1. What is Python?

Python is a programming language with objects, modules, threads, exceptions, and automatic memory management. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

2. What are the benefits of using Python?

The benefits of pythons are that it is simple and easy, portable, extensible, build-in data structure and it is open-source.

3. How is Python an interpreted language?

An interpreted language is any programming language that is not in machine-level code before runtime. Therefore, Python is an interpreted language.

4. How Python is interpreted?

Python language is an interpreted language. Python program runs directly from the source code. It converts the source code that is written by the programmer into an intermediate language, which is again translated into machine language that has to be executed.

5. How is memory managed in Python?

- Memory in Python is managed by Python private heap space. All Python objects and data structures are located in a private heap. This

private heap is taken care of by Python Interpreter itself, and a programmer doesn't have access to this private heap.

- Python memory manager takes care of the allocation of Python private heap space.
- Memory for Python private heap space is made available by Python's in-built garbage collector, which recycles and frees up all the unused memory.

6. What is pep 8?

PEP stands for Python Enhancement Proposal**.** It is a set of rules that specify how to format Python code for maximum readability.

7. How do you write comments in python?

Comments in Python start with a # character.

```
#Comment Example
```

8. How to comment on multiple lines in python?

Multi-line comments appear in more than one line. All the lines to be commented are to be prefixed by a #. You can also a very good shortcut method to comment on multiple lines. All you need to do is hold the ctrl key and left-click in every place wherever you want to include a # character and type a # just once. This will comment on all the lines where you introduced your cursor.

9. What are docstrings in Python?

Docstrings are not actually comments, but, they are *documentation strings*. These docstrings are within triple quotes. They are not assigned to

any variable and therefore, at times, serve the purpose of comments as well.

```
"""
This is Docstring example
It is useful for documentation purposes
"""
```

10. Is indentation optional in Python?

Indentation in Python is compulsory and is part of its syntax.

All programming languages have some way of defining the scope and extent of the block of codes; in Python, it is indentation. Indentation provides better readability to the code, which is probably why Python has made it compulsory.

11. What is a function in Python?

A function is a block of code that is executed only when it is called. To define a Python function, the def keyword is used. If the function returning something, they need a return keyword.

```
def example(a):
   return a*2
```

12. What are local variables and global variables in Python?

Global Variables:

Variables declared outside a function or in global space are called global variables. These variables can be accessed by any function in the program.

Local Variables:

Any variable declared inside a function is known as a local variable. This variable is present in the local space and not in the global space.

```
#Example of Global Variable
a = 1#Example of Local Variable
def sample():
   #Local Variable
   a = 1
```

13. What is a lambda function?

An anonymous or unnamed function is known as a lambda function. This function can have any number of parameters but, can have just one statement. It is often used as a one-time function rather than a function that used repeatedly.

```
#Example of Lambda Functiontest = lambda x,y: x*y
print(test(2,4))
```

14. Why lambda forms in python does not have statements?

A lambda form in python does not have statements as it is used to make new function object and then return them at runtime.

15. What are the supported data types in Python?

Python has five standard data types –

- Numbers (Integer and Float)
- String
- List
- Tuple
- Dictionary

16. What are indexes?

To access an element from ordered sequences, we simply use the index of the element, which is the position number of that particular element. The index usually starts from 0, i.e., the first element has index 0, the second has 1, and so on.

```
#Example usage of indexlist_ex = [1,2, 'Test']
print(list_ex[0])
```

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

When we use the index to access elements from the end of a list, it's called reverse indexing. In reverse indexing, the indexing of elements starts from the last element with the index number –1. The second last element has index '–2', and so on. These indexes used in reverse indexing are called negative indexes.

```
#Example usage of indexlist_ex = [1,2, 'Test']
print(list_ex[-1])
```

18. What is a dictionary in Python?

Python dictionary is one of the supported data types in Python. It is an unordered collection of elements. The elements in dictionaries are stored as key-value pairs. Dictionaries are indexed by keys. The data type is presented by `{}` .

```
#Example of Dictionarydictionary = {'key' : 'value'}
```

19. How to access values in a dictionary?

You could access the values in a dictionary by indexing using the key. Indexing is presented by `[]` .

```
#Accessing Dictionarydictionary = {'key' : 'value'}
print(dictionary['key'])
```

20. How do you get a list of all the keys in a dictionary?

In Dictionary, there is a keys() attribute we could use.

```
dictionary = {'key' : 'value', 'key1': : 'value1'}
print(dictionary.keys())
```

21. What is the difference between list and tuple?

The difference between list and tuple is that list is mutable while tuple is not. Tuple can be hashed for e.g as a key for dictionaries. The list is defined using `[]` and tuple defined using `()` .

```
#Example of list and tuple#List
list_ex = [1,2,'test']#List is mutable
list_ex[0] = 100#Tuple
tuple_ex = (1,2,'test)#Tuple is not mutable
tuple_ex[0] = 100 #It would error
```

22. In Python what are iterators?

In Python, iterators are used to iterate a group of elements, containers like the list or string. By iteration, it means that it could be looped by using a statement.

23. What does [::-1} do?

[::-1] is used to reverse the order of any iterable object.

```
#Reverse examplestring = 'this is a string'
print(string[::-1])
```

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

The Ternary operator is the operator that is used to show the conditional statements. This consists of true or false values with a statement that has to be evaluated for it.

```
#Ternary operators examplea = 1 #The true values
if a < 1:
   print('Less')
#If the previous condition haven't fulfilled
else:
   print('More')
```

25. How does break work?

The break statement allows loop termination when some condition is met and the control is transferred to the next statement.

```
#Break examplefor i in range(5):
   if i < 3:
      print(i)
   else:
      break
```

26. What is the purpose pass statement in python?

The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute.

```
#Pass examplefor i in range(10):
   if i%2 == 0:
      print(i)
   else:
      pass
```

27. What is a map function in Python?

The map() function is a function that takes a function as an argument and then applies that function to all the elements of an iterable, passed to it as another argument. It would return a map object so we need to transform it to a list object.

```
#map function exampledef number_exponential(num):
   return num**2number_list = [2,3,4,5]print(list(map(number_exponential,
number_list)))
```

28. What is a enumerate function in python?

The enumerate() method adds a counter to an iterable and returns it in a form of enumerate object. The object would consist of the counter and the iterable values.

```
#Enumerate exampleiter_example = ['test', 'test2', 'test3']
for idx, val in enumerate(iter_example):
   print(idx)
   print(val)
```

29. What is Dict and List comprehensions are?

They are syntax constructions to ease the creation of a Dictionary or List based on existing iterable. It is created by looping inside the Dictionary or List object.

```
#Dictionary comprehensiondict_comprehension = {key:val for key, val in emumerate('sample')}
print(dict_comprehension)#List comprehensionlist_comprehension = [i for i in range(5)]
print(list_comprehension)
```

30. What is slicing in Python?

Slicing is a mechanism to select a range of items from sequence types like list, tuple, strings, etc. This slicing is done by indexing method.

```
#Slicing examplelist_example = [1,2,3,4,'test','test2']
print(list_example[1:4])
```

31. What are the purposes of not in the operator?

Operators are special functions. They take one or more values and produce a corresponding result. `not` would return the inverse of the boolean value.

```
print(not 1 == 2)
```

32. What is the purpose of // in python?

It is a Floor Division operator, which is used for dividing two operands with the result showing only digits before the decimal point.

```
print(5//2)
```

33. How do you add a new value to a list object?

You can do it by using .append() attribute that list has. By passing any values to the .append() attribute, the new value would be placed at the end of the list sequence.

```
list_example = [1,2,3,4,5]
list_example.append(6)
print(list_example)
```

34. What is a shallow copy?

Shallow copy is used when a new instance type gets created and it keeps the values that are copied in the new instance. Shallow copy is used to copy the reference pointers just like it copies the values. It means when we copying an object to another variable, it would be connected.

```
#Example of shallow copylist_example = [1,2,3,4,5]
another_list = list_example
another_list[0] = 100
print(list_example)
```

35. What is a deep copy?

Deep copy is used to store the values that are already copied. The deep copy doesn't copy the reference pointers to the objects. It makes the reference to an object and the new object that is pointed by some other object gets stored. Contrast with a shallow copy, The changes made in the original copy won't affect any other copy that uses the object. It means they are not connected.

```
#Example of Deep copylist_example = [1,2,3,4,5]#Iniating Deep copy with
.copy attribute
another_list = list_example.copy()
another_list[0] = 100
print(list_example)
```

36. How to create an empty class in Python?

An empty class is a class that does not have any code defined within its block. It can be created using the *pass* keyword. However, you can create objects of this class outside the class itself. In Python, the `pass`command does nothing when its executed. it's a null statement.

```
class sample:
    pass
test=sample()
test.name="test1"
print(test.name)
```

37. What is self-keyword in Python?

Self-keyword is used as the first parameter of a function inside a class that represents the instance of the class. The object or the instance of the class is automatically passed to the method that it belongs to and is received in the 'self-keyword.' Users can use another name for the first parameter of the function that catches the object of the class, but it is recommended to use 'self-keyword' as it is more of a Python convention.

38. Will the do-while loop work if you don't end it with a semicolon?

This is a **Trick question!** Python does not support an intrinsic do-while loop. Secondly, to terminate do-while loops is a necessity for languages like C++.

39. How will you convert a list into a string?

In this case, we could use a .join() attribute from the string object. Here we passed the list object to the attribute.

```
list_example = ['apple', 'grape', 'orange']
print(' '.join(list_example))
```

40. What is a membership operator?

It is an operator that can confirm if a value is a member in another object. The operators are 'in' and 'not in'

```
#Example membership operatorsprint('me' in 'membership')
print('mes' not in 'membership')
```

41. What is identity operators in Python?

Identity operators is an operator that tell us if two values have the same identity. The operators are 'is' and 'is not'.

```
#Example Identity operators
print(1 is '1')
print(2 is not '2')
```

42. How do you take input in Python?

For taking input from the user, we could use the function input(). This function would take input from the user and return the input into a string object.

```
test = input('input a number: ')
print(test)
```

43. What does the function zip() do?

It would return an iterator of tuples. It would form an n-pair of value from iterable passed on the function. The n is the number of iterable passed on the function.

```
#Zip function exampleprint(list(zip([1,2,3], ['apple', 'grape', 'orange'], ['x', 2, True])))for num, fruit, thing in zip([1,2,3], ['apple', 'grape', 'orange'], ['x', 2, True]):
    print(num)
    print(fruit)
    print(thing)
```

44. What is the difference if range() function takes one argument, two arguments, and three arguments?

When we pass only one argument, it takes it as the stop value. Here, the start value is 0, and the step value is +1. The iteration with a range would always stop 1 value before the stop value.

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

When we pass two arguments, the first one is the start value, and the second is the stop value.

```
for i in range(1,5):
   print(i)
```

Using three arguments, the first argument is the start value, the second is the stop value, and the third is the step value.

```
for i in range(1,10,2):
   print(i)
```

45. What is the best code you can write to swap two numbers?

You can perform the swab with a single line of code.

```
a = 1
b = 2#Swab number
a, b = b, a
```

46. How can you declare multiple assignments in one line of code?

There are two ways to do this. First is by separately declare the variable in the same line.

```
a, b, c = 1,2,3
```

Another way is by declaring the variable in the same line with only one value.

```
a=b=c=1
```

47. How to break out of the Infinite loop?

You can do it by pressing Ctrl+C to interrupt the looping process.

48. What is the `with`statement in Python?

The `with` statement in Python ensures that cleanup code is executed when working with unmanaged resources by encapsulating common preparation and cleanup tasks. It may be used to open a file, do something, and then automatically close the file at the end. It may be used to open a database connection, do some processing, then automatically close the connection to ensure resources are closed and available for others. with will cleanup the resources even if an exception is thrown.

```
#Example of with statementwith open('database.txt') as data:
    print(data)
```

49. In a try-except block, when is except block executed?

The try-except block is commonly used when we want something to execute when errors were raised. The except block is executed when the code in the try block has encountered an error.

```
a = (1,2,3)
try:
   a[0] = 2
except:
   print('There is an error')
```

50. Where will you use `while` rather than `for`?

For simple repetitive looping and when we don't need to iterate through a list of items- like database records and characters in a string.

51. What is a Python module?

Modules are independent Python scripts with the .py extension that can be reused in other Python codes or scripts using the import statement. A module can consist of functions, classes, and variables, or some runnable code. Modules not only help in keeping Python codes organized but also in making codes less complex and more efficient.

```
import #name of the module
```

52. What is PYTHONPATH?

It is an environment variable that is used when a module is imported. Whenever a module is imported, PYTHONPATH is also looked up to check for the existence of the imported modules in various directories. The interpreter uses it to determine which module to load.

53. Name the example of file-processing modes with Python?

We have the following modes:

- Read-only mode('r'): Open a file for reading. It is the default mode.
- Write-only mode('w'): Open a file for writing. If the file contains data, data would be lost. Another new file is created.
- Read-Write mode('rw'): Open a file for reading, write mode. It means updating mode.
- Append mode('a'): Open for writing, append to the end of the file, if the file exists.

54. What is pickling and unpickling?

Pickle module accepts any Python object and converts it into a string representation and dumps it into a file by using a dump function, this process is called pickling. While the process of retrieving original Python objects from the stored string representation is called unpickling.

```
import picklea = 1#Pickling process
pickle.dump(a, open('file.sav', 'wb'))#Unpickling process
file = pickle.load(open('file.sav', 'rb'))
```

55. Is python NumPy array better than lists?

We use python NumPy array instead of a list because of the below three reasons:

1. Less Memory
2. Fast
3. Convenient

56. How do you calculate percentiles with NumPy?

Percentiles is the position of the ordered number in a certain percentile. We can calculate the percentile with NumPy using the following code.

```
import numpy as npa = np.array([i for i in range(100)])
p = np.percentile(a, 50) #Returns 50th percentile, e.g. medianprint(p)
```

57. How do you get the current working directory using Python?

Working with Python, you may need to read and write files from various directories. To find out which directory we're presently working under, we can use the getcwd() method from the os module.

```
import os
os.getcwd()
```

58. What do you see below? What would happen if we execute it?

```
a = '1'
b = '2'
c = '3's = a + `[` + b + `:` + c + `]`
print(s)
```

This is string concatenation. If even one of the variables isn't a string, this would raise a TypeError. What would happen is that we get an output of the string concatenation.

59. How would you randomize the contents of a list in-place?

We can use the help of function shuffle() from the module random.

```
from random import shuffle
list_example = [1,2,3,4,5,6,7,8]
shuffle(list_example)
```

60. What is a cast in Python?

Casting is when we convert a variable value from one type to another. In Python, it could be done with functions such as list(),int(), float() or str() . An example is when you convert a string into an integer object.

```
a = '1'
b = int(a)
```

61. Explain why are we getting an error here?

```
from numpy imprt stdevImportError Traceback (most recent call last)
<ipython-input-26-685c12521ed4> in <module>
----> 1 from numpy import stdev

ImportError: cannot import name 'stdev' from 'numpy'
```

In the above code, we try to import a non-exist function from the numpy module. That is why we getting an error.

62. How can you unsign or delete variables in Python?

We could use the del() function to remove or unsign a variable. This is considered a good practice to remove all the unnecessary variables when we are not using it.

```
a = 1
del a
```

63. What is a pandas in Python?

pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

64. What is the difference between append() and extend() methods?

Both append() and extend() methods are methods used to add elements at the end of a list.

- append(element): Adds the given element at the end of the list

- extend(another-list): Adds the elements of another list at the end of the list

65. How do you find the current version of Python?

We can find our Python current version by using sys.version.

```
import sys
sys.version
```

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

We use *args when we aren't sure how many arguments are going to be passed to a function, or if we want to pass a stored list or tuple of arguments to a function. **kwargs is used when we don't know how many keyword arguments will be passed to a function, or it can be used to pass the values of a dictionary as keyword arguments. The identifiers args and kwargs are optional, as you could change it to another name such as *example **another but it is better to just use the default name.

```
#Example of *args
def sample(*args):
    print(args)sample('time', 1, True)#Example of **kwargs
def sample(**kwargs):
    print(kwargs)sample(a = 'time', b = 1)
```

67. What is help() and dir() functions in Python?

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

```
import numpy
help(numpy.array)
```

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

```
import numpy
dir(numpy.array)
```

68. What is the meaning of a single- and a double-underscore before an object name?

Single Underscore — Names, in a class, with a leading underscore are simply to indicate to other programmers that the attribute or method is intended to be private. However, nothing special is done with the name itself.

Double Underscore (Name Mangling) — Any identifier of the form __spam (at least two leading underscores, at most one trailing underscore) is textually replaced with _classname__spam, where the class name is the current class name with a leading underscore(s) stripped. This mangling is done without regard to the syntactic position of the identifier, so it can be used to define class-private instance and class variables, methods, variables stored in globals, and even variables stored in instances. private to this class on instances of other classes.

69. What is the output of this below query?

```
ss = "Python Programming!"
print(ss[5])
```

The answer is 'n'

70. Write a program in Python to produce a Star triangle.

```
def star_triangle(r):
   for x in range(r):
      print(' '*(r-x-1)+'*'*(2*x+1))star_triangle(7)
```

```
      *
     ***
    *****
   *******
  *********
 ***********
*************
```

71. What is wrong with this following code?

```
counter = 0def increment():
   counter += 1increment()
```

Python doesn't have variable declarations, so it has to figure out the scope of variables itself. If there is an invitation to a variable inside a function, that variable is considered local. The counter variable above is a global variable and thus, the line of code above would raise an error.

72. How to split a string into a list?

We can use the .split() attribute from the string. It takes the separator as an argument and return list consisting of splitting results of the string based on the separator.

```
text = 'hello again world !'
text.split(' ')
```

73. Write a program in Python to check if a sequence you input is a Palindrome.

```
a=input("enter the sequence: ")
b=a[::-1]if a==b:
   print("palindrome")
else:
   print("Not a Palindrome")
```

74. What is a generator?

Python generator produces a sequence of values to iterate on, often by using a function. We define a function using `yield`that used to yield a value one by one, and then use a for loop looping to iterate on it.

```
def squares(n):
    i=1
    while(i<=n):
        yield i**2
        i+=1
for i in squares(7):
    print(i)
```

75. Write a program in Python to check if a number is prime.

```
a=int(input("enter a number"))
if a>1:
    for x in range(2,a):
        if(a%x)==0:
            print("not prime")
            break
    else:
        print("Prime")
else:
    print("not prime")
```

76. What is the purpose of the single underscore (‘_’) variable in Python?

It is to hold the result of the last executed expression(/statement) in an interactive interpreter session. This precedent was set by the standard CPython interpreter, and other interpreters have followed this.

77. What are the types of inheritance in Python?

Python supports different types of inheritance, they are:

- Single Inheritance
- Multi-level Inheritance
- Hierarchical Inheritance

- Multiple Inheritance

78. What is tuple unpacking?

Tuple unpacking is a process of unpacking the values in the tuple and input it into a few different variables.

```
tup = (1,2,3)#Tuple unpacking process
a,b,c = tup
```

79. When you exit Python, is all memory deallocated?

Exiting Python deallocates everything except:

1. Modules with circular references

2. Objects referenced from global namespaces

3. Parts of memory reserved by the C library

80. If a function does not have a return statement, is it valid?

A function that doesn't return anything returns a None object. Not necessarily does the return keyword mark the end of a function; it merely ends it when present in the function. Normally, a block of code marks a function, and where it ends, the function body ends.