1.What are local variables and global variables in Python?
---L.V:::Local variables in Python are those which are initialized inside a function and belong only to that particular function. It cannot be accessed anywhere outside the function. Let's see how to create a local variable.
---G.V:::These are those which are defined outside any function and which are accessible throughout the program, i.e., inside and outside of every function. Let's see how to create a Python global variable.

2.When to use a tuple vs list vs dictionary in Python?
tuple:::
A Tuple is a collection of Python objects separated by commas. In some ways, a tuple is similar to a list in terms of indexing, nested objects, and repetition but a tuple is immutable, unlike lists that are mutable.
ex:::
(1, 2, 3, 4, 5)

list:::
Python Lists are just like dynamic-sized arrays, declared in other languages (vector in C++ and ArrayList in Java). Lists need not be homogeneous always which makes it the most powerful tool in Python.
ex:::
Var = ["Geeks", "for", "Geeks"]

dictionary:::
Dictionary holds key:value pair. Key-value is provided in the dictionary to make it more optimized.
Example: {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}

---Use a tuple to store a sequence of items that will not change.

Use a list to store a sequence of items that may change.

Use a dict when you want to associate pairs of two items.

3.Explain some benefits of Python
===It is easy to learn and use, with a simple and expressive syntax.
It is open-source and has a large and active community of developers and learners.
It is versatile and supports multiple programming paradigms, such as object-oriented, functional, and procedural.
It has extensive libraries and frameworks for various applications, such as data analytics, web development, machine learning, and more.
It is portable and works on any computer and operating system.
It performs automatic memory management and has high productivity.

CONS:
Speed is slow.beacause of interpreted language.
it is a weak language for mobile development. It is very rarely used for mobile development.
Memory Consumption is high.

Runtime Errors
Difficulty in Using Other Languages

4.What is Lambda Functions in Python?
---A lambda function is a small anonymous function.
A lambda function can take any number of arguments, but can only have one expression.

===The power of lambda is better shown when you use them as an anonymous function inside another function.
ex::;
```
def myfunc(n):
  return lambda a : a * n
```

5.How do I modify a string in python?
Upper ()
Lower ()
strip()
replace()
split()

6.What is a Negative Index in Python?
Use negative indexes to start the slice from the end of the string:
---Using index() + len()

7.What is a dynamically typed language?
Dynamically typed language: These are the languages that do not require any pre-defined data type for any variable as it is interpreted at runtime by the machine itself. In these languages, interpreters assign the data type to a variable at runtime depending on its value. We don't even need to specify the type of variable that a function is returning or accepting in these languages. JavaScript, Python, Ruby, Perl, etc are examples of dynamically typed languages.

8.What is an Interpreted language?
An interpreted language is a programming language that uses an interpreter to execute the source code line by line. An interpreter is a software that converts the high-level code into machine language, but does not create an executable file. Some examples of interpreted languages are PHP, RUBY, Python, JavaScript, Perl, and Basic.

9.What is Scope in Python?
===A variable is only available from inside the region it is created. This is called scope.

10.What are lists and tuples? What is the key difference between the two?
List and Tuple in Python are the classes of Python Data Structures. The list is dynamic, whereas the tuple has static characteristics. This means that lists can be modified whereas tuples cannot be modified, the tuple is faster than the list because of static in nature. Lists are denoted by the square brackets but tuples are denoted as parenthesis.

11.What are the common built-in data types in Python?
---The following are the standard or built-in data types in Python:
Numeric
Sequence Type
Boolean
Set
Dictionary

12.What is pass in Python?
When the user does not know what code to write, So user simply places a pass at that line. Sometimes, the pass is used when the user doesn't want any code to execute. So users can simply place a pass where empty code is not allowed, like in loops, function definitions, class definitions, or in if statements. So using a pass statement user avoids this error.

13.What are modules and packages in Python?
---A Python module is a file containing Python definitions and statements. A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.

We can import the functions, and classes defined in a module to another module using the import statement in some other Python source file.

---A package is considered a collection of tools that allows the programmers to initiate the code. A Python package acts as a user-variable interface for any source code. This feature allows a Python package to work at a defined time for any functional script in the runtime.

14. What are global, protected and private attributes in Python?
---The members of a class that are declared public are easily accessible from any part of the program. All data members and member functions of a class are public by default.

The members of a class that are declared protected are only accessible to a class derived from it. Data members of a class are declared protected by adding a single underscore '_' symbol before the data member of that class.

The members of a class that are declared private are accessible within the class only, private access modifier is the most secure access modifier. Data members of a class are declared private by adding a double underscore '__' symbol before the data member of that class.

15. What is the use of self in Python?
---self represents the instance of the class. By using the "self" we can access the attributes and methods of the class in python. It binds the attributes with the given arguments.

---Self is always pointing to Current Object.

16. What is __init__?
---In Python, __init__ is a special method known as the constructor. It is automatically called when a new instance (object) of a class is created. The __init__ method allows you to initialize the attributes (variables) of an object.

17. What is break, continue and pass in Python?
--The break statement in Python is used to terminate the loop or statement in which it is present. After that, the control will pass to the statements that are present after the break statement, if available. If the break statement is present in the nested loop, then it terminates only those loops which contain the break statement.

---As the name suggests pass statement simply does nothing. The pass statement in Python is used when a statement is required syntactically but you do not want any command or code to execute. It is like a null operation, as nothing will happen if it is executed. Pass statements can also be used for writing empty loops. Pass is also used for empty control statements, functions, and classes.

18. What are unit tests in Python?
----Unit Testing is the first level of software testing where the smallest testable parts of a software are tested. This is used to validate that each unit of the software performs as designed.

19. What is docstring in Python?
---It's specified in source code that is used, like a comment, to document a specific segment of code. Unlike conventional source code comments, the docstring should describe what the function does, not how.

Declaring Docstrings: The docstrings are declared using "'triple single quotes"' or ""triple double quotes"" just below the class, method or function declaration. All functions should have a docstring.

20. What is slicing in Python?
---In Python, list slicing is a common practice and it is the most used technique for programmers to solve efficient problems. Consider a Python list, in order to access a range of elements in a list, you need to slice a list. One way to do this is to use the simple slicing operator i.e. colon(:). With this operator, one can specify where to start the slicing, where to end, and specify the step. List slicing returns a new list from the existing list.

Lst[ Initial : End : IndexJump ]

21. What is the difference between Python Arrays and lists?
----In Python, lists and arrays are the data structures that are used to store multiple items. They both support the indexing of elements to access them, slicing, and

iterating over the elements. However, there are some differences between them. Arrays are more efficient than lists for some uses because they are stored in contiguous memory locations. This means that accessing an element in an array is faster than accessing an element in a list. However, insertion and deletion operations are more expensive in arrays because all the elements after the inserted or deleted element have to be shifted over by one position. In contrast, insertion and deletion operations are faster in lists because only the links between the elements have to be updated1.

22. How is memory managed in Python?
---In Python, memory is managed by the Python manager which determines where to put the application data in the memory. So, we must have the knowledge of Python memory manager to write efficient code and maintainable code.
--- Memory allocation can be defined as allocating a block of space in the computer memory to a program. In Python memory allocation and deallocation method is automatic as the Python developers created a garbage collector for Python so that the user does not have to do manual garbage collection.

G.C::
Garbage collection is a process in which the interpreter frees up the memory when not in use to make it available for other objects.

23. What are Python namespaces? Why are they used?
----In Python, a namespace is a collection of currently defined symbolic names along with information about the object that each name references. You can think of a namespace as a dictionary in which the keys are the object names and the values are the objects themselves. Each key-value pair maps a name to its corresponding object. There are four types of namespaces in Python: Built-In, Global, Enclosing and Local.

Namespaces are used to avoid naming conflicts. They ensure that names are unique and can be used without any conflict. Namespaces also help in modular programming by organizing code into different namespaces. This makes it easier to maintain and debug code1
24. What is Scope Resolution in Python?
----Scope resolution LEGB rule In Python
In Python, the LEGB rule is used to decide the order in which the namespaces are to be searched for scope resolution. The scopes are listed below in terms of hierarchy(highest to lowest/narrowest to broadest):

Local(L): Defined inside function/class
Enclosed(E): Defined inside enclosing functions(Nested function concept)
Global(G): Defined at the uppermost level
Built-in(B): Reserved names in Python builtin modules

25. What are Dict and List comprehensions?
---List comprehensions and dictionary comprehensions are powerful tools in Python that allow you to create new lists or dictionaries from existing iterables. They are concise and easy to read, and can be faster than traditional for-loops.

List comprehensions are used to create new lists based on existing lists. They are written inside square brackets and consist of an expression followed by a for clause, which specifies the input sequence and an optional predicate expression.

Dictionary comprehensions are similar to list comprehensions but create dictionaries instead of lists. They are written inside curly braces and consist of a key-value pair followed by a for clause, which specifies the input sequence and an optional predicate expression.
list:::
```
squares = [x**2 for x in [1, 2, 3, 4, 5]]
```

dict:::
```
keys = ['a', 'b', 'c']
values = [1, 2, 3]
dictionary = {k:v for k,v in zip(keys, values)}
```

26. What is lambda in Python? Why is it used?
----The power of lambda is better shown when you use them as an anonymous function inside another function.
---Say you have a function definition that takes one argument, and that argument will be multiplied with an unknown number:
ex:::
```
def myfunc(n):
  return lambda a : a * n
```

27. How do you copy an object in Python?
-------So, if I change values of the fields of the new object, the old object should not be affected by that.
---I would like to create a copy of an object. I want the new object to possess all properties of the old object (values of the fields). But I want to have independent objects. So, if I change values of the fields of the new object, the old object should not be affected by that.

28. What is the difference between xrange and range in Python?
----The range() and xrange() are two functions that could be used to iterate a certain number of times in for loops in Python. In Python 3, there is no xrange, but the range function behaves like xrange in Python 2. If you want to write code that will run on both Python 2 and Python 3, you should use range().

29. What is PYTHONPATH in Python?
----In Python, PYTHONPATH is an environment variable that specifies a list of directories to search for Python modules when importing them. When you import a module in Python, Python looks for the module in the directories specified in sys.path, which is a list of directories that includes the current working directory and directories specified in PYTHONPATH.

30. What is the difference between .py and .pyc files?
---.pyc contain the compiled bytecode of Python source files. The Python interpreter loads .pyc files before .py files, so if they're present, it can save some time by not having to re-compile the Python source code. You can get rid of them if you want, but they don't cause problems, they're not big, and they may save some time when running programs.

31. How Python is interpreted?
---As an interpreted language, Python code is executed by an interpreter. The interpreter reads the Python source code and executes it one line at a time. The interpreter also performs error checking during the execution of the code. If an error is found in a particular line, the interpreter will halt the program and report the error.

32. How are arguments passed by value or by reference in python?
---In Python, all parameters (arguments) are passed by reference. It means if you change what a parameter refers to within a function, the change also reflects back in the calling function.

---Python passes arguments by object1. If you pass immutable arguments like integers, strings or tuples to a function, the passing acts like Call-by-value. If you pass mutable arguments, all parameters (arguments) in the Python language are passed by reference23. It means if you change what a parameter refers to within a function, the change also reflects back in the calling function3. When you pass function arguments by reference, those arguments are only references to existing values. In contrast, when you pass arguments by value, those arguments become independent copies of the original values4.

33. Explain how to delete a file in Python?
---To delete a file, you must import the OS module, and run its os.remove() function:
ex::
```
import os
os.remove("demofile.txt")
```

34. Explain split() and join() functions in Python?
--- In Python, we can use the function split() to split a string and join() to join a string. the split() method in Python split a string into a list of strings after breaking the given string by the specified separator. Python String join() method is a string method and returns a string in which the elements of the sequence have been joined by the str separator.

EX::
```
# Python code
# to split and join given string

# input string
s = 'Geeks for Geeks'
# print the string after split method
print(s.split(" "))
```

```
# print the string after join method
print("-".join(s.split()))
```

35. What are negative indexes and why are they used?
---Negative indexes are used to indicate the position of an element within a list or string in reverse order12. For instance, if you have to write the list in the opposite order, you can use the negative index2. Negative indexes are also used to remove any new-line spaces from the string and allow the string to except the last character that is given as S1.

---Use negative indexes to start the slice from the end of the string:
```
EX::b = "Hello, World!"
print(b[-5:-2])
```

36. What are the key features of Python?
---1. Free and Open Source
2. Easy to code
3. Easy to Read
Interpreted Language
4. Object-Oriented Language
5. High-Level Language
Large Community Support
Python is an Integrated language
. Python is a Portable language

37. What type of language is python? Programming or scripting?
---Scripting language
---Python is both a programming and a scripting language. It is a general-purpose, high-level, and interpreted language that includes the object-oriented programming approach. Python is also a dynamic type of discipline and has strong typing. Python is an interpreted language, which means that it uses an interpreter to translate and run its code. Hence, Python is also a scripting language 1

38. What is pep 8?
---PEP 8 is a document that provides guidelines and best practices on how to write Python code1234. It was written in 2001 by Guido van Rossum, Barry Warsaw, and Nick Coghlan12. The primary focus of PEP 8 is to improve the readability and consistency of Python code12. PEP 8 is one of the most readable and eye-pleasing coding styles and is used by many projects all across the world3.

39. What are Keywords in Python?
----Keywords in Python are reserved words that can not be used as a variable name, function name, or any other identifier.

----In Python, keywords are the reserved words that cannot be used as variable names, function names or any other identifiers. They are used to define the syntax and structure of the Python language. Some of the keywords in Python are True, False, None, and, as, assert, break, class, continue, def, del, elif, else, except, finally, for,

from, global, if, import, in, is, lambda, nonlocal, not, or, pass, raise, return, try, and while1

40. What are the new features added in Python 3.9.0.0 version?
---1. Dictionary merge and update operators : Python 3.9 introduces merge(|) and update(|=) operators in the dict class. If you have two dictionaries a and b, you can now use these operators to do merge and update them.

2. removeprefix() and removesuffix() string methods : In Python's str class, the new update introduces new removeprefix() and removesuffix() methods.

3. New Parser : Python 3.9 uses a new parser which is a PEG-based parser. Previously, Python used LL(1). PEG is more flexible than LL(1) when it comes to building new features in the language. The official documentation says that this flexibility will be seen in Python 3.10 and later.

4. Type Hinting : Python dynamically specifies datatypes to a variable. For static allocation of data types, type hinting is used. This was introduced in Python 3.5. You can now use built-in collection types (list and dict ) as generic types. Earlier, you had to import the capital types (List or Dict) from typing.

41. Is python case sensitive?
---Python is a case-sensitive language12345. This means that uppercase and lowercase characters are distinguished during execution2. Identifiers, keywords, and names of classes must be written in lowercase letters4. However, uppercase letters can be used for constants, built-in functions, and class and module names4. Two terms in Python are treated differently if their case is different, even if the characters are the same2. This can result in an error if the case is not matched23. Case sensitivity in Python increases the number of identifiers or symbols you can use1

42. What is type conversion in Python?
---Python defines type conversion functions to directly convert one data type to another which is useful in day-to-day and competitive programming. This article is aimed at providing information about certain conversion functions.

There are two types of Type Conversion in Python:

Implicit Type Conversion
Explicit Type Conversion

43. Is indentation required in python?
---Indentation is a very important concept of Python because without properly indenting the Python code, you will end up seeing IndentationError and the code will not get compiled.

Python Indentation
Python indentation refers to adding white space before a statement to a particular block of code. In another word, all the statements with the same space to the right, belong to the same code block.

44. What are functions in Python?

---Python Functions is a block of statements that return the specific task. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can do the function calls to reuse code contained in it over and over again.

Some Benefits of Using Functions

Increase Code Readability
Increase Code Reusability

45. How do you write comments in python?

---Comments in Python are the lines in the code that are ignored by the interpreter during the execution of the program. Comments enhance the readability of the code and help the programmers to understand the code very carefully.

Single line Comments------#
Multiline Comments-------Multiline comments using multiple hashtags (#,#...)
Docstring Comments----"""....."""

46. What are the generators in python?

----A Generator in Python is a function that returns an iterator using the Yield keyword. In this article, we will discuss how the generator function works in Python.

Generator Function in Python

A generator function in Python is defined like a normal function, but whenever it needs to generate a value, it does so with the yield keyword rather than return. If the body of a def contains yield, the function automatically becomes a Python generator function.

Create a Generator in Python

In Python, we can create a generator function by simply using the def keyword and the yield keyword. The generator has the following syntax in Python:

```
def function_name():
    yield statement

# A generator function that yields 1 for first time,
# 2 second time and 3 third time
def simpleGeneratorFun():
        yield 1
        yield 2
        yield 3

# Driver code to check above generator function
for value in simpleGeneratorFun():
        print(value)
```

47. How to comment multiple lines in python?

----You can comment out multiple lines in Python using the hash character # or by turning the lines into a string. To use the hash character, place a # and a space before each line of code you want to comment out. Alternatively, you can turn the lines into a string by placing \"\"\" before and after the block of code. Here's an example:

```
# This is a comment
# that spans multiple lines

\"\"\"
This is another way to comment out
multiple lines of code in Python.
\"\"\"
```

48. What is a dictionary in Python?

---In Python, a dictionary is a collection of key-value pairs. It is used to store data values like a map, which unlike other data types which hold only a single value as an element. A dictionary holds key:value pair. Key-Value is provided in the dictionary to make it more optimized. The keys in a dictionary must be unique and immutable while the values can be of any type and can be duplicated. Dictionaries are written with curly brackets, and have keys and values

```
Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}
print(Dict)
```

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

---Yes, Python has a ternary conditional operator, also known as the conditional expression or the ternary operator. The syntax of the ternary operator in Python is:

```
value_if_true if condition else value_if_false

x = 5
result = "Even" if (x % 2 == 0) else "Odd"
print(result)
```

===You can use the ternary operator in Python to write simple if/else statements in a single line. It returns a true or false value by evaluating a boolean condition. It is shorter and more readable than simple if/else statements. You can also implement ternary operators with tuples, lists, dictionaries, or lambda functions

50. What does len() do?

---The LEN function is a text function in Excel that returns a string/ text length. The LEN function in Excel can count the number of characters in a text string and count letters, numbers, special characters, non-printable characters, and all spaces from an Excel cell.

51. How to add values to a python array?

---You can add elements to a Python array using the append(), extend(), and insert() methods. The append() method adds a single element to the end of the array. The

extend() method adds a list, array, or other iterable to the end of the array. The insert() method inserts an element before the given index of the array1.

```
import array

arr = array.array('i', [1, 2, 3])
arr.append(4)
arr.extend([5, 6])
arr.insert(0, 0)

print(arr) # Output: array('i', [0, 1, 2, 3, 4, 5, 6])
```

52. How to remove values to a python array?

To remove values from an array in Python, you can use one of the following methods123:

Use the del statement to delete one or more items from an array.
Use the pop() method to remove an element from the array.
Use the remove() method to remove an element from the array by passing the value of the element you want to remove.

53. What is the difference between deep and shallow copy?

---The difference between shallow and deep copy is:

A shallow copy of a collection is a copy of the collection structure, not the elements. With a shallow copy, two collections now share the individual elements.
A deep copy creates a new object that is entirely independent of the original object. Any changes made to the deep copy will not affect the original object and vice versa.
A shallow copy can be useful for creating references to existing objects without consuming more memory.

54. How is Multithreading achieved in Python?

---Multithreading in Python can be achieved using the threading module. The threading module provides a very simple and intuitive API for spawning multiple threads in a program. It is similar to the _thread module but has the following differences:

Threads can be created using the threading.Thread class.
Once a thread is created, it can be started by calling its start() method.
The thread will run in the background until the application exits or until the thread is explicitly stopped.

55. What are Python libraries? Name a few of them.

---Python libraries are a collection of precompiled codes that can be used later on in a program for some specific well-defined operations. Other than pre-compiled codes, a library may contain documentation, configuration data, message templates, classes, and values, etc. A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer. As we don't need to write the same code again and again for different programs. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc

ex:::
TensorFlow
Matplotlib
Pandas
Numpy
SciPy

56. What is split used for?
---The split() method is used to split a string into an array of substrings, and returns the new array. The split() method does not change the original string. The separator specifies the character(s) to use for separating the string. If the separator is not specified or is None, any whitespace string is a separator and empty strings are removed from the result. Here's an example in Python:

```
txt = "apple#banana#cherry#orange"

x = txt.split("#")

print(x)
```

57. How to import modules in python?
--- Python modules can get access to code from another module by importing the file/function using import. The import statement is the most common way of invoking the import machinery, but it is not the only way.

---import module_name

When the import is used, it searches for the module initially in the local scope by calling __import__() function. The value returned by the function is then reflected in the output of the initial code.

ex:

```
import math
pie = math.pi
print("The value of pi is : ",pie)
```

58. What are the applications of Python?
---Python is a general-purpose language that can be used in a range of applications1. Some of the major applications of Python include23:
Data Science: Analytics and Visualisation
Machine Learning
Web development: Website and Web Application
Financial Analysis
Desktop Applications

59. What are the advantages of Python?
---Some of the advantages of python are12345:

It is a powerful, flexible, and easy-to-use language that has English-like syntax and supports multiple programming paradigms.
It is free and open source, with a very active and supportive community.
It is an interpreted language that performs automatic memory management and executes code line by line.
It is dynamically typed, which means that variables do not need to be declared before use.
It has vast libraries that provide a wide range of functionalities and features

60. What is the difference between remove() function and del statement?
---The basic difference between "del" and "remove()" statement in Python is that "del" removes a name from a namespace, an item from a dictionary, or an item from a list, while "remove()" is a member function of the 'list' class that finds a specific entry in the list and removes it123. "remove()" removes the first matching value or object, not a specific indexing, while "del" removes the item at a specific index23. "del" is a statement, while "remove()" is a method123.

61. What is swapcase() function in the Python?
===Make the lower case letters upper case and the upper case letters lower case:

===The swapcase() method returns a string whereall the upper case letters are lower case and vice versa.

62. How to remove leading whitespaces from a string in the Python?
===Use the replaceAll() method of the String class with two arguments, which is replaceAll("\s", ""); where \s is a single space in unicode.
Use the String class trim method. It will remove all leading and trailing whitespace.
Use str.strip() to remove leading and ending spaces. If you want to remove all space characters, use str.replace(). Note that this only removes the “normal” ASCII space character ' ' U+0020 but not any other whitespace.

63. Why do we use join() function in Python?
---The Python join() function is used to join all the elements from an iterable and create a string12345. It returns a new string which is the concatenation of the other strings in the iterable specified1. The join() method provides a flexible way to create strings from iterable objects3. It joins each element of an iterable (such as list, string, and tuple) by a string separator (the string on which the join() method is called) and returns the concatenated string3. A string must be specified as the separator

64. Give an example of shuffle() method?
===Shuffling refers to the rearranging of elements in a random order which means no specific sequence is considered while arranging the elements.

In this tutorial, we will learn how we can shuffle the elements of a list using Python.

The different approaches that we will use to shuffle the elements are as follows-

Using Fisher-Yates shuffle algorithm
Using shuffle()

Using sample()
Random selection of elements and then appending them in a list

65. What is tuple in Python?
----Tuples are used to store multiple items in a single variable.

--Tuples allow duplicate values:

---A tuple is a collection which is ordered and unchangeable.

66. What are the different types of operators in Python?
---Types of Operators in Python
Arithmetic Operators
Comparison Operators
Logical Operators
Bitwise Operators
Assignment Operators
Identity Operators and Membership Operators

67. What is the namespace in Python?
---A namespace is a collection of currently defined symbolic names along with information about the object that each name references. You can think of a namespace as a dictionary in which the keys are the object names and the values are the objects themselves. Each key-value pair maps a name to its corresponding object.

68. What is Pass in Python?
---The Python pass statement is a null statement. But the difference between pass and comment is that comment is ignored by the interpreter whereas pass is not ignored.

69. Which programming language is a good choice between Java and Python?
===Both Java and Python are popular programming languages. Java is faster than Python because it is compiled, while Python is interpreted. Java is built to favor web development and programming, instead of data science. Python has been making amazing progress in the last few years and is widely searched on the internet1.

Java is the faster language when it comes to sheer speed 2. According to Stack Overflow, Java is the sixth most commonly used programming language
70. What are the differences between Python 2.x and Python 3.x?
===Python 3 syntax is simpler and easily understandable whereas Python 2 syntax is comparatively difficult to understand.
Python 3 default storing of strings is Unicode whereas Python 2 stores need to define Unicode string value with “u.”
Python 3 value of variables never changes whereas in Python 2 value of the global variable will be changed while using it inside for-loop.
Python 3 exceptions should be enclosed in parenthesis while Python 2 exceptions should be enclosed in notations.

Python 3 rules of ordering comparisons are simplified whereas Python 2 rules of ordering comparison are complex.
Python 3 offers Range() function to perform iterations whereas, In Python 2, the xrange() is used for iterations.

71. How Python does Compile-time and Run-time code checking?
===At compile time, Python only checks that the syntax is legal. It does no type checking at all. At run time, Python can type check things by either explicit checks using issubclass and isinstance, or by attempting the operation and seeing if they fail.

72. What is the usage of enumerate () function in Python?
---Often, when dealing with iterators, we also get need to keep a count of iterations. Python eases the programmers' task by providing a built-in function enumerate() for this task. Enumerate() method adds a counter to an iterable and returns it in a form of enumerating object. This enumerated object can then be used directly for loops or converted into a list of tuples using the list() function.

73. What is the purpose of "is", "not" and "in" operators?
---is: returns the true value when both the operands are true (Example: "x" is 'x') not: returns the inverse of the boolean value based upon the operands (example:"1" returns "0" and vice-versa. In: helps to check if the element is present in a given Sequence or not.

74. What Does the 'is' Operator Do?
--The is operator is used to check if the run-time type of an object is compatible with the given type or not. It returns true if the given object is of the same type otherwise, return false. It also returns false for null objects.

75. How Will You Check If All the Characters in a String Are Alphanumeric?
==Use the isalnum() method. This method returns True if all the characters are alphanumeric, meaning alphabet letter (a-z) and numbers (0-9)14.

76. How Would You Remove All Leading Whitespace in a String?
===Use the replaceAll() method of the String class with two arguments, which is replaceAll("\s", ""); where \s is a single space in unicode.
Use the String class trim method. It will remove all leading and trailing whitespace.
Use str.strip() to remove leading and ending spaces. If you want to remove all space characters, use str.replace(). Note that this only removes the "normal" ASCII space character ' ' U+0020 but not any other whitespace.

77. How Would You Replace All Occurrences of a Substring with a New String?
---To replace all occurrences of a substring with a new string, you can use the replace() method.

78. Differentiate Between append() and extend().
===Both append and extend are useful methods for adding elements to a list. The main difference is that append adds an element to the end of a list, while extend adds elements from another list (or any iterable) to the end of the list. In terms of performance, append is faster than extend.

79. How Do You Use Print() Without the Newline?
====In Python 3, you can use the sep= and end= parameters of the print function:

To not add a newline to the end of the string:

print('.', end='')
To not add a space between all the function arguments you want to print:

print('a', 'b', 'c', sep='')

80. What is a map() function in Python?
===map() function returns a map object(which is an iterator) of the results after applying the given function to each item of a given iterable (list, tuple etc.) Syntax :

map(fun, iter)

---map() is a function which the data can be converted into the required type.it takes the many inputs in the needed type.
81. What is the difference between / and // operator in Python?
----The difference between the / and the // operators in Python is that the / operator performs classic division, while the // operator performs floor division12. Classic division returns a floating-point number that is the exact quotient of the operands, while floor division returns an integer that is the largest whole number less than or equal to the quotient2. For example, 5/2 is 2.5, but 5//2 is 22.

---/:::gives the data in the form of floating point.
//::gives the data in the form of integer.

82. What are Python Decorators?
---In general, a decorator is:

A function that takes another function (original function) as an argument and returns another function (or closure)
The closure typically accepts any combination of positional and keyword-only arguments.

The closure function calls the original function using the arguments passed to the closure and returns the result of the function.

83. How does a function return values?
---The void keyword, used in the previous examples, indicates that the function should not return a value. If you want the function to return a value, you can use a data type (such as int, string, etc.) instead of void, and use the return keyword inside the function:

84. Define package in Python?

--A Python package is a directory that contains zero or more Python modules. A Python package can contain sub-packages, which are also directories containing modules. Each package always contains a special file named __init__.py.

85. What is polymorphism in Python?
---What is Polymorphism: The word polymorphism means having many forms. In programming, polymorphism means the same function name (but different signatures) being used for different types. The key difference is the data types and number of arguments used in function.

86. Define encapsulation in Python?
---Encapsulation is one of the fundamental concepts in object-oriented programming (OOP). It describes the idea of wrapping data and the methods that work on data within one unit. This puts restrictions on accessing variables and methods directly and can prevent the accidental modification of data. To prevent accidental change, an object's variable can only be changed by an object's method. Those types of variables are known as private variables.

A class is an example of encapsulation as it encapsulates all the data that is member functions, variables, etc. The goal of information hiding is to ensure that an object's state is always valid by controlling access to attributes that are hidden from the outside world.

87. Does python support multiple inheritance?
---Yes, Python supports multiple inheritance. Like C++, a class can be derived from more than one base classes in Python. This is called Multiple Inheritance.

In multiple inheritance, the features of all the base classes are inherited into the derived class

88. How do you do data abstraction in Python?
--Data Abstraction in Python can be achieved through creating abstract classes and inheriting them later12. An abstract class is a class that consists of one or more abstract methods, which do not contain their implementation2. Abstract classes can be inherited by the subclass, and the abstract method gets its definition in the subclass2. Abstraction in Python can also be achieved using classes and header files3

89. Does python make use of access specifiers?
---Python doesn't have access specifiers like C++ or Java. However, it uses a convention of prefixing the name of the variable/method with a single or double underscore to emulate the behavior of protected and private access specifiers1.

The single underscore prefix is used to indicate that the variable/method is intended for internal use only and should not be accessed from outside the class. The double underscore prefix invokes name mangling which makes it difficult to access the variable/method from outside the class2.

90. How to create an empty class in Python?

---Use the pass command after the definition of the class object.
Use type to create a new class on the fly and then instantiate it.
Define an empty class with no attributes or methods35.

91. What does an object() do?
---The object () function returns an empty object. You cannot add new properties or methods to this object. This object is the base for all classes, it holds the built-in properties and methods which are default for all classes.

92.Is python numpy better than lists?
===Yes, the NumPy array is better than lists.

NumPy is the fundamental package for scientific computing in Python.

Advantages of using Numpy Arrays Over Python Lists:

Consume less memory.
Fast as compared to the python list.
Convenient to use.
NumPy has optimized functions such as linear algebra operations built-in.

===Python NumPy arrays are more compact than lists, which means they take up less memory space. Reading and writing items is faster with NumPy than with lists. Using NumPy is more convenient than using the standard list because it provides fast and efficient operations on arrays of homogeneous data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences. NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Some important points about NumPy arrays are:

We can create an N-dimensional array in python using numpy.array().
The array is by default Homogeneous, which means data inside an array must be of the same Datatype.
Element-wise operation is possible.
Numpy array has various functions, methods, and variables, to ease our task of matrix computation.
Elements of an array are stored contiguously in memory.
On the other hand, Python lists are collections that are ordered and changeable. In Python, lists are written with square brackets. Some important points about Python Lists are:

The list can be homogeneous or heterogeneous.
Element-wise operation is not possible on the list.
Python list is by default 1-dimensional. But we can create an N-Dimensional list. But then too it will be 1 D list storing another 1D list.
Elements of a list need not be contiguous in memory.
Here's a comparison between Numpy array and Python List

93. How do you access parent members in the child class?
---To access the parent member in the child class, you need to call the constructor of the parent class inside the constructor of the child class. This can be done with the

help of the super() method. Another way to do this is to use inheritance, which allows the child class to inherit all of the parent class' methods and properties. The child class can then use these inherited methods and properties to access any data within the parent class.

94. Is it possible to call parent class without its instance creation?
---Yes, it is possible to call a parent class without its instance creation. Here are some ways to do it:

If it is a static method.
By inheriting from that class.
From derived classes using base keyword.

95. Differentiate between new and override modifiers.
----A method marked with "override" replaces a method from the base class. No matter what the variable type,the actual object type's method will be called. A method marked with "new" doesn't replace the base method. It's a new method. Casting the object to the base class lets you call the base method.
---You can specify how the methods interact by using the new and override keywords. The override modifier extends the base class virtual method, and the new modifier hides an accessible base class method. The difference is illustrated in the examples in this topic1.

96. Why is finalize used?
---Why finalize () method is used? finalize () method releases system resources before the garbage collector runs for a specific object. JVM allows finalize () to be invoked only once per object.
---The finalize() method is used for freeing up the unmanaged resources and cleaning up before the garbage collection method is invoked. This helps in performing memory management tasks

97. What is init method in python?
---In Python, __init__ is a special method known as the constructor. It is automatically called when a new instance (object) of a class is created. The __init__ method allows you to initialize the attributes (variables) of an object.

98. How will you check if a class is a child of another class?
---Python issubclass () is built-in function used to check if a class is a subclass of another class or not. This function returns True if the given class is the subclass of given class else it returns False.

99. Explain Normalization
---Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

===Normalization is the process of organizing the data in the database.
Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies.
Normalization divides the larger table into smaller and links them using relationships. The normal form is used to reduce redundancy from the database table.

100. How do you copy an object in Python?
---To get a fully independent copy of an object you can use the copy.deepcopy() function.

101. Differentiate between deep and shallow copy.
---Deep copy and shallow copy are two ways of copying objects in programming.Shallow copy creates a copy of the object structure, but maintains references to the original objects.Deep copy creates a copy of the object structure and clones the objects so that they bear no relation to the original.For example, if you have an array of pointers to structs or objects, a shallow copy will copy the array and maintain references to the original objects, while a deep copy will copy the objects too.

====By "shallow copying" it means the content of the dictionary is not copied by value, but just creating a new reference.

====In contrast, a deep copy will copy all contents by value.

Syntax: copy.deepcopy(x)

Syntax: copy.copy(x)

102. How are classes created in Python?
---Classes are created by keyword class. Attributes are the variables that belong to a class. Attributes are always public and can be accessed using the dot (.) operator.
=== class ClassName:

103. How Python module is imported?
---from a module, by using the from keyword.
ex:;
from mymodule import person1

print (person1["age"])

104. In Python,how do you abstract data?
---An abstract data type is an abstraction of a data structure that provides only the interface to which the data structure must adhere. The interface does not give any specific details about something should be implemented or in what programming language.