**Python interview question**

**Junior and middle questions.**

1.  What is the lambda function in Python? Why does it exist in Python?

2. What is pass in Python?

3. What is \*args, \*\*kwargs in function definition?

4. What is docstring in Python? How to write them? Are they required?

5. What are the built-in data types that Python provides? Which of them are mutable, which are immutable?

6. What is the difference between list and tuple types in Python?

7, What keywords can be used in conjunction with the for keyword?

8, What's the difference between globals(), locals(), and vars()?

9, Is it possible to have a negative index in iterative types in Python?

10. What is the \_\_init\_\_.py module? What it's for?

11. How can I swap values of variables in Python? Please give an example

12. How do I view object methods? Please give an example

13. What is a module in python? What is a package? What is the difference between packages and modules in Python? Please give an example module and package

14. What is the \_\_init\_\_ function used for?

15.  Explain how to make a Python script executable on Unix?

16. What is the output of -12 % 10 and -12 // 10.

17. Why shouldn't you make the default arguments an empty list?

18. What is the yield keyword used for in Python?

19. What is an iterator in Python? Can you write an example?

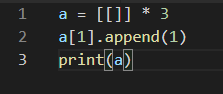
20. What is the difference between \_\_iter\_\_ and \_\_next\_\_?

21. What is unittest module in Python? How to write tests in Python?

22. What are metaclasses in Python?

**Junior and middle code involving**

1. Write a function that produces the Fibonacci sequence.
2. How to translate a string containing a binary code (1 and 0) into a number (integer)? Write a function to do this.
3. How to check that tuple A contains all elements of tuple B. Do both tuples contain unique values? Write a function to do this.
4. How to convert a string to a number that consists of letters ASCII code. Example: 'abcd' -> 979899100. Write a function to do this.
5. How to remove empty lines from a list of lines (with a length of 0). Write a function to do this.
6. Write a function that returns a string of numbers from 0 to 100, "0123456789101112...".
7. Write a function that makes a list with unique items from a list with duplicate items. Example: [1, 1, 2, 3, 3] -> [1, 2, 3]
8. Write a program that prints the numbers from 1 to 20. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers that are multiples of both three and five print “FizzBuzz”.
9. What will be the output of the following code? Please explain why?



1. We have the following code with the unknown function f(). In f(), we do not want to use a return, instead, we may want to use a generator.



The output looks like this:



Write a function f() so that we can have the output above.

**ANSWERS**

**Junior and middle questions:**

1. Lambda function in Python is a small anonymous function which can take any number of arguments, but can only have one expression.

Lambda function was born to reduce the number of rewrites of the expression.

2. The pass statement is a null statement. We use the pass statement to construct a body that does nothing.

3.

“\*args” is used for transferring an unknown number of parameters for a function. If the function has some specific parameters, then \*args means “other parameters”. \*args has to be the last parameter in the function since parameters after \*args will never be transferred into the function.

“\*\*kwarg” is used for transferring an unknown number of named arguments or keyword arguments for a function. Name arguments or keyword arguments are defined with default values. In function, kwargs will consider each argument as “key” and its value will be “value” (same as dictionary type).

In addition, order of args and kwargs is

(1) Normal arguments

(2) \*args

(3) \*\*kwargs

4. Docstring in Python is used to describe the function/uses/purpose of the object/function.

Example:

def sample\_foo():

“”” The sample\_ foo() function need to be implemented.

Currently, this function does nothing.”””

Class Data:

“”” This class is used to hold Data objects information. “””

Docstring in Python is required.

5. Built-in data types in Python:

– Dict: mutable

– List: mutable

– Set: immutable but can add or remove items from it

– Frozenset: immutable

– Tuple: immutable

6. The main difference between “list” and “tuple” is the “list” are mutable whereas “tuple” is immutable.

7. The “for” keyword is used to create a “for loop”.

– Use the “break” keyword to break out of a loop

– Use the “continue” keyword to end the current iteration, but continue with the next

8.

– globals() always returns the dictionary of the global namespace

– locals() always returns the dictionary of the current namespace

– vars() returns either a dictionary of the current namespace (if called with no argument) or the dictionary of the argument.

9. A negative index in iterative types in Python is possible.

Example:

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

array[-1] = 8

array[-2] = 7

array[-3] = 6

.............

10.

Each package has a \_\_init\_\_.py, it runs automatically when the import package.

The \_\_init\_\_.py file is intended to initialize what is needed for the package in case you import this package.

11.

– Using a temporary variable:

temp = x

x = y

y = temp

– Without Using Temporary Variable:

x, y = y, x

– Addition and Subtraction:

x = x + y

y = x - y

x = x - y

– Multiplication and Division:

x = x \* y

y = x / y

x = x / y

– XOR swap:

x = x ^ y

y = x ^ y

x = x ^ y

12.

To list the methods for this class, one approach is to use the dir() function in Python. The dir() function will return all functions and properties of the class.

Example:

class MyClass(object):

# MyClass property

property1 = [1, 2, 3]

def \_\_init\_\_(self, a):

assert isinstance(a, float) or isinstance(a, int)

self.state = a

def add(self, a):

assert isinstance(a, float) or isinstance(a, int)

self.state = self.state + a

return self.state

def subtract(self, a):

assert isinstance(a, float) or isinstance(a, int)

self.state = self.state - a

return self.state

def multiply(self, a):

assert isinstance(a, float) or isinstance(a, int)

self.state = self.state \* a

return self.state

def divide(self, a):

assert isinstance(a, float) or isinstance(a, int)

self.state = self.state / a

return self.state

method\_list = [method for method in dir(MyClass) if method.startswith('\_') is False]

print(method\_list)

Output:

['add', 'divide', 'multiply', 'subtract']

13.

– A module is a single file (or files) that are imported under one import and used.

Example:

import my\_module

– A package is a collection of modules in directories that give a package hierarchy.

Example:

from my\_package.timing.danger import foo

14. "\_\_init\_\_" is a reserved method in python classes. It is called as a constructor in object-oriented terminology. This method is called when an object is created from a class and it allows the class to initialize the attributes of the class.

15. On Unix it works usually in the following way:

1. Put #!/usr/bin/env python in the first line of your .py script.

2. Add execution permissions to the file (using chmod).

3. Execute the script from command line, eg. by providing ./my\_script.py when in the same directory.

OR

Just put this in the first line of your script:

1. #!/usr/bin/env python

Make the file executable with:

1. chmod +x myfile.py

Execute with:

1. ./myfile.py

16.

-12 % 10 = 8

-12 // 10 = 2

17. A new list is created once when the function is defined, and the same list is used in each successive call. Python’s default arguments are evaluated once when the function is defined, not each time the function is called.

def append\_to\_list(element, list\_to\_append=[]):

list\_to\_append.append(element)

return list\_to\_append

>>> a = append\_to\_list(10)

[10]

>>> b = append\_to\_list(20)

[10, 20]

18.

“yield” is a keyword in Python that is used to return from a function without destroying the states of its local variable and when the function is called, the execution starts from the last yield statement. Any function that contains a yield keyword is termed as generator. Hence, yield is what makes a generator.

19.

An iterator is an object that contains a countable number of values.

An iterator is an object that can be iterated upon, meaning that you can traverse through all the values.

Example:

mystr = "banana"

for x in mystr:

print(x)

20.

\_\_iter\_\_ is a method that is defined on iterables - things that you can iterate over (eg. lists). When you call it, it should return an iterator, which is a thing that does the iterating.

\_\_next\_\_ is a method of iterators. It tells it to give you the next item, and advance to the subsequent one.

21.

The unittest module provides a rich set of tools for constructing and running tests. This section demonstrates that a small subset of the tools suffice to meet the needs of most users.

Example:

import unittest

class TestStringMethods(unittest.TestCase):

def test\_upper(self):

self.assertEqual('foo'.upper(), 'FOO')

def test\_isupper(self):

self.assertTrue('FOO'.isupper())

self.assertFalse('Foo'.isupper())

def test\_split(self):

s = 'hello world'

self.assertEqual(s.split(), ['hello', 'world'])

# check that s.split fails when the separator is not a string

with self.assertRaises(TypeError):

s.split(2)

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

22.

A metaclass is the class of a class. A class defines how an instance of the class (i.e. an object) behaves while a metaclass defines how a class behaves. A class is an instance of a metaclass.

**Junior and middle code involving**

1.

*def* fibonacci\_to(*n*):

    fibs = [0, 1]

    for i in range(2, *n*+1):

        fibs.append(fibs[-1] + fibs[-2])

    return fibs

*def* main():

    order = input("Enter the order of the Fibonacci number: ")

    print(fibonacci\_to(int(order)))

if \_\_name\_\_ == '\_\_main\_\_':

    main()

2.

*def* main():

    string = input("Input string of binary number: ")

    print(int(string,2))

if \_\_name\_\_ == '\_\_main\_\_':

    main()

3.