# Python Interview Questions

## 1. How will you improve the performance of a program in Python?

There are many ways to improve the performance of a Python program. Some of these are as follows:

i. Data Structure: We have to select the right data structure for our purpose in a Python program.
ii. Standard Library: Wherever possible, we should use methods from standard library. Methods implemented in standard library have much better performance than user implementation.
iii. Abstraction: At times, a lot of abstraction and indirection can cause slow performance of a program. We should remove the redundant abstraction in code.
iv. Algorithm: Use of right algorithm can make a big difference in a program. We have to find and select the suitable algorithm to solve our problem with high performance.

## 2. What are the benefits of using Python?

Python is strong that even Google uses it. Some of the benefits of using Python are as follows:

i. Efficient: Python is very efficient in memory management. For a large data set like Big Data, it is much easier to program in Python.
ii. Faster: Though Python code is interpreted, still Python has very fast performance.
iii. Wide usage: Python is widely used among different organizations for different projects. Due to this wide usage, there are thousands of add-ons available for use with Python.
iv. Easy to learn: Python is quite easy to learn. This is the biggest benefit of using Python. Complex tasks can be very easily implemented in Python.

## 3. How will you specify source code encoding in a Python source file?

By default, every source code file in Python is in UTF-8 encoding. But we can also specify our own encoding for source files. This can be done by adding following line after #! line in the source file.

```
# -*- coding: encoding -*-
```

In the above line we can replace encoding with the encoding that we want to use.

## 4. What is the use of PEP 8 in Python?

PEP 8 is a style guide for Python code. This document provides the coding conventions for writing code in Python. Coding conventions are about indentation, formatting, tabs, maximum line length, imports organization, line spacing etc. We use PEP 8 to bring consistency in our code. We consistency it is easier for other developers to read the code.

## 5. What is Pickling in Python?

Pickling is a process by which a Python object hierarchy can be converted into a byte stream. The reverse operation of Pickling is Un pickling.

Python has a module named pickle.

This module has the implementation of a powerful algorithm for serialization and de-serialization of Python object structure.

Some people also call Pickling as Serialization or Marshalling. With Serialization we can transfer Python objects over the network.

It is also used in persisting the state of a Python object. We can write it to a file or a database.

## 6. How does memory management work in Python?

There is a private heap space in Python that contains all the Python objects and data structures. In CPython there is a memory manager responsible for managing the heap space.

There are different components in Python memory manager that handle segmentation, sharing, caching, memory pre-allocation etc.

Python memory manager also takes care of garbage collection by using Reference counting algorithm.

## 7. How will you perform Static Analysis on a Python Script?

We can use Static Analysis tool called PyChecker for this purpose. PyChecker can detect errors in Python code. PyChecker also gives warnings for any style issues.

Some other tools to find bugs in Python code are pylint and pyflakes.

## 8. What is the difference between a Tuple and List in Python?

In Python, Tuple and List are built-in data structures. Some of the differences between Tuple and List are as follows:

I. Syntax: A Tuple is enclosed in parentheses:

  E.g. myTuple = (10, 20, “apple”); A List is enclosed in brackets

  E.g. myList = [10, 20, 30];

II. Mutable: Tuple is an immutable data structure. Whereas, a List is a mutable data structure.

III. Size: A Tuple takes much lesser space than a List in Python.

IV. Performance: Tuple is faster than a List in Python. So it gives us good performance.

V. Use case: Since Tuple is immutable, we can use it in cases like Dictionary creation. Whereas, a List is preferred in the use case where data can alter.

## 9. What is a Python Decorator?

A Python Decorator is a mechanism to wrap a Python function and modify its behavior by adding more functionality to it. We can use @ symbol to call a Python Decorator function.

## 10. How are arguments passed in a Python method? By value or by reference?

Every argument in a Python method is an Object. All the variables in Python have reference to an Object. Therefore, arguments in Python method are passed by Reference.

Since some of the objects passed as reference are mutable, we can change those objects in a method. But for an Immutable object like String, any change done within a method is not reflected outside.

## 11. What is the difference between List and Dictionary data types in Python?

Main differences between List and Dictionary data types in Python are as follows:

I. Syntax: In a List we store objects in a sequence. In a Dictionary we store objects in key-value pairs.

II. Reference: In List we access objects by index number. It starts from 0 index. In a Dictionary we access objects by key specified at the time of Dictionary creation.

III. Ordering: In a List objects are stored in an ordered sequence. In a Dictionary objects are not stored in an ordered sequence.

IV. Hashing: In a Dictionary, keys have to be hashable. In a List there is no need for hashing.

## 12. What are the different built-in data types available in Python?

Some of the built-in data types available in Python are as follows:

Numeric types: These are the data types used to represent numbers in Python.

int: It is used for Integers long: It is used for very large integers of non-limited length.

float: It is used for decimal numbers.

complex: This one is for representing complex numbers

Sequence types: These data types are used to represent sequence of characters or objects.

str: This is similar to String in Java. It can represent a sequence of characters.

bytes: This is a sequence of integers in the range of 0-255.

byte array: like bytes, but mutable (see below); only available in Python 3.x

list: This is a sequence of objects.

tuple: This is a sequence of immutable objects.

Sets: These are unordered collections.

set: This is a collection of unique objects.

frozen set: This is a collection of unique immutable objects.

Mappings: This is similar to a Map in Java.

Dict: This is also called hashmap. It has key value pair to store information by using hashing.

## 13. What is a Namespace in Python?

A Namespace in Python is a mapping between a name and an object. It is currently implemented as Python dictionary.

E.g. the set of built-in exception names, the set of built-in names, local names in a function

At different moments in Python, different Namespaces are created. Each Namespace in Python can have a different lifetime.

For the list of built-in names, Namespace is created when Python interpreter starts.

When Python interpreter reads the definition of a module, it creates global namespace for that module.

When Python interpreter calls a function, it creates local namespace for that function.

## 14. How will you concatenate multiple strings together in Python?

We can use following ways to concatenate multiple string together in Python:

I. use + operator:

```
E.g. >>> fname="John"
     >>> lname="Ray"
     >>> print fname+lname
 JohnRay
```

II. use join function:

```
E.g. >>> ''.join(['John','Ray'])
'JohnRay'
```

## 15. What is the use of Pass statement in Python?

The use of Pass statement is to do nothing. It is just a placeholder for a statement that is required for syntax purpose. It does not execute any code or command. Some of the use cases for pass statement are as follows:

I. Syntax purpose:

```
>>> while True:
        ... pass # Wait till user input is received
```

II. Minimal Class: It can be used for creating minimal classes:

```
>>> class MyMinimalClass:
        ... pass
```

III. Place-holder for TODO work: We can also use it as a placeholder for TODO work on a function or code that needs to be implemented at a later point of time.

```
>>> def initialization():
       ... pass # TODO
```

## 16. What is the use of Slicing in Python?

We can use Slicing in Python to get a substring from a String. The syntax of Slicing is very convenient to use.

E.g. In following example we are getting a substring out of the name John.

```
>>> name="John"

>>> name[1:3]

    'oh'
```

In Slicing we can give two indices in the String to create a Substring. If we do not give first index, then it defaults to 0.

```
E.g. >>> name="John"

    >>> name[:2]

        'Jo'
```

If we do not give second index, then it defaults to the size of the String.

```
>>> name="John"

>>> name[3:]

 'n'
```

## 17. What is the difference between Docstring in Python and Javadoc in Java?

A Docstring in Python is a string used for adding comments or summarizing a piece of code in Python.

The main difference between Javadoc and Docstring is that docstring is available during runtime as well. Whereas, Javadoc is removed from the Bytecode and it is not present in .class file.

We can even use Docstring comments at run time as an interactive help manual.

In Python, we have to specify docstring as the first statement of a code object, just after the def or class statement.

The docstring for a code object can be accessed from the '__doc__' attribute of that object.

## 18. How do you perform unit testing for Python code?

We can use the unit testing modules unittest or unittest2 to create and run unit tests for Python code.

We can even do automation of tests with these modules. Some of the main components of unittest are as follows:

I. Test fixture: We use test fixture to create preparation methods required to run a test. It can even perform post-test cleanup.
II. Test case: This is main unit test that we run on a piece of code. We can use Testcase base class to create new test cases.
III. Test suite: We can aggregate our unit test cases in a Test suite.
IV. Test runner: We use test runner to execute unit tests and produce reports of the test run.

## 19. What is the difference between an Iterator and Iterable in Python?

An Iterable is an object that can be iterated by an Iterator.

In Python, Iterator object provides _iter_() and next() methods.

In Python, an Iterable object has _iter_ function that returns an Iterator object.

When we work on a map or a for loop in Python, we can use next() method to get an Iterable item from the Iterator.

## 20. What is the use of Generator in Python?

We can use Generator to create Iterators in Python. A Generator is written like a regular function. It can make use yield statement to return data during the function call. In this way we can write complex logic that works as an Iterator.

A Generator is more compact than an Iterator due to the fact that _iter_() and next() functions are automatically created in a Generator.

Also within a Generator code, local variables and execution state are saved between multiple calls. Therefore, there is no need to add extra variables like self.index etc to keep track of iteration.

Generator also increases the readability of the code written in Python. It is a very simple implementation of an Iterator.

## 21. What is the significance of functions that start and end with _ symbol in Python?

Python provides many built-in functions that are surrounded by _ symbol at the start and end of the function name. As per Python documentation, double _ symbol is used for reserved names of functions.

These are also known as System-defined names.

Some of the important functions are:

Object._new_

Object._init_

Object._del_

## 22. What is the difference between xrange and range in Python?

In Python, we use range(0,10) to create a list in memory for 10 numbers.

Python provides another function xrange() that is similar to range() but xrange() returns a sequence object instead of list object. In xrange() all the values are not stored simultaneously in memory. It is a lazy loading based function.

But as per Python documentation, the benefit of xrange() over range() is very minimal in regular scenarios.

As of version 3.1, xrange is deprecated.

## 23. What is lambda expression in Python?

A lambda expression in Python is used for creating an anonymous function.

Wherever we need a function, we can also use a lambda expression.

We have to use lambda keyword for creating a lambda expression.

Syntax of lambda function is as follows: lambda argumentList: expression

E.g. lambda a,b: a+b

The above mentioned lambda expression takes two arguments and returns their sum.

We can use lambda expression to return a function.

A lambda expression can be used to pass a function as an argument in another function.

## 24. How will you copy an object in Python?

In Python we have two options to copy an object. It is similar to cloning an object in Java.

I. Shallow Copy: To create a shallow copy we call copy.copy(x). In a shallow copy, Python creates a new compound object based on the original object. And it tries to put references from the original object into copy object.

II. Deep Copy: To create a deep copy, we call copy.deepcopy(x). In a deep copy, Python creates a new object and recursively creates and inserts copies of the objects from original object into copy object. In a deep copy, we may face the issue of recursive loop due to infinite recursion.

## 25. What are the main benefits of using Python?

Some of the main benefits of using Python are as follows:

I. Easy to learn: Python is simple language. It is easy to learn for a new programmer.

II. Large library: There is a large library for utilities in Python that can be used for different kinds of applications.

III. Readability: Python has a variety of statements and expressions that are quite readable and very explicit in their use. It increases the readability of overall code.

IV. Memory management: In Python, memory management is built into the Interpreter. So a developer does not have to spend effort on managing memory among objects.

V. Complex built-in Data types: Python has built-in Complex data types like list, set, dict etc. These data types give very good performance as well as save time in coding new features.

## 26. What is a metaclass in Python?

A metaclass in Python is also known as class of a class. A class defines the behavior of an instance. A metaclass defines the behavior of a class.

One of the most common metaclass in Python is type. We can subclass type to create our own metaclass.

We can use metaclass as a class-factory to create different types of classes.

## 27. What is the use of frozenset in Python?

A frozenset is a collection of unique values in Python. In addition to all the properties of set, a frozenset is immutable and hashable.

Once we have set the values in a frozenset, we cannot change. So we cannot use and update methods from set on frozenset.

Being hashable, we can use the objects in frozenset as keys in a Dictionary.

## 28. What is Python Flask?

Python Flask is a micro-framework based on Python to develop a web application.

It is a very simple application framework that has many extensions to build an enterprise level application.

Flask does not provide a data abstraction layer or form validation by default. We can use external libraries on top of Flask to perform such tasks.

## 29. What is None in Python?

None is a reserved keyword used in Python for null objects. It is neither a null value nor a null pointer. It is an actual object in Python. But there is only one instance of None in a Python environment.

We can use None as a default argument in a function.

During comparison we have to use “is” operator instead of “==” for None.

## 30. What is the use of zip() function in Python?

In Python, we have a built-in function zip() that can be used to aggregate all the Iterable objects of an Iterator.

We can use it to aggregate Iterable objects from two iterators as well.

```
E.g. list_1 = ['a', 'b', 'c']

     list_2 = ['1', '2', '3']

    for a, b in zip(list_1, list_2):

        print a, b
```

Output:

```
a1

b2

c3
```

By using zip() function we can divide our input data from different sources into fixed number of sets.

## 31. What is the use of // operator in Python?

Python provides // operator to perform floor division of a number by another.

The result of // operator is a whole number (without decimal part) quotient that we get by dividing left number with right number.

It can also be used floordiv(a,b).

E.g. 10// 4 = 2

-10//4 = -3

## 32. What is a Module in Python?

A Module is a script written in Python with import statements, classes, functions etc. We can use a module in another Python script by importing it or by giving the complete namespace.

With Modules, we can divide the functionality of our application in smaller chunks that can be easily managed.

## 33. How can we create a dictionary with ordered set of keys in Python?

In a normal dictionary in Python, there is no order maintained between keys. To solve this problem, we can use OrderDict class in Python. This class is available for use since version 2.7.

It is similar to a dictionary in Python, but it maintains the insertion order of keys in the dictionary collection.

## 34. Python is an Object Oriented programming language or a functional programming language?

Python uses most of the Object Oriented programming concepts. But we can also do functional programming in Python. As per the opinion of experts, Python is a multi-paradigm programming language.

We can do functional, procedural, object-oriented and imperative programming with the help of Python.

## 35. How can we retrieve data from a MySQL database in a Python script?

To retrieve data from a database we have to make use of the module available for that database. For MySQL database, we import MySQLdb module in our Python script.

We have to first connect to a specific database by passing URL, username, password and the name of database.

Once we establish the connection, we can open a cursor with cursor() function. On an open cursor, we can run fetch() function to execute queries and retrieve data from the database tables.

## 36. What is the difference between append() and extend() functions of a list in Python?

In Python, we get a built-in sequence called list. We can call standard functions like append() and extend() on a list.

We call append() method to add an item to the end of a list.

We call extend() method to add another list to the end of a list.

In append() we have to add items one by one. But in extend() multiple items from another list can be added at the same time.

## 37. How will you handle an error condition in Python code?

We can implement exception handling to handle error conditions in Python code. If we are expecting an error condition that we cannot handle, we can raise an error with appropriate message.

```
E.g.    >>> if student_score < 0:

                raise ValueError(“Score cannot be negative”)
```

If we do not want to stop the program, we can just catch the error condition, print a message and continue with our program.

E.g. In following code snippet we are catching the error and continuing with the default value of age.

```
#!/usr/bin/python

try:
    age=18+'duration'
except:
   print("duration has to be a number")
age=18
print(age)
```

## 38. What is the difference between split() and slicing in Python?

Both split() function and slicing work on a String object. By using split() function, we can get the list of words from a String.

E.g. 'a b c '.split() returns [‘a’, ‘b’, ‘c’]

Slicing is a way of getting substring from a String. It returns another String.

E.g. >>> 'a b c'[2:3] returns b

## 39. How will you check in Python, if a class is subclass of another class?

Python provides a useful method issubclass(a,b) to check whether class a is a subclass of b.

E.g. int is not a subclass of long

```
 >>> issubclass(int,long)
False
```

bool is a subclass of int

```
>>> issubclass(bool,int)
True
```

## 40. How will you debug a piece of code in Python?

In Python, we can use the debugger pdb for debugging the code. To start debugging we have to enter following lines on the top of a Python script.

```
 import pdb
pdb.set_trace()
```

After adding these lines, our code runs in debug mode. Now we can use commands like breakpoint, step through, step into etc for debugging.

## 41. How do you profile a Python script?

Python provides a profiler called cProfile that can be used for profiling Python code.

We can call it from our code as well as from the interpreter.

It gives use the number of function calls as well as the total time taken to run the script.

We can even write the profile results to a file instead of standard out.

## 42. What is the difference between ‘is’ and ‘==’ in Python?

We use ‘is’ to check an object against its identity.

We use ‘==’ to check equality of two objects.

```
E.g. >>> lst = [10,20, 20]
     >>> lst == lst[:]
     True
     >>> lst is lst[:]
    False
```

## 43. How will you share variables across modules in Python?

We can create a common module with variables that we want to share.

This common module can be imported in all the modules in which we want to share the variables.

In this way, all the shared variables will be in one module and available for sharing with any new module as well.

## 44. How can we do Functional programming in Python?

In Functional Programming, we decompose a program into functions. These functions take input and after processing give an output. The function does not maintain any state.

Python provides built-in functions that can be used for Functional programming.

Some of these functions are:

I. Map()  
II. reduce()  
III. filter()

Event iterators and generators can be used for Functional programming in Python.

## 45. What is the improvement in enumerate() function of Python?

In Python, enumerate() function is an improvement over regular iteration.

The enumerate() function returns an iterator that gives (0, item[0]).

```
E.g. >>> thelist=['a','b']

       >>> for i,j in enumerate(thelist):

       ... print i,j

       ...

       0 a

       1 b
```

## 46. How will you execute a Python script in Unix?

To execute a Python script in Unix, we need to have Python executor in Unix environment. In addition to that we have to add following line as the first line in a Python script file.

```
#!/usr/local/bin/python
```

This will tell Unix to use Python interpreter to execute the script.

## 47. What are the popular Python libraries used in Data analysis?

Some of the popular libraries of Python used for Data analysis are:

I. Pandas: Powerful Python Data Analysis Toolkit.  
II. SciKit: This is a machine learning library in Python.  
III. Seaborn: This is a statistical data visualization library in Python.  
IV. SciPy: This is an open source system for science, mathematics and engineering implemented in Python.

## 48. What is the output of following code in Python?

```
>>> thelist=['a','b']
>>> print thelist[3:]
```

Ans: The output of this code is following:

```
[]
```

Even though the list has only 2 elements, the call to thelist with index 3 does not give any index error.

## 4. What is the output of following code in Python?

```
>>>name='John Smith'
>>>print name[:5] + name[5:]
```

Ans: Output of this will be

```
John Smith
```

This is an example of Slicing. Since we are slicing at the same index, the first name[:5] gives the substring name upto 5th location excluding 5th location. The name[5:] gives the rest of the substring of name from the 5th location. So we get the full name as output.

## 50. If you have data with name of customers and their location, which data type will you use to store it in Python?

In Python, we can use dict data type to store key value pairs. In this example, customer name can be the key and their location can be the value in a dict data type.

Dictionary is an efficient way to store data that can be looked up based on a key.

## 51. What is __init__?

__init__ is a contructor method in Python and is automatically called to allocate memory when a new object/instance is created. All classes have a **__init__** method associated with them. It helps in distinguishing methods and attributes of a class from local variables.

## 52. What is the difference between Python Arrays and lists?

Arrays in python can only contain elements of same data types i.e., data type of array should be homogeneous. It is a thin wrapper around C language arrays and consumes far less memory than lists.

Lists in python can contain elements of different data types i.e., data type of lists can be heterogeneous. It has the disadvantage of consuming large memory.

## 53. What is slicing in Python?

As the name suggests, 'slicing' is taking parts of.

Syntax for slicing is **[start : stop : step]**

**start** is the starting index from where to slice a list or tuple

**stop** is the ending index or where to sop.

**step** is the number of steps to jump.

Default value for **start** is 0, **stop** is number of items, **step** is 1.

Slicing can be done on **strings, arrays, lists**, and **tuples**.

## 54. What is docstring in Python?

Documentation string or docstring is a multiline string used to document a specific code segment.

The docstring should describe what the function or method does.

## 55. What are unit tests in Python?

Unit test is a unit testing framework of Python.

Unit testing means testing different components of software separately. Can you think about why unit testing is important? Imagine a scenario, you are building software that uses three components namely A, B, and C. Now, suppose your software breaks at a point time. How will you find which component was responsible for breaking the software? Maybe it was component A that failed, which in turn failed component B, and this actually failed the software. There can be many such combinations.

This is why it is necessary to test each and every component properly so that we know which component might be highly responsible for the failure of the software.

## 56. What is break, continue and pass in Python?

| | |
|---|---|
| Break | The break statement terminates the loop immediately and the control flows to the statement after the body of the loop. |
| Continue | The continue statement terminates the current iteration of the statement, skips the rest of the code in the current iteration and the control flows to the next iteration of the loop. |
| Pass | As explained above, the pass keyword in Python is generally used to fill up empty blocks and is similar to an empty statement represented by a semi-colon in languages such as Java, C++, Javascript, etc. |

## 57. What are global, protected and private attributes in Python?

**Global** variables are public variables that are defined in the global scope. To use the variable in the global scope inside a function, we use the global keyword.

**Protected** attributes are attributes defined with an underscore prefixed to their identifier

eg. _sara.

They can still be accessed and modified from outside the class they are defined in but a responsible developer should refrain from doing so.

**Private** attributes are attributes with double underscore prefixed to their identifier

eg. __ansh.

They cannot be accessed or modified from the outside directly and will result in an AttributeError if such an attempt is made.

## 58. What is an Interpreted language?

An Interpreted language executes its statements line by line. Languages such as Python, Javascript, R, PHP, and Ruby are prime examples of Interpreted languages. Programs written in an interpreted language runs directly from the source code, with no intermediary compilation step.

## 59. What is a dynamically typed language?

Before we understand a dynamically typed language, we should learn about what typing is. **Typing** refers to type-checking in programming languages. In a **strongly-typed** language, such as Python, **"1" + 2** will result in a type error since these languages don't allow for "type-coercion" (implicit conversion of data types). On the other hand, a **weakly-typed** language, such as Javascript, will simply output **"12"** as result.

Type-checking can be done at two stages -

**Static** - Data Types are checked before execution.

**Dynamic** - Data Types are checked during execution.

Python is an interpreted language, executes each statement line by line and thus type-checking is done on the fly, during execution. Hence, Python is a Dynamically Typed Language.

## 60. What is Python? What are the benefits of using Python

Python is a high-level, interpreted, general-purpose programming language. Being a general-purpose language, it can be used to build almost any type of application with the right tools/libraries. Additionally, python supports objects, modules, threads, exception-handling, and automatic memory management which help in modelling real-world problems and building applications to solve these problems.

**Benefits of using Python:**

- Python is a general-purpose programming language that has a simple, easy-to-learn syntax that emphasizes readability and therefore reduces the cost of program maintenance. Moreover, the language is capable of scripting, is completely open-source, and supports third-party packages encouraging modularity and code reuse.
- Its high-level data structures, combined with dynamic typing and dynamic binding, attract a huge community of developers for Rapid Application Development and deployment.

## 61. Explain how to delete a file in Python?

Use command **os.remove(file_name)**

## 62.What are negative indexes and why are they used?

Negative indexes are the indexes from the end of the list or tuple or string.

**Arr[-1]** means the last element of array **Arr[]**

## 63. What does *args and **kwargs mean?

### *args

*args is a special syntax used in the function definition to pass variable-length arguments.

"*" means variable length and "args" is the name used by convention. You can use any other.

### **kwargs

**kwargs is a special syntax used in the function definition to pass variable-length keyworded arguments.

Here, also, "kwargs" is used just by convention. You can use any other name.

Keyworded argument means a variable that has a name when passed to a function.

It is actually a dictionary of the variable names and its value.

## 64. What is the difference between .py and .pyc files?

.py files contain the source code of a program. Whereas, .pyc file contains the bytecode of your program. We get bytecode after compilation of .py file (source code). .pyc files are not created for all the files that you run. It is only created for the files that you import.

Before executing a python program python interpreter checks for the compiled files. If the file is present, the virtual machine executes it. If not found, it checks for .py file. If found, compiles it to .pyc file and then python virtual machine executes it.

Having .pyc file saves you the compilation time.

## 65. What is the use of help() and dir() functions?

**help()** function in Python is used to display the documentation of modules, classes, functions, keywords, etc. If no parameter is passed to the help() function, then an interactive **help utility** is launched on the console.

**dir()** function tries to return a valid list of attributes and methods of the object it is called upon. It behaves differently with different objects, as it aims to produce the most relevant data, rather than the complete information.

For Modules/Library objects, it returns a list of all attributes, contained in that module.

For Class Objects, it returns a list of all valid attributes and base attributes.

With no arguments passed, it returns a list of attributes in the current scope.

## 66. What is PYTHONPATH in Python?

PYTHONPATH is an environment variable which you can set to add additional directories where Python will look for modules and packages. This is especially useful in maintaining Python libraries that you do not wish to install in the global default location.

## 67. How do you copy an object in Python?

In Python, the assignment statement (= operator) does not copy objects. Instead, it creates a binding between the existing object and the target variable name. To create copies of an object in Python, we need to use the **copy** module. Moreover, there are two ways of creating copies for the given object using the **copy** module -

**Shallow Copy** is a bit-wise copy of an object. The copied object created has an exact copy of the values in the original object. If either of the values is a reference to other objects, just the reference addresses for the same are copied.
**Deep Copy** copies all values recursively from source to target object, i.e. it even duplicates the objects referenced by the source object.

## 68. Why is finalize used?

Finalize method is used for freeing up the unmanaged resources and clean up before the garbage collection method is invoked. This helps in performing memory management tasks.

## 69. Differentiate between new and override modifiers.

The new modifier is used to instruct the compiler to use the new implementation and not the base class function. The Override modifier is useful for overriding a base class function inside the child class.

## 70. Is it possible to call parent class without its instance creation?

Yes, it is possible if the base class is instantiated by other child classes or if the base class is a static method.

## 71. Are access specifiers used in python?

Python does not make use of access specifiers specifically like private, public, protected, etc. However, it does not derive this from any variables. It has the concept of imitating the behavior of variables by making use of a single (protected) or double underscore (private) as prefixed to the variable names. By default, the variables without prefixed underscores are public.

## 72. What is the difference between a mutable data type and an immutable data type?

Mutable data types:

**Definition:** Mutable data types are those that can be modified after their creation.

**Examples:** List, Dictionary, Set.

**Characteristics:** Elements can be added, removed, or changed.

**Use Case:** Suitable for collections of items where frequent updates are needed.

Immutable data types:

**Definition:** Immutable data types are those that cannot be modified after their creation.

**Examples:** Numeric (int, float), String, Tuple.

**Characteristics:** Elements cannot be changed once set; any operation that appears to modify an immutable object will create a new object.

## 73. What is the Python “with” statement designed for?

The `with` statement is used for exception handling to make code cleaner and simpler. It is generally used for the management of common resources like creating, editing, and saving a file.

**Example**:

Instead of writing multiple lines of open, try, finally, and close, you can create and write a text file using the `with` statement. It is simple.

## 74. Why use else in try/except construct in Python?

`try:` and `except:` are commonly known for exceptional handling in Python, so where does `else:` come in handy? `else:` will be triggered when no exception is raised.

**Example**:

Let’s learn more about `else:` with a couple of examples.

1. On the first try, we entered 2 as the numerator and “d” as the denominator. Which is incorrect, and `except:` was triggered with “Invalid input!”.
2. On the second try, we entered 2 as the numerator and 1 as the denominator and got the result 2. No exception was raised, so it triggered the `else:` printing the message “Division is successful.”

## 75. What type of language is python? Programming or scripting?

**Ans:** Python is capable of scripting, but in general sense, it is considered as a general-purpose programming language.

## 76. Python an interpreted language. Explain.

**Ans:** An interpreted language is any programming language which is not in machine-level code before runtime. Therefore, Python is an interpreted language.

## 77. What are the key features of Python?

- Python is an **interpreted** language. That means that, unlike languages like *C* and its variants, Python does not need to be compiled before it is run. Other interpreted languages include *PHP* and *Ruby*.

- Python is **dynamically typed**, this means that you don't need to state the types of variables when you declare them or anything like that. You can do things like x=111 and then x="I'm a string" without error
- Python is well suited to **object orientated programming** in that it allows the definition of classes along with composition and inheritance. Python does not have access specifiers (like C++'s public, private).
- In Python, **functions** are **first-class objects**. This means that they can be assigned to variables, returned from other functions and passed into functions. Classes are also first class objects
- **Writing Python code is quick** but running it is often slower than compiled languages. Fortunately, Python allows the inclusion of C-based extensions so bottlenecks can be optimized away and often are. The numpy package is a good example of this, it's really quite quick because a lot of the number-crunching it does isn't actually done by Python
- Python finds **use in many spheres** – web applications, automation, scientific modeling, big data applications and many more. It's also often used as "glue" code to get other languages and components to play nice. Learn more about Big Data and its applications from the Azure data engineer training course.

## 78. What are Keywords in Python?

Ans: Keywords in python are reserved words that have special meaning.They are generally used to define type of variables. Keywords cannot be used for variable or function names. There are following 33 keywords in python-

- And
- Or
- Not
- If
- Elif
- Else
- For
- While
- Break
- As
- Def
- Lambda
- Pass
- Return
- True
- False
- Try
- With
- Assert
- Class

- Continue
- Del
- Except
- Finally
- From
- Global
- Import
- In
- Is
- None
- Nonlocal
- Raise
- Yield

## 79. 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 or variables. A Python module is a .py file containing executable code.

Some of the commonly used built-in modules are:

- os
- sys
- math
- random
- data time
- JSON

## 80. 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.

## 81. Is python case sensitive?

**Ans:** Yes. Python is a case sensitive language.

## 82. What is type conversion in Python?

**Ans:** Type conversion refers to the conversion of one data type into 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 function converts real numbers to complex(real,imag) number.

## 83. 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 your PC using the following command on your command prompt: cmd python.
- Then go to advanced system settings and add a new variable and name it as PYTHON_NAME and paste the copied path.
- 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%

## 84. Is indentation required in python?

**Ans:** Indentation is necessary for Python. It specifies a block of code. All code within loops, classes, functions, etc is specified within an indented block. It is usually done using four space characters. If your code is not indented necessarily, it will not execute accurately and will throw errors as well.

## 85.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 function, the **def** keyword is used.

Eg - def fun():

## 86. What is self in Python?

**Ans:** Self is an instance or an object of a class. In Python, this is explicitly included as the first parameter. However, this is not the case in Java where it's optional. It helps to differentiate between the methods and attributes of a class with local variables.

The self variable in the init method refers to the newly created object while in other methods, it refers to the object whose method was called.

## 87. What does [::-1] do?

**Ans:** [::-1] is used to reverse the order of an array or a sequence.

## 88. How do you write comments in python?

**Ans:** Comments in Python start with a # character. However, alternatively at times, commenting is done using docstrings(strings enclosed within triple quotes).

## 89. 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 already consists of a capital letter at the beginning, then, it returns the original string.

## 90**.** 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 prefixed by a #. You can also a very good shortcut method to comment 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 all the lines where you introduced your cursor.

## 91. 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 result.

**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.

## 92. Whenever Python exits, why isn’t all the memory de-allocated?

**Ans:**

1. Whenever Python exits, especially those Python modules which are having circular references to other objects or the objects that are referenced from the global namespaces are not always de-allocated or freed.
2. It is impossible to de-allocate those portions of memory that are reserved by the C library.
3. On exit, because of having its own efficient clean up mechanism, Python would try to de-allocate/destroy every other object.

## 93. What is a dictionary in Python?

**Ans:** The built-in datatypes in Python is called dictionary. It defines one-to-one relationship between keys and values. Dictionaries contain pair of keys and their corresponding values. Dictionaries are indexed by keys.

dict={'Country':'India','Capital':'Delhi','PM':'Modi'}

## 94. 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 consists of the true or false values with a statement that has to be evaluated for it.

**Syntax**:
[on_true] if [expression] else [on_false]x, y = 25, 50big = x if x < y else y

Eg : The expression gets evaluated like if x<y else y, in this case if x<y is true then the value is returned as big=x and if it is incorrect then big=y will be sent as a result.

## 95. What does len() do?

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

## 96. 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 different string
- **subn()** – it is similar to sub() and also returns the new string along with the no. of replacements.

## 97. How to remove values to a python array?

**Ans:** Array elements can be removed using **pop()** or **remove()** method. The difference between these two functions is that the former returns the deleted value whereas the latter does not.

## 98. Does Python have OOPS concepts?

**Ans:** Python is an object-oriented programming language. This means that any program can be solved in python by creating an object model. However, Python can be treated as a procedural as well as structural language.

## 99. How is Multithreading achieved in Python?

**Ans:**

1. Python has a multi-threading package but if you want to multi-thread to speed your code up, then it's usually not a good idea to use it.
2. Python has a construct called the Global Interpreter Lock (GIL). The GIL makes sure that only one of your 'threads' can execute at any one time. A thread acquires the GIL, does a little work, then passes the GIL onto the next thread.
3. This happens very quickly so to the human eye it may seem like your threads are executing in parallel, but they are really just taking turns using the same CPU core.
4. All this GIL passing adds overhead to execution. This means that if you want to make your code run faster then using the threading package often isn't a good idea.

## 100. What is the process of compilation and linking in python?

**Ans:** The compiling and linking allow the new extensions to be compiled properly without any error and the linking can be done only when it passes the compiled procedure. If the dynamic loading is used then it depends on the style that is being provided with the system. The python interpreter can be used to provide the dynamic loading of the configuration setup files and will rebuild the interpreter.

The steps that are required in this as:

1. Create a file with any name and in any language that is supported by the compiler of your system. For example file.c or file.cpp
2. Place this file in the Modules/ directory of the distribution which is getting used.
3. Add a line in the file Setup.local that is present in the Modules/ directory.
4. Run the file using spam file.o
5. After a successful run of this rebuild the interpreter by using the make command on the top-level directory.
6. If the file is changed then run rebuildMakefile by using the command as 'make Makefile'.

## 101. What is an ordered dictionary in Python?

Ans. OrderedDict() is used to maintains the sequence in which keys are added, ensuring that the order is preserved during iteration. In contrast, a standard dictionary does not guarantee any specific order when iterated, providing values in an arbitrary sequence. OrderedDict() distinguishes itself by retaining the original insertion order of items.

## 102. What is the difference between ‘return’ and ‘yield’ keywords?

**Ans.** In Python, ‘return’ sends a value and terminates a function, while ‘yield’ produces a value but retains the function’s state, allowing it to resume from where it left off.

| YIELD | RETURN |
|---|---|
| It replace the return of a function to suspend its execution without destroying local variables. | It exits from a function and handing back a value to its caller. |
| It is used when the generator returns an intermediate result to the caller. | It is used when a function is ready to send a value. |
| Code written after yield statement execute in next function call. | while, code written after return statement wont execute. |
| It can run multiple times. | It only runs single time. |
| Yield statement function is executed from the last state from where the function get paused. | Every function calls run the function from the start. |

## 103. What’s the difference between a set() and a frozenset()?

**Ans.** Set and frozenset are two built-in collection data types in Python that are used to store a collection of unique elements. While set is mutable, meaning that we can add, remove, or change elements in a set, frozenset is immutable and cannot be modified after creation.

## 104. How to import modules in python?

Ans. Modules can be imported using the import keyword. You can import modules in three ways-

Example:

```
import array            #importing using the original module name
import array as arr     # importing using an alias name
from array import *     #imports everything present in the array module
```

## 105. Explain Inheritance in Python with an example.

**Ans:** Inheritance allows One class to gain all the members(say attributes and methods) of another class. Inheritance provides code reusability, makes it easier to create and maintain an application. The class from which we are inheriting is called super-class and the class that is inherited is called a derived / child class.

They are different types of inheritance supported by Python:

1. **Single Inheritance** – where a derived class acquires the members of a single super class.
2. **Multi-level inheritance** – a derived class d1 in inherited from base class base1, and d2 are inherited from base2.
3. **Hierarchical inheritance** – from one base class you can inherit any number of child classes
4. **Multiple inheritance** – a derived class is inherited from more than one base class.

## 106. How are classes created in Python?

**Ans:** Class in Python is created using the **class** keyword.

Example**:**

```
Class Employee

   Def __init__(self,name)

      Self.name = name

E1 = Employee("abc")

Print(E1.name)
```

## 107. Does python support multiple inheritance?

**Ans:** Multiple inheritance means that a class can be derived from more than one parent classes. Python does support multiple inheritance, unlike Java.

## 108. What is Polymorphism in Python?

**Ans:** Polymorphism means the ability to take multiple forms. So, for instance, if the parent class has a method named ABC then the child class also can have a method with the same name ABC having its own parameters and variables. Python allows polymorphism.

## 109. Define encapsulation in Python?

**Ans:** Encapsulation means binding the code and the data together. A Python class in an example of encapsulation.

## 110. How do you do data abstraction in Python?

**Ans:** Data Abstraction is providing only the required details and hiding the implementation from the world. It can be achieved in Python by using interfaces and abstract classes.

## 111. Does python make use of access specifiers?

**Ans:** Python does not deprive access to an instance variable or function. Python lays down the concept of prefixing the name of the variable, function or method with a single or double underscore to imitate the behavior of protected and private access specifiers.

## 112. What does an object() do?

**Ans:** It returns a featureless object that is a base for all classes. Also, it does not take any parameters.

## 113. How to access an element of a list?

Ans. The element in a list can be accessed using list_name [index]. For instance:

Given a list [1, 2, 3, 4].

The indexing of the list starts from 0. The first element of the list can be accessed using list[0],

which will print element “1”.The second element can be accessed using list[1] and so on.

## 114. Discuss different ways of deleting an element from a list.

Ans. There are two ways in which we can delete elements from the list:

1. By using the remove() function

   The remove () function deletes the mentioned element from the list.

```
list1 = [1, 2, 3, 4]
list1.remove(2)
print(list1)
```

**Output:** [1, 3, 4]

2. By using the pop() function
   Pop() function delete element mentioned at a specific index from the list

```
list1.pop(1)
print(list1)
```

Output: [1, 4]

## 115. Which is faster, Python list or Numpy arrays, and why?

Ans. NumPy arrays are notably faster than Python lists for numerical operations. NumPy is an open-source library designed for efficient array operations in Python, leveraging optimized implementations in C. Unlike Python lists, which are interpreted, NumPy arrays are executed in a compiled language, enhancing their performance significantly.

Python also includes a built-in array module for basic operations, which can be imported using import array as arr.

## 116. What are Python sets? Explain some of the properties of sets.

Ans. In Python, a set is an unordered collection of unique objects. Sets are often used to store a collection of distinct objects and to perform membership tests (i.e., to check if an object is in the set). Sets are defined using curly braces ({ and }) and a comma-separated list of values.

Here are some key properties of sets in Python:

- Sets are unordered: Sets do not have a specific order, so you cannot index or slice them like you can with lists or tuples.
- Sets are unique: Sets only allow unique objects, so if you try to add a duplicate object to a set, it will not be added.

- Sets are mutable: You can add or remove elements from a set using the add and remove methods.
- Sets are not indexed: Sets do not support indexing or slicing, so you cannot access individual elements of a set using an index.
- Sets are not hashable: Sets are mutable, so they cannot be used as keys in dictionaries or as elements in other sets. If you need to use a mutable object as a key or an element in a set, you can use a tuple or a frozen set (an immutable version of a set).

## 117. Explain the logical operations in Python.

Ans. In Python, the logical operations and, or, and not can be used to perform boolean operations on truth values (True and False).

The **and** operator returns True if both the operands are True, and False otherwise.

The **or** operator returns True if either of the operands is True, and False if both operands are False.

The **not** operator inverts the boolean value of its operand. If the operand is True, **not** return False, and if the operand is False, **not** return True.

## 118. Explain the top 5 functions used for Python strings.

Ans. Here are the top 5 Python string functions:

- **len():** This function returns the length of a string.

```
 s = 'Hello, World!'
print(len(s))
13
```

- **strip():** This function removes leading and trailing whitespace from a string.

```
 s = '   Hello, World!   '
 print(s.strip())
'Hello, World!'
```

- **replace():** This function replaces all occurrences of a specified string with another string.

```
 s = 'Hello, World!'
 print(s.replace('World', 'Universe'))
'Hello, Universe!'
```

- **split():** This function splits a string into a list of substrings based on a delimiter.

```
 s = 'Hello, World!'
 print(s.split(','))
['Hello', ' World!']
```

- **upper() and lower():** These functions convert a string to uppercase or lowercase, respectively.

```
s = 'Hello, World!'
print(s.upper())
'HELLO, WORLD!'
s.lower()
'hello, world!'
```

In addition to them, string also has capitalize, isalnum, isalpha, and other methods.

## 119. What is the use of the continue keyword in Python?

Ans. Continue is used in a loop to skip over the current iteration and move on to the next one. When continue is encountered, the current iteration of the loop is terminated, and the next one begins

## 120. What is the use of the pass keyword in Python?

Ans. Pass is a null statement that does nothing. It is often used as a placeholder where a statement is required syntactically, but no action needs to be taken. For example, if you want to define a function or a class but haven't yet decided what it should do, you can use the pass as a placeholder.

## 121. Name 2 mutable and 2 immutable data types in Python.

Ans. 2 **mutable** data types are **Dictionary** and **List**. You can change/edit the values in a Python dictionary and a list. It is not necessary to make a new list which means that it satisfies the property of mutability.

2 **immutable** data types are **Tuples** and **String**. You cannot edit a string or a value in a tuple once it is created. You need to either assign the values to the tuple or make a new tuple.

## 122. What are Python functions, and how do they help in code optimization?

Ans. In Python, a function is a block of code that can be called by other parts of your program. Functions are useful because they allow you to reuse code and divide your code into logical blocks that can be tested and maintained separately.

To call a function in Python, you simply use the function name followed by a pair of parentheses and any necessary arguments. The function may or may not return a value that depends on the usage of the turn statement.

Functions can also help in code optimization:

- Code reuse: Functions allow you to reuse code by encapsulating it in a single place and calling it multiple times from different parts of your program. This can help to reduce redundancy and make your code more concise and easier to maintain.
- Improved readability: By dividing your code into logical blocks, functions can make your code more readable and easier to understand. This can make it easier to identify bugs and make changes to your code.
- Easier testing: Functions allow you to test individual blocks of code separately, which can make it easier to find and fix bugs.
- Improved performance: Functions can also help to improve the performance of your code by allowing you to use optimized code libraries or by allowing the Python interpreter to optimize the code more effectively.

## 123. What is the use of the 'assert' keyword in Python?

Ans. In Python, the assert statement is used to test a condition. If the condition is True, then the program continues to execute. If the condition is False, then the program raises an AssertionError exception.

The assert statement is often used to check the internal consistency of a program. For example, you might use an assert statement to check that a list is sorted before performing a binary search on the list.

It's important to note that the assert statement is used for debugging purposes and is not intended to be used as a way to handle runtime errors. In production code, you should use try and except blocks to handle exceptions that might be raised at runtime.

## 124. Where can we use a tuple instead of a list?

Ans. We can use tuples as dictionary keys as they are hashable. Since tuples are immutable, it is safer to use if we don't want values to change. Tuples are faster and have less memory, so we can use tuples to access only the elements.

## 125. Is removing the first item or last item takes the same time in the Python list?

Ans. No, removing the last item is O(1), while removing the first item is O(n).

## 126. How can we remove any element from a list efficiently?

Ans. We can use a deque from the collections module, which is implemented as a doubly linked list, to remove elements faster at any index.

## 127. Why do floating-point calculations seem inaccurate in Python?

Ans. While representing floating point numbers like 2.5 is easier in the decimal system, representing the same in a binary system needs a lot of bits of information. Since the number of bits is limited, there can be rounding errors in floating-point calculations.

## 128. What is the use of generators in Python?

Ans. Since the generator doesn't produce all the values at the same time, it saves memory if we use the generator to process the sequence of values without the need to save the initial values.

## 129. How can we iterate through multiple lists at the same time?

Ans. We can use zip() function to aggregate multiple lists and return an iterator of tuples where each tuple contains elements of different lists of the same index.

## 130. What are the various ways of adding elements to a list?

Ans. Here are the various ways of adding elements to a list:

1. We can use insert() to add a given index to the existing list.
2. We can use append() to add at the end of a list a single item.
3. We can use extend() to add each element of an iterable(list, tuple, or set) separately at the end of the list.
4. We can also use the + operator to concatenate two lists, similar to extend, but it works only with one list to another list but not one list to another tuple or set.

## 131. What is the difference between del and remove on lists?

Ans. ‘Remove’ removes the first occurrence of a given element. ‘Del’ removes elements based on the given index.

## 132. What are the different types of variables in Python OOP?

Ans: The 3 different types of variables in Python OOP (object-oriented programming) are:

1. Class variables: They are defined inside the class but outside other methods and are available to access for any instance of the class.
2. Instance variables: They are defined for each instance of the class separately and accessible by that instance of the class.

3. Local variables: They are defined inside the method and accessible only inside that method.

## 134. What are the different types of methods in Python OOP?

Ans: The 3 different types of methods in Python OOP are:

1. Class methods: They can access only class variables and are used to modify the class state.
2. Instance methods: They can access both class and instance variables and are used to modify the object (instance of a class) state.
3. Static methods: They can’t access either class or instance variables and can be used for functions that are suitable to be in class without accessing class or instance variables.

## 135. How can we check for an array with only zeros?

Ans. We can use the size method of the array to check whether it returns 0. Then it means the array can contain only zeros.

## 136. What’s the difference between split() and array_split()?

Ans. The split() function is used to split an array in n number of equal parts. If it is not possible to split it into an equal number of parts, split() raises an error. On the other hand, array_split() splits an array into n unequal parts.

## 137. How can we remove the leading whitespace for a string?

Ans. We can use the .lstrip() method to remove the leading whitespace for a string.

## 138. What is enumerate() in Python?

Ans. enumerate() in Python iterates a sequence and returns the index position and its corresponding value.

## 139. What is a callable object in Python?

Ans. An object which can invoke a process is a callable object. It uses the **call** method. Functions are examples of that. Callable objects have () at the end, while non-callable methods don't have () at the end.

## 140. What is the purpose of garbage collection in Python?

Python automatically frees unused memory resources, optimizing memory usage.

## 141. Explain the concept of class and object in Python.

- Classes define blueprints for objects (instances) with attributes and methods.
- Objects represent specific instances of a class with unique attribute values.

## 142. How do you implement a loop in Python? What are the different loop types?

- for loop: Iterates through a sequence of elements.
- while loop: Executes a block of code repeatedly while a condition is true.
- nested loops: Loops within loops for complex iteration needs.

## 143. What is file handling in Python? What are the various file-handling operations in Python?

File handling also known as Python I/O involves working with files on a computer's file system using Python as a programming language.

Python File Handling Operations can be categorized into the following categories:

1. Reading a file

2. Creating a file
3. Writing in a file
4. Deleting a file

## 144. Do we need to declare variables with respective data types in Python?

No. Python is a dynamically typed language, i.e., the Python Interpreter automatically identifies the data type of a variable based on the type of value assigned.

## 145. What do you know about Dict and List Comprehension?

Python Comprehensions are like decorators that help to build altered and filtered lists, dictionaries, or sets from a given list, dictionary, or a set. Comprehension is a powerful feature in Python that offers a convenient way to create lists, dictionaries, and sets with concise expressions. It eliminates the need for explicit loops, which can help reduce code size and save time during development.

Comprehensions are beneficial in the following scenarios:

- Performing mathematical operations on the entire list
- Performing conditional filtering operations on the entire list
- Combining multiple lists into one
- Flattening a multi-dimensional list

## 146. Is Python fully object-oriented?

Python follows an object-oriented programming paradigm and has all the basic OOPs concepts, such as inheritance, polymorphism, and more, with the exception of access specifiers. Python doesn’t support strong encapsulation (adding a private keyword before data members). Although, it has a convention that can be used for data hiding, i.e., prefixing a data member with two underscores.

## 147. Explain all file processing modes supported in Python?

Python has various file-processing modes.

For opening files, there are three modes:

- read-only mode (r)
- write-only mode (w)
- read–write mode (rw)

For opening a text file using the above modes, we will have to append ‘t’ with them as follows:

- read-only mode (rt)
- write-only mode (wt)
- read–write mode (rwt)

Similarly, a binary file can be opened by appending ‘b’ with them as follows:

- read-only mode (rb)
- write-only mode (wb)
- read–write mode (rwb)

To append the content in the files, we can use the append mode (a):

- For text files, the mode would be ‘at’
- For binary files, it would be ‘ab’

## 148. What do file-related modules in Python do? Can you name some file-related modules in Python?

Python comes with some file-related modules that have functions to manipulate text files and binary files in a file system. These modules can be used to create text or binary files, update content by carrying out operations like copy, delete, and more.

Some file-related modules are os, os.path, and shutil.os. The os.path module has functions to access the file system, while the shutil.os module can be used to copy or delete files.

## 149. How will you reverse a list in Python?

To reverse a list in Python, you can use the slicing technique. Here's a brief explanation of the process:

Start with the original list that you want to reverse.

Use the slicing syntax [::-1] to create a new list that includes all elements from the original list in reverse order.

Assign the reversed list to a new variable or overwrite the original list with the reversed version.

```
original_list = [1, 2, 3, 4, 5]

reversed_list = original_list[::-1]
```

## 150. What is the difference between %, /, // ?

In Python, %, /, and // are arithmetic operators with distinct functions:

- The ' % ' is the modulo operator, which returns the remainder of a division. For instance, 5 % 2 would return 1.
- The ' / ' is the division operator that performs floating-point division and returns a float. For example, 5 / 2 would return 2.5.
- The ' // ' is the floor division operator that performs division but rounds down the result to the nearest whole number. So 5 // 2 would return 2.

## 151. What is scope resolution?

In Python, a scope defines the region of code where an object remains valid and accessible. Every object in Python operates within its designated scope. Namespaces are used to uniquely identify objects within a program, and each namespace is associated with a specific scope where objects can be used without any prefix. The scope of a variable determines its accessibility and lifespan.

Let’s explore the different scopes created during code execution:

- Local scope: This refers to the objects that are defined within the current function and are accessible only within that function.
- Global scope: Objects in the global scope are available throughout the execution of the code.
- Module-level scope: This scope encompasses global objects that are associated with the current module in the program. These objects are accessible within the module.
- Outermost scope: This refers to all the built-in names that can be called from anywhere in the program.

## 152. Why doesn't Python deallocate all memory upon exit?

- Whenever Python exits, especially those Python modules, which are having circular references to other objects or the objects that are referenced from the global namespaces, the memory is not always de-allocated or freed.
- It is not possible 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 will try to de-allocate every object.

## 153. Why is a set known as unordered? Is it mutable or immutable?

A set is called “unordered” because the items in a set don’t have a specific order or sequence like a list does. It’s more like a collection of items, and you can’t access them by their position.

Sets in Python are mutable, which means you can add or remove items from a set after it’s created. However, the items within the set (the elements) are themselves immutable, meaning they cannot be changed. You can add or remove elements from a set, but you can’t modify the elements themselves once they’re in the set.

## 154. What is a regular expression, and how do you use it in Python?

The concept of regular expressions emerges with the need to optimize searching algorithms for strings. Match patterns called regular expressions are used to find or replace the matching patterns in strings while performing string operations.

Let’s take a look at a simple example to understand the usage of regular expressions:

```
import re
string = "Intellipaat is a fast growing global Ed-Tech brand"
x = re.search('\s', string) #first white space search
x.start()
```

## 155. What are character classes in regular expressions?

The following are some of the character classes in regular expressions:

1. [abc] – Matching the alphabets i.e a b or c.
2. [a-z] [A-Z] – Matching the alphabets both lowercase and uppercase, in the range a to z.
3. [0-9] – matching the letters in the range specified.
4. [a-zA-Z0-9] – To match any alphanumeric character.
5. [^abc] – Match anything but a b or c.

## 156. How Do You Display the Contents of a Text File in Reverse Order?

This is one of the most asked python basic interview question. You can display the contents of a text file in reverse order using the following steps:

- Open the file using the open() function
- Store the contents of the file into a list
- Reverse the contents of the list
- Run a for loop to iterate through the list

## 157. “in Python, Functions Are First-class Objects.” What Do You Infer from This?

It means that a function can be treated just like an object. You can assign them to variables, or pass them as arguments to other functions. You can even return them from other functions.

## 158. What method will you use to convert a string to all lowercase?

The lower() function can be used to convert a string to lowercase.

## 159. Explain Python packages.

Packages in Python are namespaces that contain numerous modules.

## 160. What are the various types of operators in Python?

- Bitwise operators
- Identity operators
- Membership operators
- Logical operators
- Assignment operators
- Relational operators
- Arithmetic operators

## 161. What does the ‘#’ symbol do in Python?

‘#’ is used to comment on everything that comes after on the line.

## 162. Can we Pass a function as an argument in Python?

Yes, Several arguments can be passed to a function, including objects, variables (of the same or distinct data types), and functions. Functions can be passed as parameters to other functions because they are objects. Higher-order functions are functions that can take other functions as arguments.

## 163. What is a dynamically typed language?

Typed languages are the languages in which we define the type of data type and it will be known by the machine at the compile-time or at runtime. Typed languages can be classified into two categories:

- **Statically typed languages:** In this type of language, the data type of a variable is known at the compile time which means the programmer has to specify the data type of a variable at the time of its declaration.
- **Dynamically typed languages:** 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.

## 164. How do you floor a number in Python?

The Python math module includes a method that can be used to calculate the floor of a number.

- floor() method in Python returns the floor of x i.e., the largest integer not greater than x.
- Also, The method ceil(x) in Python returns a ceiling value of x i.e., the smallest integer greater than or equal to x.

## 165. What Are Some of the Disadvantages of Python?

One of the most common complaints that developers have about Python is poor memory efficiency. Its inefficient use of memory can heavily tax your RAM and make it difficult to run Python-based programs without using significant portions of memory. Python can also run very slowly and, as a result, can suffer from runtime errors. This is largely due to how Python is interpreted and is something that can bother developers who are used to much faster languages like C++ or Java.

Most of Python’s issues are related to its versatility. While it is not optimized for one particular task, it is still a very flexible language. It’s through this flexibility that many use cases for Python have developed over the years, making it virtually indispensable in a developer’s portfolio. The disadvantages of Python make it a less-than-ideal language for developing web applications and for game development due to its poor memory management and slow processing speeds.

## 166. What is the best way to debug a Python program?

This command can be used to debug a Python program.

```
Python -m pdb Python-script.py
```

## 167. How can I make a tuple out of a list?

We can transform a list into a tuple using the Python tuple() method. Since a tuple is immutable, we can't update the list after it has been converted to a tuple.

**Code**

```
month = ['jan','feb','mar','apr','may','jun','jul','aug','sep','oct','nov','dec']
converting_list = tuple(month)
print(converting_list)
print(type(converting_list))
```

**Output:**

```
('jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec')
<class 'tuple'>
```

## 168. In Python, how do you create random numbers?

We can create random data in Python utilizing several functions. They are as follows:

- random() - This instruction gives a floating-point value ranging from 0 to 1.
- uniform(X, Y) - This function gives a floating-point value in the X and Y range.
- randint(X, Y) - This function gives a random integer between X and Y values.

## 169. How do you print the summation of all the numbers from 1 to 101?

Using this program, we can display the summation of all numbers from 1 to 101:

**Code**

```
Print( sum(range(1,102)) )
```

**Output:**

```
5151
```

## 170. What are the advantages of Python?

The advantages of utilizing Python are as follows:

- **Simple to understand and utilize-** Python is a powerful language of programming that is simple to learn, read, and write.
- **Interpreted language-** Python is an interpreted language, which means it runs the program line by line & pauses if any line contains an error.
- **Dynamically typed-** when coding, the programmer does not set data types to variables. During execution, it is automatically assigned.
- Python is free and open-source to use and share. It's free and open source.
- **Extensive library support-** Python has a large library of functions that can perform practically any task. It also allows you to use Python Package Manager to import additional packages (pip).
- Python applications are portable and can execute on any system without modification.
- Python's data structures are easy to understand.
- It allows for additional functionality while requiring less coding.

## 171. What are the applications of Python?

The applications of Python are as follows:

- GUI-based desktop applications
- Image processing applications
- Business and Enterprise applications
- Prototyping
- Web and web framework applications

## 172. How does a function return values?

Functions return values using the return statement.

## 173. What happens when a function doesn’t have a return statement? Is this valid?

Yes, this is valid. The function will then return a None object. The end of a function is defined by the block of code that is executed (i.e., the indenting) not by any explicit keyword.

## 174. What is a boolean in Python?

Boolean is one of the built-in data types in Python, it mainly contains two values, which are true and false.

Python bool() is the method used to convert a value to a boolean value.

```
Syntax for bool() method: bool([a])
```

## 175. What are Python String formats and Python String replacements?

Python String Format: The String format() method in Python is mainly used to format the given string into an accurate output or result.

**Syntax for String format() method:**

```
template.format(p0, p1, ..., k0=v0, k1=v1, ...)
```

Python String Replace: This method is mainly used to return a copy of the string in which all the occurrence of the substring is replaced by another substring.

**Syntax for String replace() method:**

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

## 176. When would you use triple quotes as a delimiter?

Triple quotes ‘” or ‘“ are string delimiters that can span multiple lines in Python. Triple quotes are usually used when spanning multiple lines, or enclosing a string that has a mix of single and double quotes contained therein.

## 177. What is Try Block?

A block that is preceded by the try keyword is known as a try block

**Syntax:**

```
try{
  //statements that may cause an exception
}
```

## 178. Define what is “Method” in Python programming?

The Method is defined as the function associated with a particular object. The method which we define should not be unique as a class instance. Any type of object can have methods.

## 179. Define Constructor in Python?

Constructor is a special type of method with a block of code to initialize the state of instance members of the class. A constructor is called only when the instance of the object is created. It is also used to verify that they are sufficient resources for objects to perform a specific task.

There are two types of constructors in Python, and they are:

- Parameterized constructor
- Non-parameterized constructor

## 180. What are the cool things you can do with Python?

The following are some of the things that you can perform using Python:

- Automate tasks
- Play games
- Build a Blockchain to mine Bitcoins
- Build a chatbot interface combined with AI

## 181. Why should you choose Python?

There is no doubt, Python is the best choice for coding in the interview. Other than Python, if you prefer to choose C++ or java you need to worry about structure and syntax.

## 182. What type of language is Python?

Python is an interpreted, interactive, object-oriented programming language. Classes, modules, exceptions, dynamic typing, and extremely high-level dynamic data types are all present.

## 183. What is recursion?

Recursion is a function calling itself one or more times in it body. One very important condition a recursive function should have to be used in a program is, it should terminate, else there would be a problem of an infinite loop.

## 184. What is string interpolation in Python?

String interpolation in Python allows you to embed expressions or variables within a string, making it easier to construct dynamic strings. It can be done using f-strings or the format() method.

## 185. What are Python conditional statements?

Python conditional statements, such as if, elif, and else, allow you to perform different actions based on certain conditions. They control the flow of the program based on the truthfulness of the condition.

## 186. What is the difference between a function and a method in Python?

In Python, a function is a standalone block of code that can be called independently. A method, on the other hand, is a function that is associated with an object or a class and can access the object's data.

## 187. What is the purpose of the @property decorator in Python?

The @property decorator in Python is used to define a method as a getter for a class attribute. It allows accessing the attribute as if it were a normal attribute, while internally calling the getter method.

## 188. What is the purpose of the @staticmethod decorator in Python?

The @staticmethod decorator in Python is used to define a static method in a class. Static methods do not require an instance of the class to be called and can be accessed directly from the class itself.

## 189. What is the purpose of the @classmethod decorator in Python?

The @classmethod decorator in Python is used to define a class method. Class methods receive the class itself as the first parameter, allowing them to access and modify class- level attributes and perform operations specific to the class.

## 190. What are the advantages of using Python for web development?

Python offers several advantages for web development, including a wide range of frameworks (such as Django and Flask), a large community, extensive libraries, and easy integration with other technologies.

## 191.What is the purpose of the sys module in Python?

The sys module in Python provides access to system-specific parameters and functions. It allows interaction with the Python interpreter and provides information about the runtime environment.

## 192. What is the purpose of the os module in Python?

The os module in Python provides a way to interact with the operating system. It allows performing various operations related to file and directory manipulation, process management, and environment variables.

## 193. What is the purpose of the datetime module in Python?

The datetime module in Python provides classes for manipulating dates and times. It allows creating, formatting, and performing operations on dates and times.

### 194. What is the purpose of the random module in Python?

The random module in Python provides functions for generating random numbers. It allows you to generate random integers, floating-point numbers, and make random selections from lists.

### 195. What is the purpose of the json module in Python?

The json module in Python provides functions for working with JSON (JavaScript Object Notation) data. It allows encoding Python objects into JSON strings and decoding JSON strings into Python object.

### 196. What is the Global Interpreter Lock (GIL) in Python?

The Global Interpreter Lock (GIL) is a mechanism in the CPython interpreter (the reference implementation of Python) that allows only one thread to execute Python bytecode at a time. This restricts the parallel execution of Python threads and can impact performance in certain scenarios.

### 197. How do you open and close a file in Python?

Files can be opened in Python using the open() function, which takes the file name and the mode of operation as arguments. The close() method is used to close an opened file and free up system resources.

### 198. What is the purpose of the _main_ block in Python?

The _main_ block in Python is used to define the entry point of a Python program. The code inside the if _name_ == "_main_": block will only execute if the script is run directly, not when it is imported as a module.

## 199. What is the purpose of the _str_ method in Python?

The _str_ method in Python is a special method that returns a string representation of an object. It is used to provide a human-readable representation of the object when the str() function is called or when the object is printed.

## 200. What is the purpose of the _repr_ method in Python?

The _repr_ method in Python is a special method that returns a string representation of an object that can be used to recreate the object. It is used to provide a detailed and unambiguous representation of the object.