator function."
2273
2274  print(next(generate()))
2275  The output is:
2276
2277  Python Generator function.
2278
2279
2280  238.What Are Closures In Python?
2281  Ans: Python closures are function objects returned by another function. We use them to eliminat
2282
2283  In the example below, we've written a simple closure for multiplying numbers.
2284
2285  def multiply_number(num):
2286      def product(number):
2287          'product() here is a closure'
2288          return num * number
2289      return product
2290
2291  num_2 = multiply_number(2)
2292  print(num_2(11))
2293  print(num_2(24))
2294
2295  num_6 = multiply_number(6)
2296  print(num_6(1))
2297  The output is:
2298
2299  22
2300  48
2301  6
2302  239. What Are Decorators In Python?
2303  Ans: Python decorator gives us the ability to add new behavior to the given objects dynamically
2304
2305  def decorator_sample(func):
2306      def decorator_hook(*args, **kwargs):
2307          print("Before the function call")
2308          result = func(*args, **kwargs)
2309          print("After the function call")
2310          return result
2311      return decorator_hook
2312
2313  @decorator_sample
2314  def product(x, y):
2315      "Function to multiply two numbers."
2316      return x * y
2317
2318  print(product(3, 3))
2319  The output is:
2320
```

```
Before the function call
After the function call
9
240. How Do You Create A Dictionary In Python?
Ans: Let’s take the example of building site statistics. For this, we first need to break up th

However, we can take values of any kind. For distinguishing the data pairs, we can use a comma(

 >>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> type(site_stats)
<class 'dict'>
>>> print(site_stats)
{'type': 'organic', 'site': 'tecbeamers.com', 'traffic': 10000}
241. How Do You Read From A Dictionary In Python?
Ans:To fetch data from a dictionary, we can directly access using the keys. We can enclose a “k

>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> print(site_stats["traffic"])
We can even call the get method to fetch the values from a dict. It also let us set a default v

>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> print(site_stats.get('site'))
tecbeamers.com
242. How Do You Traverse Through A Dictionary Object In Python?
Ans: We can use the “for” and “in” loop for traversing the dictionary object.

>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> for k, v in site_stats.items():
    print("The key is: %s" % k)
    print("The value is: %s" % v)
    print("++++++++++++++++++++++++")
The output is:

The key is: type
The value is: organic
++++++++++++++++++++++++
The key is: site
The value is: tecbeamers.com
++++++++++++++++++++++++
The key is: traffic
The value is: 10000
++++++++++++++++++++++++
243. How Do You Add Elements To A Dictionary In Python?
Ans: We can add elements by modifying the dictionary with a fresh key and then set the value to

>>> # Setup a blank dictionary
>>> site_stats = {}
>>> site_stats['site'] = 'google.com'
>>> site_stats['traffic'] = 10000000000
>>> site_stats['type'] = 'Referral'
>>> print(site_stats)
{'type': 'Referral', 'site': 'google.com', 'traffic': 10000000000}
```

```
We can even join two dictionaries to get a bigger dictionary with the help of the update() meth

>>> site_stats['site'] = 'google.co.in'
>>> print(site_stats)
{'site': 'google.co.in'}
>>> site_stats_new = {'traffic': 1000000, "type": "social media"}
>>> site_stats.update(site_stats_new)
>>> print(site_stats)
{'type': 'social media', 'site': 'google.co.in', 'traffic': 1000000}
244. How Do You Delete Elements Of A Dictionary In Python?
Ans: We can delete a key in a dictionary by using the del() method.

>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> del site_stats["type"]
>>> print(site_stats)
{'site': 'google.co.in', 'traffic': 1000000}
Another method, we can use is the pop() function. It accepts the key as the parameter. Also, a

>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> print(site_stats.pop("type", None))
organic
>>> print(site_stats)
{'site': 'tecbeamers.com', 'traffic': 10000}
245. How Do You Check The Presence Of A Key In A Dictionary?
Ans: We can use Python's "in" operator to test the presence of a key inside a dict object.

>>> site_stats = {'site': 'tecbeamers.com', 'traffic': 10000, "type": "organic"}
>>> 'site' in site_stats
True
>>> 'traffic' in site_stats
True
>>> "type" in site_stats
True
Earlier, Python also provided the has_key() method which got deprecated.

246. What Is The Syntax For List Comprehension In Python?
Ans: The signature for the list comprehension is as follows:

[ expression(var) for var in iterable ]
For example, the below code will return all the numbers from 10 to 20 and store them in a list

>>> alist = [var for var in range(10, 20)]
>>> print(alist)
247. What Is The Syntax For Dictionary Comprehension In Python?
A dictionary has the same syntax as was for the list comprehension but the difference is that i

{ aKey, itsValue for aKey in iterable }
For example, the below code will return all the numbers 10 to 20 as the keys and will store the

>>> adict = {var:var**2 for var in range(10, 20)}
>>> print(adict)
248. What Is The Syntax For Generator Expression In Python?
```

Ans: The syntax for generator expression matches with the list comprehension, but the differenc

(expression(var) for var in iterable)
For example, the below code will create a generator object that generates the values from 10 to

>>> (var for var in range(10, 20))
 at 0x0000000003668728>
>>> list((var for var in range(10, 20)))
Now, see more Python interview questions for practice.

249. How Do You Write A Conditional Expression In Python?
Ans: We can utilize the following single statement as a conditional expression. default_statmer

>>> no_of_days = 366
>>> is_leap_year = "Yes" if no_of_days == 366 else "No"
>>> print(is_leap_year)
Yes
250. What Do You Know About The Python Enumerate?
Ans: While using the iterators, sometimes we might have a use case to store the count of iterat

The enumerate() function attaches a counter variable to an iterable and returns it as the “enum

We can use this object directly in the “for” loops or transform it into a list of tuples by cal

enumerate(iterable, to_begin=0)
Arguments:
iterable: array type object which enables iteration
to_begin: the base index for the counter is to get started, its default value is 0
# Example - enumerate function
alist = ["apple","mango", "orange"]
astr = "banana"

# Let's set the enumerate objects
list_obj = enumerate(alist)
str_obj = enumerate(astr)

print("list_obj type:", type(list_obj))
print("str_obj type:", type(str_obj))

print(list(enumerate(alist)))
# Move the starting index to two from zero
print(list(enumerate(astr, 2)))
The output is:

list_obj type: <class 'enumerate'>
str_obj type: <class 'enumerate'>
[(0, 'apple'), (1, 'mango'), (2, 'orange')]
[(2, 'b'), (3, 'a'), (4, 'n'), (5, 'a'), (6, 'n'), (7, 'a')]
251. What Is The Use Of Globals() Function In Python?
Ans: The globals() function in Python returns the current global symbol table as a dictionary c

Python maintains a symbol table to keep all necessary information about a program. This info ir

```
2477
2478  All the information in this table remains in the global scope of the program and Python allows
2479
2480  Signature: globals()
2481
2482  Arguments: None
2483  # Example: globals() function
2484  x = 9
2485  def fn():
2486      y = 3
2487      z = y + x
2488      # Calling the globals() method
2489      z = globals()['x'] = z
2490      return z
2491
2492  # Test Code
2493  ret = fn()
2494  print(ret)
2495  The output is:
2496
2497  12
2498  252. Why Do You Use The Zip() Method In Python?
2499  Ans: The zip method lets us map the corresponding index of multiple containers so that we can u
2500
2501  Signature:
2502   zip(*iterators)
2503  Arguments:
2504   Python iterables or collections (e.g., list, string, etc.)
2505  Returns:
2506   A single iterator object with combined mapped values
2507  # Example: zip() function
2508
2509  emp = [ "tom", "john", "jerry", "jake" ]
2510  age = [ 32, 28, 33, 44 ]
2511  dept = [ 'HR', 'Accounts', 'R&D', 'IT' ]
2512
2513  # call zip() to map values
2514  out = zip(emp, age, dept)
2515
2516  # convert all values for printing them as set
2517  out = set(out)
2518
2519  # Displaying the final values
2520  print ("The output of zip() is : ",end="")
2521  print (out)
2522  The output is:
2523
2524  The output of zip() is : {('jerry', 33, 'R&D'), ('jake', 44, 'IT'), ('john', 28, 'Accounts'), (
2525  253. What Are Class Or Static Variables In Python Programming?
2526  Ans: In Python, all the objects share common class or static variables.
2527
2528  But the instance or non-static variables are altogether different for different objects.
```

```
2529
2530 The programming languages like C++ and Java need to use the static keyword to make a variable a
2531
2532 All names initialized with a value in the class declaration becomes the class variables. And th
2533
2534 # Example
2535 class Test:
2536     aclass = 'programming' # A class variable
2537     def __init__(self, ainst):
2538         self.ainst = ainst # An instance variable
2539
2540 # Objects of CSStudent class
2541 test1 = Test(1)
2542 test2 = Test(2)
2543
2544 print(test1.aclass)
2545 print(test2.aclass)
2546 print(test1.ainst)
2547 print(test2.ainst)
2548
2549 # A class variable is also accessible using the class name
2550 print(Test.aclass)
2551 The output is:
2552
2553 programming
2554 programming
2555 1
2556 2
2557 programming
2558 Let’s now answer some advanced-level Python interview questions.
2559
2560 254. How Does The Ternary Operator Work In Python?
2561 Ans: The ternary operator is an alternative for the conditional statements. It combines true or
2562
2563 The syntax would look like the one given below.
2564
2565 [onTrue] if [Condition] else [onFalse]
2566
2567 x, y = 35, 75
2568 smaller = x if x < y else y
2569 print(smaller)
2570 255. What Does The “Self” Keyword Do?
2571 Ans: The self is a Python keyword which represents a variable that holds the instance of an obj
2572
2573 In almost, all the object-oriented languages, it is passed to the methods as a hidden parameter
2574
2575 256. What Are The Different Methods To Copy An Object In Python?
2576 Ans: There are two ways to copy objects in Python.
2577
2578 copy.copy() function
2579 It makes a copy of the file from source to destination.
2580 It’ll return a shallow copy of the parameter.
```

```
copy.deepcopy() function
It also produces the copy of an object from the source to destination.
It'll return a deep copy of the parameter that you can pass to the function.
257: What Is The Purpose Of Docstrings In Python?
Ans: In Python, the docstring is what we call as the docstrings. It sets a process of recording

258. Which Python Function Will You Use To Convert A Number To A String?
Ans: For converting a number into a string, you can use the built-in function str().  If you wa



259. How Do You Debug A Program In Python? Is It Possible To Step Through The Python Code?
Ans: Yes, we can use the Python debugger (pdb) to debug any Python program. And if we start a p

260. List Down Some Of The PDB Commands For Debugging Python Programs?
Ans: Here are a few PDB commands to start debugging Python code.

Add breakpoint (b)
Resume execution (c)
Step by step debugging (s)
Move to the next line (n)
List source code (l)
Print an expression (p)
261. What Is The Command To Debug A Python Program?
Ans: The following command helps run a Python program in debug mode.

$ python -m pdb python-script.py
262. How Do You Monitor The Code Flow Of A Program In Python?
Ans: In Python, we can use the sys module's settrace() method to setup trace hooks and monitor

You need to define a trace callback method and pass it to the settrace() function. The callback

import sys

def trace_calls(frame, event, arg):
    # The 'call' event occurs before a function gets executed.
    if event != 'call':
        return
    # Next, inspect the frame data and print information.
    print 'Function name=%s, line num=%s' % (frame.f_code.co_name, frame.f_lineno)
    return

def demo2():
    print 'in demo2()'

def demo1():
    print 'in demo1()'
    demo2()

sys.settrace(trace_calls)
demo1()

```

263. How long can an identifier be in Python?
Ans: According to the official Python documentation, an identifier can be of any length. Howeve

Apart from that, there are certain rules we must follow to name one:

According to the official Python documentation, an identifier can be of any length. However, PE

Apart from that, there are certain rules we must follow to name one:

It can only begin with an underscore or a character from A-Z or a-z.
The rest of it can contain anything from the following: A-Z/a-z/_/0-9.
Python is case-sensitive, as we discussed in the previous question.
Keywords cannot be used as identifiers. Python has the following keywords:

```
and     def     False   import  not     True
as      del     finally in      or      try
assert  elif    for     is      pass    while
break   else    from    lambda  print   with
class   except  global  None    raise   yield
continue        exec    if      nonlocal        return
```

264. How would you convert a string into lowercase?
Ans: We use the lower() method for this.

```
>>> ‘AyuShi’.lower()
‘ayushi’
```

To convert it into uppercase, then, we use upper().

```
>>> ‘AyuShi’.upper()
‘AYUSHI’
```

Also, to check if a string is in all uppercase or all lowercase, we use the methods isupper() a

```
>>> ‘AyuShi’.isupper()
False

>>> ‘AYUSHI’.isupper()
True

>>> ‘ayushi’.islower()
True

>>> ‘@yu$hi’.islower()
True

>>> ‘@YU$HI’.isupper()
True
```

So, characters like @ and $ will suffice for both cases

Also, istitle() will tell us if a string is in title case.

>>> ‘The Corpse Bride’.istitle()
True

265. What is the pass statement in Python?
There may be times in our code when we haven’t decided what to do yet, but we must type somethi

>>> def func(*args):
pass
>>>
Similarly, the break statement breaks out of a loop.

>>> for i in range(7):
if i==3: break
print(i)
1

2

Finally, the continue statement skips to the next iteration.

>>> for i in range(7):
if i==3: continue
print(i)
1

2

4

5

6

266. Explain help() and dir() functions in Python?
Ans:

The help() function displays the documentation string and help for its argument.

>>> import copy
>>> help(copy.copy)
Help on function copy in module copy:

copy(x)

```
Shallow copy operation on arbitrary Python objects.

See the module's __doc__ string for more info.

The dir() function displays all the members of an object(any kind).

>>> dir(copy.copy)
['__annotations__', '__call__', '__class__', '__closure__', '__code__', '__defaults__', '__dela



267. How do you get a list of all the keys in a dictionary?
Ans:For this, we use the function keys().

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

268. How will you check if all characters in a string are alphanumeric?
Ans: For this, we use the method isalnum().



269. With Python, how do you find out which directory you are currently in?
Ans:  To find this, we use the function/method getcwd(). We import it from the module os.

>>> import os
>>> os.getcwd()
'C:\\Users\\lifei\\AppData\\Local\\Programs\\Python\\Python36-32'

>>> type(os.getcwd)
<class 'builtin_function_or_method'>

We can also change the current working directory with chdir().

>>> os.chdir('C:\\Users\\lifei\\Desktop')
>>> os.getcwd()
'C:\\Users\\lifei\\Desktop'



270. How do you insert an object at a given index in Python?
Ans: Let's build a list first.

>>> a=[1,2,4]
Now, we use the method insert. The first argument is the index at which to insert, the second i

>>> a.insert(2,3)
>>> a
[1, 2, 3, 4]
```

271. how do you reverse a list?
Ans: Using the reverse() method.

```
>>> a.reverse()
>>> a
[4, 3, 2, 1]
```

You can also do it via slicing from right to left:

```
>>> a[::-1]
>>> a
[1, 2, 3, 4]
```

This gives us the original list because we already reversed it once. However, this does not mod

272. How does a function return values?
Ans: A function uses the ‘return’ keyword to return a value. Take a look:

```
>>> def add(a,b):
return a+b
```

273. How would you define a block in Python?
Ans: For any kind of statements, we possibly need to define a block of code under them. Howeve

```
>>> if 3>1:
print(“Hello”)
print(“Goodbye”)
Hello

Goodbye
```

274. Will the do-while loop work if you don’t end it with a semicolon?
Ans: Trick question! Python does not support an intrinsic do-while loop. Secondly, to terminate

275. In one line, show us how you’ll get the max alphabetical character from a string.?
Ans: For this, we’ll simply use the max function.

```
>>> max(‘flyiNg’)
‘y’
```

The following are the ASCII values for all the letters of this string-

```
f- 102

l- 108

y- 121

i- 105

N- 78

g- 103

By this logic, try to explain the following line of code-

>>> max('fly{}iNg')
'}'

(Bonus: } – 125)



276. What is Python good for?
Ans: Python is a jack of many trades, check out Applications of Python to find out more.

Meanwhile, we'll say we can use it for:

Web and Internet Development
Desktop GUI
Scientific and Numeric Applications
Software Development Applications
Applications in Education
Applications in Business
Database Access
Network Programming
Games, 3D Graphics
Other Python Applications


277. Can you name ten built-in functions in Python and explain each in brief?
Ans: Ten Built-in Functions, you say? Okay, here you go.

complex()- Creates a complex number.

>>> complex(3.5,4)
(3.5+4j)

eval()- Parses a string as an expression.

>>> eval('print(max(22,22.0)-min(2,3))')
20

filter()- Filters in items for which the condition is true.
```

```
>>> list(filter(lambda x:x%2==0,[1,2,0,False]))
[2, 0, False]

format()- Lets us format a string.

>>> print("a={0} but b={1}".format(a,b))
a=2 but b=3

hash()- Returns the hash value of an object.

>>> hash(3.7)
644245917

hex()- Converts an integer to a hexadecimal.

>>> hex(14)
'0xe'

input()- Reads and returns a line of string.

>>> input('Enter a number')
Enter a number7

'7'
len()- Returns the length of an object.

>>> len('Ayushi')
6

locals()- Returns a dictionary of the current local symbol table.

>>> locals()
{'__name__': '__main__', '__doc__': None, '__package__': None, '__loader__': <class '_frozen_im

open()- Opens a file.

>>> file=open('tabs.txt')


278. How will you convert a list into a string?
Ans: We will use the join() method for this.

>>> nums=['one','two','three','four','five','six','seven']
>>> s=' '.join(nums)
>>> s
o/p= 'one two three four five six seven'



279. How will you remove a duplicate element from a list?
Ans: We can turn it into a set to do that.
```

```
>>> list=[1,2,1,3,4,2]
>>> set(list)
{1, 2, 3, 4}



280. Can you explain the life cycle of a thread?
Ans:

python scripting interview questions

To create a thread, we create a class that we make override the run method of the thread class
A thread that we just created is in the new state. When we make a call to start() on it, it for
When execution begins, the thread is in the running state.
Calls to methods like sleep() and join() make a thread wait. Such a thread is in the waiting/bl
When a thread is done waiting or executing, other waiting threads are sent for scheduling.
A running thread that is done executing terminates and is in the dead state.


281. Explain the //, %, and ** operators in Python?
Ans: The // operator performs floor division. It will return the integer part of the result on

>>> 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.

>>> 2**10
1024

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

>>> 13%7
6

>>> 3.5%1.5
0.5



282. What are membership operators?
Ans:  With the operators ‘in’ and ‘not in’, we can confirm if a value is a member in another.

>>> ‘me’ in ‘disappointment’
True

>>> ‘us’ not in ‘disappointment’
True

```

283. Explain identity operators in Python?
Ans: The operators ‘is’ and ‘is not’ tell us if two values have the same identity.

```
>>> 10 is ’10’
False

>>> True is not False
True
```

284. Finally, tell us about bitwise operators in Python?

Ans:

python interview questions for freshers

These operate on values bit by bit.

AND (&) This performs & on each bit pair.

```
>>> 0b110 & 0b010
2
```

OR (|) This performs | on each bit pair.

```
>>> 3|2
3
```

XOR (^) This performs an exclusive-OR operation on each bit pair.

```
>>> 3^2
1
```

Binary One’s Complement (~) This returns the one’s complement of a value.

```
>>> ~2
-3
```

Binary Left-Shift (<<) This shifts the bits to the left by the specified amount.

```
>>> 1<<2
4
```

Here, 001 was shifted to the left by two places to get 100, which is binary for 4.

Binary Right-Shift (>>)

```
>>> 4>>2
1
```

285. What data types does Python support?
Ans: Python provides us with five kinds of data types:

Numbers – Numbers use to hold numerical values.

```
>>> a=7.0
>>>
```

Strings – A string is a sequence of characters. We declare it using single or double quotes.

```
>>> title="Ayushi's Book"
```

Lists – A list is an ordered collection of values, and we declare it using square brackets.

```
>>> colors=['red','green','blue']
>>> type(colors)
<class 'list'>
```

Tuples – A tuple, like a list, is an ordered collection of values. The difference. However, is

```
>>> name=('Ayushi','Sharma')
>>> name[0]='Avery'
Traceback (most recent call last):

File "<pyshell#129>", line 1, in <module>

name[0]='Avery'

TypeError: 'tuple' object does not support item assignment
```

Dictionary – A dictionary is a data structure that holds key-value pairs. We declare it using c

```
>>> squares={1:1,2:4,3:9,4:16,5:25}
>>> type(squares)
<class 'dict'>

>>> type({})
<class 'dict'>
```

We can also use a dictionary comprehension:

```
>>> squares={x:x**2 for x in range(1,6)}
>>> squares
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```

286. How would you convert a string into an int in Python?
Ans: If a string contains only numerical characters, you can convert it into an integer using t

```
>>> int('227')
```

```
227

Let’s check the types:

>>> type(‘227’)
<class ‘str’>

>>> type(int(‘227’))
<class ‘int’>



287. How do you take input in Python?
Ans:  For taking input from the user, we have the function input(). In Python 2, we had another

The input() function takes, as an argument, the text to be displayed for the task:

>>> a=input(‘Enter a number’)
Enter a number7

But if you have paid attention, you know that it takes input in the form of a string.

>>> type(a)
<class ‘str’>

Multiplying this by 2 gives us this:

>>> a*=2
>>> a
’77’

So, what if we need to work on an integer instead?

We use the int() function for this.

>>> a=int(input(‘Enter a number’))
Enter a number7

Now when we multiply it by 2, we get this:

>>> a*=2
>>> a
14



288. What is a function?
Ans: When we want to execute a sequence of statements, we can give it a name. Let’s define a fu

>>> def greater(a,b):
return a is a>b else b
>>> greater(3,3.5)
```

3.5

289. What is recursion?
Ans: When a function makes a call to itself, it is termed recursion. But then, in order for it

Let’s take an example.

```
>>> def facto(n):
if n==1: return 1
return n*facto(n-1)
>>> facto(4)
24
```

290. What do you know about relational operators in Python?
Ans:

Top python interview questions with answers

Relational operators compare values.

Less than (<) If the value on the left is lesser, it returns True.

```
>>> ‘hi’<‘Hi’
False
```

Greater than (>) If the value on the left is greater, it returns True.

```
>>> 1.1+2.2>3.3
True
```

This is because of the flawed floating-point arithmetic in Python, due to hardware dependencies

Less than or equal to (<=) If the value on the left is lesser than or equal to, it returns True

```
>>> 3.0<=3
True
```

Greater than or equal to (>=) If the value on the left is greater than or equal to, it returns

```
>>> True>=False
True
```

Equal to (==) If the two values are equal, it returns True.

```
>>> {1,3,2,2}=={1,2,3}
True

Not equal to (!=) If the two values are unequal, it returns True.

>>> True!=0.1
True

>>> False!=0.1
True



291. What are assignment operators in Python?
Ans:



python coding interview questions

We can combine all arithmetic operators with the assignment symbol.

>>> a=7
>>> a+=1
>>> a
8

>>> a-=1
>>> a
7

>>> a*=2
>>> a
14

>>> a/=2
>>> a
7.0

>>> a**=2
>>> a
49.0

>>> a//=3
>>> a
16.0

>>> a%=4
>>> a
0.0
```

292. Explain logical operators in Python.?
Ans: We have three logical operators- and, or, not.

```
>>> False and True
False

>>> 7<7 or True
True

>>> not 2==2
False
```

293. What does the function zip() do?
Ans: One of the less common functions with beginners, zip() returns an iterator of tuples.

```
>>> list(zip([‘a’,‘b’,‘c’],[1,2,3]))
[(‘a’, 1), (‘b’, 2), (‘c’, 3)]
```

Here, it pairs items from the two lists and creates tuples with those. But it doesn’t have to b

```
>>> list(zip((‘a’,‘b’,‘c’),(1,2,3)))
[(‘a’, 1), (‘b’, 2), (‘c’, 3)]
```

294. How can you declare multiple assignments in one statement?
Ans: There are two ways to do this:

First –

```
>>> a,b,c=3,4,5 #This assigns 3, 4, and 5 to a, b, and c respectively
```

Second –

```
>>> a=b=c=3 #This assigns 3 to a, b, and c
```

295. If you are ever stuck in an infinite loop, how will you break out of it?
Ans: For this, we press Ctrl+C. This interrupts the execution. Let’s create an infinite loop to

```
>>> def counterfunc(n):
while(n==7):print(n)
>>> counterfunc(7)
```

```
7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

7

Traceback (most recent call last):

File “<pyshell#332>”, line 1, in <module>

counterfunc(7)

File “<pyshell#331>”, line 2, in counterfunc

while(n==7):print(n)

KeyboardInterrupt




296. How is a .pyc file different from a .py file?
Ans: While both files hold bytecode, .pyc is the compiled version of a Python file. It has plat
```

297. How many types of objects does Python support?
Ans: Immutable objects- Those which do not let us modify their contents. Examples of these will

```
>>> tuple=(1,2,4)
>>> tuple
(1, 2, 4)

>>> 2+4j
(2+4j)
```

Mutable objects – Those that let you modify their contents. Examples of these are lists, sets,

```
>>> [2,4,9]
[2, 4, 9]

>>> dict1={1:1,2:2}
>>> dict1
{1: 1, 2: 2}
```

While two equal immutable objects' reference variables share the same address, it is possible t

298. When is the else part of a try-except block executed?
Ans: In an if-else block, the else part is executed when the condition in the if-statement is F

299. Explain join() and split() in Python?
Ans:

1)join() lets us join characters from a string together by a character we specify.

```
>>> ‘,’.join(‘12345’)
‘1,2,3,4,5’
```

2) split() lets us split a string around the character we specify.

```
>>> ‘1,2,3,4,5’.split(‘,’)
[‘1’, ‘2’, ‘3’, ‘4’, ‘5’]
```

300. Explain a few methods to implement Functionally Oriented Programming in Python?
Ans:

Sometimes, when we want to iterate over a list, a few methods come in handy.

```
3413
3414  a. filter()
3415
3416  Filter lets us filter in some values based on conditional logic.
3417
3418  >>> list(filter(lambda x:x>5,range(8)))
3419  [6, 7]
3420
3421  b. map()
3422
3423  Map applies a function to every element in an iterable.
3424
3425  >>> list(map(lambda x:x**2,range(8)))
3426  [0, 1, 4, 9, 16, 25, 36, 49]
3427  c. reduce()
3428
3429  Reduce repeatedly reduces a sequence pair-wise until we reach a single value.
3430
3431  >>> from functools import reduce
3432  >>> reduce(lambda x,y:x-y,[1,2,3,4,5])
3433  -13
3434
3435  300. Explain a few methods to implement Functionally Oriented Programming in Python?
3436  Ans:
3437
3438  Sometimes, when we want to iterate over a list, a few methods come in handy.
3439
3440  a. filter()
3441
3442  Filter lets us filter in some values based on conditional logic.
3443
3444  >>> list(filter(lambda x:x>5,range(8)))
3445  [6, 7]
3446
3447  b. map()
3448
3449  Map applies a function to every element in an iterable.
3450
3451  >>> list(map(lambda x:x**2,range(8)))
3452  [0, 1, 4, 9, 16, 25, 36, 49]
3453  c. reduce()
3454
3455  Reduce repeatedly reduces a sequence pair-wise until we reach a single value.
3456
3457  >>> from functools import reduce
3458  >>> reduce(lambda x,y:x-y,[1,2,3,4,5])
3459  -13
3460
3461
3462
3463  301. Is del the same as remove()? What are they?
3464  Ans:
```

```
3465
3466  del and remove() are methods on lists/ ways to eliminate elements.
3467
3468  >>> list=[3,4,5,6,7]
3469  >>> del list[3]
3470  >>> list
3471  [3, 4, 5, 7]
3472
3473  >>> list.remove(5)
3474  >>> list
3475  [3, 4, 7]
3476
3477  While del lets us delete an element at a certain index, remove() lets us remove an element by i
3478
3479
3480
3481  302. Explain a few methods to implement Functionally Oriented Programming in Python?
3482  Ans: Sometimes, when we want to iterate over a list, a few methods come in handy.
3483
3484  a. filter()
3485
3486  Filter lets us filter in some values based on conditional logic.
3487
3488  >>> list(filter(lambda x:x>5,range(8)))
3489  [6, 7]
3490
3491  b. map()
3492
3493  Map applies a function to every element in an iterable.
3494
3495  >>> list(map(lambda x:x**2,range(8)))
3496  [0, 1, 4, 9, 16, 25, 36, 49]
3497  c. reduce()
3498
3499  Reduce repeatedly reduces a sequence pair-wise until we reach a single value.
3500
3501  >>> from functools import reduce
3502  >>> reduce(lambda x,y:x-y,[1,2,3,4,5])
3503  -13
3504
3505
3506
3507  304. How do you open a file for writing?
3508  Ans: Let’s create a text file on our Desktop and call it tabs.txt. To open it to be able to wri
3509
3510  >>> file=open(‘tabs.txt’,‘w’)
3511  This opens the file in writing mode. You should close it once you’re done.
3512
3513  >>> file.close()
3514
3515
3516
```

305. Difference between the append() and extend() methods of a list.
Ans: The methods append() and extend() work on lists. While append() adds an element to the enc

Let’s take two lists.

```
>>> list1,list2=[1,2,3],[5,6,7,8]
```
This is how append() works:

```
>>> list1.append(4)
>>> list1
[1, 2, 3, 4]
```

And this is how extend() works:

```
>>> list1.extend(list2)
>>> list1
[1, 2, 3, 4, 5, 6, 7, 8]
```

306. What are the different file-processing modes with Python?
We have the following modes-

read-only – ‘r’
write-only – ‘w’
read-write – ‘rw’
append – ‘a’
We can open a text file with the option ‘t’. So to open a text file to read it, we can use the

307. What does the map() function do?
Ans: map() executes the function we pass to it as the first argument; it does so on all element

```
>>> for i in map(lambda i:i**3, (2,3,7)):
print(i)
8
27
343
```

This gives us the cubes of the values 2, 3, and 7.

308. Is there a way to remove the last object from a list?
Yes, there is. Try running the following piece of code-

```
>>> list=[1,2,3,4,5
>>> list.pop(-1)
5

>>> list
[1, 2, 3, 4]
```

309. How will you convert an integer to a Unicode character?
Ans: This is simple. All we need is the chr(x) built-in function. See how.

```
>>> chr(52)
‘4’

>>> chr(49)
‘1’

>>> chr(67)
‘C’
```

310. So does recursion cause any trouble?
Ans: Sure does:

Needs more function calls.
Each function call stores a state variable to the program stack- consumes memory, can cause mem
Calling a function consumes time.

311. What good is recursion?
Ans: With recursion, we observe the following:

Need to put in less efforts.
Smaller code than that by loops.
Easier-to-understand code.

312. Can you remove the whitespaces from the string “aaa bbb ccc ddd eee”?
Ans: I can think of three ways to do this.

Using join-

```
>>> s=‘aaa bbb ccc ddd eee’
>>> s1=”.join(s.split())
>>> s1
‘aaabbbcccdddeee’
```

Using a list comprehension–

```
>>> s=‘aaa bbb ccc ddd eee’
>>> s1=str(”.join(([i for i in s if i!=‘ ‘])))
>>> s1
‘aaabbbcccdddeee’
```

Using replace()-

```
>>> s=‘aaa bbb ccc ddd eee’
>>> s1 = s.replace(‘ ‘,”)
>>> s1
‘aaabbbcccdddeee’
```

313. How do you get the current working directory using Python?
Ans: Working on software with Python, you may need to read and write files from various directo

```
>>> import os
>>> os.getcwd()
‘C:\\Users\\Raj\\AppData\\Local\\Programs\\Python\\Python37-32’
```

314. What are the file-related modules we have in Python?
Ans: We have the following libraries and modules that let us manipulate text and binary files o

os
os.path
shutil

315. Explain the uses of the modules sqlite3, ctypes, pickle, traceback, and itertools.
sqlite3- Helps with handling databases of type SQLite
ctypes- Lets create and manipulate C data types in Python
pickle- Lets put any data structure to external files
traceback- Allows extraction, formatting, and printing of stack traces
itertools– Supports working with permutations, combinations, and other useful iterables.

316. How will you print the contents of a file?
>>> try:
with open(‘tabs.txt’,‘r’) as f:
print(f.read())
except IOError:
print(“File not found”)

317. What is Virtualenv in Python?
Ans: virtualenv is a tool to create isolated Python environments. virtualenv creates a folder w

Install virtualenv via pip: $ pip install virtualenv.

318. What is the function of “self”?
Ans: Self is a variable that represents the instance of the object to itself. In most of the ob

Let’s say you have a class ClassA which contains a method methodA defined as:

def methodA(self, arg1, arg2): #do something
and ObjectA is an instance of this class.

Now when ObjectA.methodA(arg1, arg2) is called, python internally converts it for you as:

ClassA.methodA(ObjectA, arg1, arg2)
The self variable refers to the object itself.

319. What does the Python nonlocal statement do (in Python 3.0 and later)?
Ans: In short, it lets you assign values to a variable in an outer (but non-global) scope.

The nonlocal statement causes the listed identifiers to refer to previously bound variables in

For example the counter generator can be rewritten to use this so that it looks more like the i

```
def make_counter():
    count = 0
    def counter():
        nonlocal count
        count += 1
        return count
    return counter
```

320. What are the wheels and eggs? What is the difference?
Ans:

Wheel and Egg are both packaging formats that aim to support the use case of needing an install

The Egg format was introduced by setuptools in 2004, whereas the Wheel format was introduced by

Wheel is currently considered the standard for built and binary packaging for Python.

Here’s a breakdown of the important differences between Wheel and Egg.

Wheel has an official PEP. Egg did not.
Wheel is a distribution format, i.e a packaging format. 1 Egg was both a distribution format ar
Wheel archives do not include .pyc files. Therefore, when the distribution only contains Python
Wheel uses PEP376-compliant .dist-info directories. Egg used .egg-info.
Wheel has a richer file naming convention. A single wheel archive can indicate its compatibilit
Wheel is versioned. Every wheel file contains the version of the wheel specification and the im
Wheel is internally organized by sysconfig path type, therefore making it easier to convert to

321. What is webpack?
Ans: Webpack is a build tool that puts all of your assets, including Javascript, images, fonts,

322. Name some benefits of using webpack
Ans: Webpack and static assets in a dependency graph offers many benefits. Here’s a few:

Dead asset elimination. This is killer, especially for CSS rules. You only build the images and
Easier code splitting. For example, because you know that your file Homepage.js only requires s
You control how assets are processed. If an image is below a certain size, you could base64 enc
Stable production deploys. You can’t accidentally deploy code with images missing, or outdated
Webpack will slow you down at the start, but give you great speed benefits when used correctly
Webpack is the main build tool adopted by the React community.

323. Name some plugins you think are very important and helpful?
Ans:

CommonsChunkPlugin – creates a separate file (known as a chunk), consisting of common modules s
DefinePlugin – allows you to create global constants which can be configured at compile time.
HtmlWebpackPlugin – simplifies creation of HTML files to serve your webpack bundles.
ExtractTextWebpackPlugin – Extract text from a bundle, or bundles, into a separate file.
CompressionWebpackPlugin – Prepare compressed versions of assets to serve them with Content-Enc

324. Webpack gives us a dependency graph. What does that mean?
Ans: Any time one file depends on another, webpack treats this as a dependency. This allows we

Webpack lets you use require() on local “static assets”:

When webpack processes your application, it starts from a list of modules defined on the commar

The require('logo.png') source code never actually gets executed in the browser (nor in Node.js

325. What are metaclasses in Python?
Ans: A metaclass is the class of a class. A class defines how an instance of the class (i.e. ar

326. How to make a chain of function decorators?
Ans: How can I make two decorators in Python that would do the following?

```
@makebold
@makeitalic
def say():
   return "Hello"
```

which should return:

```
"<b><i>Hello</i></b>"
```

Answer:
Consider:

```
from functools import wraps

def makebold(fn):
    @wraps(fn)
    def wrapped(*args, **kwargs):
        return "<b>" + fn(*args, **kwargs) + "</b>"
    return wrapped

def makeitalic(fn):
    @wraps(fn)
    def wrapped(*args, **kwargs):
        return "<i>" + fn(*args, **kwargs) + "</i>"
    return wrapped

@makebold
@makeitalic
def hello():
    return "hello world"

@makebold
@makeitalic
def log(s):
    return s

print hello()        # returns "<b><i>hello world</i></b>"
print hello.__name__ # with functools.wraps() this returns "hello"
print log('hello')   # returns "<b><i>hello</i></b>"
```

327. What is the difference between @staticmethod and @classmethod?
Ans: A staticmethod is a method that knows nothing about the class or instance it was called or

```
3881
3882  class C:
3883      @staticmethod
3884      def f(arg1, arg2, ...): ...
3885  A classmethod, on the other hand, is a method that gets passed the class it was called on, or t
3886
3887  class C:
3888     @classmethod
3889     def f(cls, arg1, arg2, ...): ...
3890  If your method accesses other variables/methods in your class then use @classmethod.
3891
3892
3893
3894  328. What’s the difference between a Python module and a Python package?
3895  Ans: Any Python file is a module, its name being the file’s base name without the .py extensior
3896
3897  import my_module
3898  A package is a collection of Python modules: while a module is a single Python file, a package
3899
3900  Packages are modules too. They are just packaged up differently; they are formed by the combina
3901
3902  from my_package.timing.danger.internets import function_of_love
3903
3904
3905
3906
3907  329. What is GIL?
3908  Ans: Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that o
3909
3910
3911
3912
3913
3914  330. Is it a good idea to use multi-thread to speed your Python code?
3915  Ans:  Python doesn’t allow multi-threading in the truest sense of the word. It has a multi-thre
3916
3917  Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only o
3918
3919
3920
3921
3922
3923  331. How do I write a function with output parameters (call by reference)?
3924  Ans: In Python arguments are passed by assignment. When you call a function with a parameter, a
3925
3926  If you pass a mutable object into a method, the method gets a reference to that same object and
3927
3928  So to achieve the desired effect your best choice is to return a tuple containing the multiple
3929
3930  def func2(a, b):
3931      a = 'new-value'        # a and b are local names
3932      b = b + 1              # assigned to new objects
```

```
    return a, b            # return new values

x, y = 'old-value', 99
x, y = func2(x, y)
print(x, y)
```

332. Whenever you exit Python, is all memory de-allocated?
Ans: The answer here is no. The modules with circular references to other objects, or to objec

333. What is the purpose of the single underscore “_” variable in Python?
Ans: has 4 main conventional uses in Python:

To hold the result of the last executed expression(/statement) in an interactive interpreter se
For translation lookup in i18n (see the gettext documentation for example), as in code like: ra
As a general purpose “throwaway” variable name to indicate that part of a function result is be
As part of a function definition (using either def or lambda), where the signature is fixed (e

334. How is set() implemented internally?
I’ve seen people say that set objects in python have O(1) membership-checking. How are they imp

Ans:
Indeed, CPython’s sets are implemented as something like dictionaries with dummy values (the ke

So basically a set uses a hashtable as its underlying data structure. This explains the O(1) me

Also, it worth to mention when people say sets have O(1) membership-checking, they are talking

335. What is MRO in Python? How does it work?

Ans: Method Resolution Order (MRO) it denotes the way a programming language resolves a method

In Python,** method resolution order** defines the order in which the base classes are searched

Python resolves method and attribute lookups using the C3 linearisation of the class and its pa

336. What is the difference between old style and new style classes in Python?
Ans: Declaration-wise:

New-style classes inherit from object, or from another new-style class.

```
class NewStyleClass(object):
    pass

class AnotherNewStyleClass(NewStyleClass):
    pass
```

Old-style classes don’t.

```
class OldStyleClass():
    pass
```

Python 3 Note:

Python 3 doesn’t support old style classes, so either form noted above results in a new-style c

Also, MRO (Method Resolution Order) changed:

Classic classes do a depth first search from left to right. Stop on first match. They do not ha
New-style classes MRO is more complicated to synthesize in a single English sentence. One of it
Some other notes:

New style class objects cannot be raised unless derived from Exception.
Old style classes are still marginally faster for attribute lookup.

337. Why Python (CPython and others) uses the GIL?
Ans: In CPython, the global interpreter lock, or GIL, is a mutex that prevents multiple native

Python has a GIL as opposed to fine-grained locking for several reasons:

It is faster in the single-threaded case.

```
4037 It is faster in the multi-threaded case for i/o bound programs.
4038 It is faster in the multi-threaded case for cpu-bound programs that do their compute-intensive
4039 It makes C extensions easier to write: there will be no switch of Python threads except where y
4040 It makes wrapping C libraries easier. You don’t have to worry about thread-safety. If the libra
4041
4042
4043
4044
4045
4046
4047 338. How are arguments passed by value or by reference in python?
4048 Ans:
4049
4050 Pass by value: Copy of the actual object is passed. Changing the value of the copy of the objec
4051 Pass by reference: Reference to the actual object is passed. Changing the value of the new obje
4052 In Python, arguments are passed by reference, i.e., reference to the actual object is passed.
4053
4054 def appendNumber(arr):
4055     arr.append(4)
4056
4057 arr = [1, 2, 3]
4058
4059 print(arr)  #Output: => [1, 2, 3]
4060 appendNumber(arr)
4061 print(arr)  #Output: => [1, 2, 3, 4]
4062
4063
4064
4065
4066
4067
4068 339. What is a boolean in Python?
4069 Ans: Boolean is one of the built-in data types in Python, it mainly contains two values, and th
4070
4071 Python bool() is the method used to convert a value to a boolean value.
4072
4073 1
4074 Syntax for bool() method: bool([a])
4075
4076
4077
4078
4079
4080
4081 340. What is Python String format and Python String replace?
4082 Ans:  Python String Format: The String format() method in Python is mainly used to format the g
4083
4084 Syntax for String format() method:
4085
4086 1
4087 template.format(p0, p1, ..., k0=v0, k1=v1, ...)
4088 Python String Replace: This method is mainly used to return a copy of the string in which all t
```

Syntax for String replace() method:

```
str.replace(old, new [, count])
```

341. Name some of the built-in modules in Python?
Ans: The built-in modules in Python are:

sys module
OS module
random module
collection module
JSON
Math module

342. How do we convert the string to lowercase?
Ans: lower() function is used to convert string to lowercase.

Example:

```
str = 'XYZ'
print(str.lower())
```

Output:

```
xyz
```

343. How to remove values from a Python array?
Ans: The elements can be removed from a Python array using remove() or pop() function. The diff

Example:

```
1
2
3
4
5
x = arr.array('d',  [ 1.0, 2.2, 3.4, 4.8, 5.2, 6.6, 7.3])
print(x.pop())
print(x.pop(3))
x.remove(1.0)
print(a)
```

Output:

```
1
2
3
7.3
4.8
array(‘d’, [2.2, 3.4, 5.2, 6.6])
```

344. What is Try Block?
A block which is preceded by the try keyword is known as a try block

Syntax:

```
1
2
3
try{
   //statements that may cause an exception
}
```

345. How can we access a module written in Python from C?
Ans: We can access the module written in Python from C by using the following method.

```
1
```

```
Module == PyImport_ImportModule("<modulename>");
```

346. Write a program to count the number of capital letters in a file?
Ans:

```
1
2
3
4
5
6
with open(SOME_LARGE_FILE) as countletter:
count = 0
text = countletter.read()
for character in text:
if character.isupper():
count += 1
```

347. Write a program to display the Fibonacci sequence in Python?
Ans:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

```
# Displaying Fibonacci sequence
n = 10
# first two terms
n0 = 0
n1 = 1
#Count
x = 0
# check if the number of terms is valid
if n <= 0:
   print("Enter positive integer")
elif n == 1:
   print("Numbers in Fibonacci sequence upto",n,":")
   print(n0)
else:
   print("Numbers in Fibonacci sequence upto",n,":")
   while x < n:
       print(n0,end=', ')
       nth = n0 + n1
       n0 = n1
       n1 = nth
       x += 1
Output:

1
0, 1, 1, 2, 3, 5, 8, 13, 21, 34,
```

348. Write a program in Python to produce Star triangle?

Ans: The code to produce star triangle is as follows:

```
1
2
3
4
def pyfun(r):
for a in range(r):
print(' '*(r-x-1)+'*'*(2*x+1))
pyfun(9)
Output:

1
2
3
4
5
6
7
8
```

```
9
         *
        ***
       *****
      *******
     *********
    ***********
   *************
  ***************
*****************



```

349. Write a program to check whether the given number is prime or not?
Ans: The code to check prime number is as follows:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
# program to check the number is prime or not
n1 = 409
# num1 = int(input("Enter any one number: "))
# prime number is greater than 1
if n1 > 1:
# check the following factors
for x is in range of(2,num1):
if (n1 % x) == 0:
print(n1,"is not a prime number")
print(x,"times",n1//x,"is",num)
break
else:
print(n1,"is a prime number")
# if input number is smaller than
# or equal to the value 1, then it is not prime number
else:
print(n1,"is not a prime number")
Output:
```

```
4349
4350  1
4351  409 is a prime number
4352
4353
4354
4355
4356
4357
4358
4359
4360  350.  Write Python code to check the given sequence is a palindrome or not?
4361  Ans:
4362
4363  1
4364  2
4365  3
4366  4
4367  5
4368  6
4369  7
4370  8
4371  9
4372  10
4373  11
4374  # Python code to check a given sequence
4375  #  is palindrome or not
4376  my_string1 = 'MOM'
4377  My_string1 = my_string1.casefold()
4378  # reverse the given string
4379  rev_string1  = reversed(my_string1)
4380  # check whether the string is equal to the reverse of it or not
4381  if list(my_string1) == list(rev_string1):
4382  print("It is a palindrome")
4383  else:
4384  print("It is not a palindrome")
4385  Output:
4386
4387  1
4388  it is a palindrome
4389
4390
4391
4392
4393
4394
4395
4396
4397  351. Write Python code to sort a numerical dataset?
4398  Ans: The code to sort a numerical dataset is as follows:
4399
4400  1
```

```
2
3
4
list = [ "13", "16", "1", "5" , "8"]
list = [int(x) for x in the list]
list.sort()
print(list)
Output:

1
1, 5, 8, 13, 16
```