**1. What is Python? List some popular applications of Python in the world of technology.**

Python is a widely-used general-purpose, high-level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.  
It is used for:

* System Scripting
* Web Development
* Game Development
* Software Development
* Complex Mathematics

**2.What are the benefits of using Python language as a tool in the present scenario?**

The following are the benefits of using Python language:

* Object-Oriented Language
* High-Level Language
* Dynamically Typed language
* Extensive support Libraries
* Presence of third-party modules
* Open source and community development
* Portable and Interactive
* Portable across Operating systems

**3. Is Python a compiled language or an interpreted language?**

Actually, Python is a partially compiled language and partially interpreted language. The compilation part is done first when we execute our code and this will generate byte code internally this byte code gets converted by the Python virtual machine(p.v.m) according to the underlying platform(machine+operating system).

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

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

**5. What is the difference between a Mutable datatype and an Immutable data type?**

Mutable data types can be edited i.e., they can change at runtime. Eg – List, Dictionary, etc.Immutable data types can not be edited i.e., they can not change at runtime. Eg – String, Tuple, etc.

**6. How are arguments passed by value or by reference in Python?**

Everything in Python is an object and all variables hold references to the objects. The reference values are according to the functions; as a result, you cannot change the value of the references. However, you can change the objects if it is mutable.

**7. What is the difference between a Set and Dictionary?**

The set is an unordered collection of data types that is iterable, mutable and has no duplicate elements.  
A dictionary in Python is an ordered collection of data values, used to store data values like a map.

**8. What is List Comprehension? Give an Example.**

List comprehension is a syntax construction to ease the creation of a list based on existing iterable.

For Example:

my\_list = [i for i in range(1, 10)]

**9. What is a lambda function?**

A lambda function is an anonymous function. This function can have any number of parameters but, can have just one statement. For Example:

a = lambda x, y : x\*y

print(a(7, 19))

**10. What is a pass in Python?**

Pass means performing no operation or in other words, it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.

**11. What is the difference between / and // in Python?**

/ represents precise division (result is a floating point number) whereas // represents floor division (result is an integer). For Example:

5//2 = 2

5/2 = 2.5

**12. How is Exceptional handling done in Python?**

There are 3 main keywords i.e. try, except, and finally which are used to catch exceptions and handle the recovering mechanism accordingly. Try is the block of a code that is monitored for errors. Except block gets executed when an error occurs.

The beauty of the final block is to execute the code after trying for an error. This block gets executed irrespective of whether an error occurred or not. Finally, block is used to do the required cleanup activities of objects/variables.

**13. What is swapcase function in Python?**

It is a string’s function that converts all uppercase characters into lowercase and vice versa. It is used to alter the existing case of the string. This method creates a copy of the string which contains all the characters in the swap case. For Example:

string = "GeeksforGeeks"

string.swapcase() ---> "gEEKSFORgEEKS"

**14. Difference between for loop and while loop in Python**

The “for” Loop is generally used to iterate through the elements of various collection types such as [List](https://www.geeksforgeeks.org/python-lists/), [Tuple](https://www.geeksforgeeks.org/python-tuples/), [Set](https://www.geeksforgeeks.org/sets-in-python/), and [Dictionary](https://www.geeksforgeeks.org/python-dictionary/). Developers use a “for” loop where they have both the conditions start and the end. Whereas, the “while” loop is the actual looping feature that is used in any other programming language. Programmers use a Python while loop where they just have the end conditions.

**15. 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.

To read more, refer to the article: [Passing function as an argument in Python](https://www.geeksforgeeks.org/passing-function-as-an-argument-in-python/)

**16. What are \*args and \*kwargs?**

To pass a variable number of arguments to a function in Python, use the special syntax [\*args and \*\*kwargs](https://www.geeksforgeeks.org/args-kwargs-python/) in the function specification. It is used to pass a variable-length, keyword-free argument list. By using the \*, the variable we associate with the \* becomes iterable, allowing you to do operations on it such as iterating over it and using higher-order operations like map and filter.

**17. Is Indentation Required in Python?**

Yes, [indentation](https://www.geeksforgeeks.org/indentation-in-python/) is required in Python. A [Python](https://www.geeksforgeeks.org/python-programming-language/) interpreter can be informed that a group of statements belongs to a specific block of code by using Python indentation. Indentations make the code easy to read for developers in all programming languages but in Python, it is very important to indent the code in a specific order.

**18. What is Scope in Python?**

The location where we can find a variable and also access it if required is called the scope of a variable.

* **Python Local variable:** Local variables are those that are initialized within a function and are unique to that function. It cannot be accessed outside of the function.
* **Python Global variables:** Global variables are the ones that are defined and declared outside any function and are not specified to any function.
* **Module-level scope:** It refers to the global objects of the current module accessible in the program.
* **Outermost scope:** It refers to any built-in names that the program can call. The name referenced is located last among the objects in this scope.

**19. What is docstring in Python?**

Python documentation strings (or docstrings) provide a convenient way of associating documentation with Python modules, functions, classes, and methods.

* **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.
* **Accessing Docstrings:** The docstrings can be accessed using the \_\_doc\_\_ method of the object or using the help function.

**20. What is a dynamically typed language?**

[Typed languages](https://www.geeksforgeeks.org/what-is-a-typed-language/) 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.

**21. What is a break, continue, and pass in Python?**

The [break statement](https://www.geeksforgeeks.org/python-break-statement/) 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.

[Continue](https://www.geeksforgeeks.org/python-continue-statement/) is also a loop control statement just like the break statement. continue statement is opposite to that of the break statement, instead of terminating the loop, it forces to execute the next iteration of the loop.

[Pass](https://www.geeksforgeeks.org/python-pass-statement/) means performing no operation or in other words, it is a placeholder in the compound statement, where there should be a blank left and nothing has to be written there.

**22. What are Built-in data types in Python?**

The following are the standard or built-in data types in Python:

* **Numeric:** The numeric data type in Python represents the data that has a numeric value. A numeric value can be an integer, a floating number, a Boolean**,** or even a complex number.
* **Sequence Type: T**he sequence Data Type in Python is the ordered collection of similar or different data types. There are several sequence types in Python:
  + [Python String](https://www.geeksforgeeks.org/python-string/)
  + [Python List](https://www.geeksforgeeks.org/python-lists/)
  + [Python Tuple](https://www.geeksforgeeks.org/python-tuples/)
  + [Python range](https://www.geeksforgeeks.org/python-range-function/)
* **Mapping Types:** In Python, hashable data can be mapped to random objects using a mapping object. There is currently only one common mapping type, the dictionary, and mapping objects are mutable.
  + [Python Dictionary](https://www.geeksforgeeks.org/python-dictionary/)
* **Set Types:** In Python, a [Set](https://www.geeksforgeeks.org/sets-in-python/) is an unordered collection of data types that is iterable, mutable, and has no duplicate elements. The order of elements in a set is undefined though it may consist of various elements.

**23. 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()](https://www.geeksforgeeks.org/floor-ceil-function-python/) 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.

**24. What is the difference between xrange and range functions?**

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.

* *range()* – This returns a list of numbers created using the range() function.
* *xrange()* – This function returns the generator object that can be used to display numbers only by looping. The only particular range is displayed on demand and hence called *lazy evaluation*.

**25. What is Dictionary Comprehension? Give an Example**

Dictionary Comprehension is a syntax construction to ease the creation of a dictionary based on the existing iterable.

For Example: *my\_dict = {i:i+7 for i in range(1, 10)}*

**26. Is Tuple Comprehension? If yes, how, and if not why?**

(i for i in (1, 2, 3))

Tuple comprehension is not possible in Python because it will end up in a generator, not a tuple comprehension.

**27. Differentiate between List and Tuple?**

Let’s analyze the differences between List and Tuple:

**List**

* Lists are Mutable datatype.
* Lists consume more memory
* The list is better for performing operations, such as insertion and deletion.
* The implication of iterations is Time-consuming

**Tuple**

* Tuples are Immutable datatype.
* Tuple consumes less memory as compared to the list
* A Tuple data type is appropriate for accessing the elements
* The implication of iterations is comparatively Faster

**28. What is the difference between a shallow copy and a deep copy?**

Shallow copy is used when a new instance type gets created and it keeps values that are copied whereas deep copy stores values that are already copied.

A shallow copy has faster program execution whereas a deep copy makes it slow.

**29. Which sorting technique is used by sort() and sorted() functions of python?**

Python uses the [**Tim Sort**](https://www.geeksforgeeks.org/timsort/) algorithm for sorting. It’s a stable sorting whose worst case is O(N log N). It’s a hybrid sorting algorithm, derived from merge sort and insertion sort, designed to perform well on many kinds of real-world data.

**30. What are Decorators?**

Decorators are a very powerful and useful tool in Python as they are the specific change that we make in Python syntax to alter functions easily.

**31. How do you debug a Python program?**

By using this command we can debug a Python program:

$ python -m pdb python-script.py

**32. What are Iterators in Python?**

In Python, iterators are used to iterate a group of elements, containers like a list. Iterators are collections of items, and they can be a list, tuples, or a dictionary. Python iterator implements \_\_itr\_\_ and the next() method to iterate the stored elements. We generally use loops to iterate over the collections (list, tuple) in Python.

**33. What are Generators in Python?**

In Python, the generator is a way that specifies how to implement iterators. It is a normal function except that it yields expression in the function. It does not implement \_\_itr\_\_ and next() method and reduces other overheads as well.

If a function contains at least a yield statement, it becomes a generator. The yield keyword pauses the current execution by saving its states and then resumes from the same when required.

**34. Does Python supports multiple Inheritance?**

Python does support multiple inheritances, unlike Java. Multiple inheritances mean that a class can be derived from more than one parent class.

**35. What is Polymorphism in Python?**

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.

**36. Define encapsulation in Python?**

Encapsulation means binding the code and the data together. A Python class is an example of encapsulation.

**37. How do you do data abstraction in Python?**

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

**38. How is memory management done in Python?**

Python uses its private heap space to manage the memory. Basically, all the objects and data structures are stored in the private heap space. Even the programmer can not access this private space as the interpreter takes care of this space. Python also has an inbuilt garbage collector, which recycles all the unused memory and frees the memory and makes it available to the heap space.

**39. How to delete a file using Python?**

We can delete a file using Python by following approaches:

* os.remove()
* os.unlink()

**40. What is slicing in Python?**

[Python Slicing](https://www.geeksforgeeks.org/python-slice-function/) is a string operation for extracting a part of the string, or some part of a list. 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.

Syntax: Lst[ Initial : End : IndexJump ]

**41. What is a namespace in Python?**

A namespace is a naming system used to make sure that names are unique to avoid naming conflicts.

**42. What is PIP?**

PIP is an acronym for Python Installer Package which provides a seamless interface to install various Python modules. It is a command-line tool that can search for packages over the internet and install them without any user interaction.

**43. What is a zip function?**

Python zip() function returns a zip object, which maps a similar index of multiple containers. It takes an iterable, converts it into an iterator and aggregates the elements based on iterables passed. It returns an iterator of tuples.

**44. What are Pickling and Unpickling?**

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

**45. What is monkey patching in Python?**

In Python, the term monkey patch only refers to dynamic modifications of a class or module at run-time.

# g.py

class GeeksClass:

def function(self):

print "function()"

import m

def monkey\_function(self):

print "monkey\_function()"

m.GeeksClass.function = monkey\_function

obj = m.GeeksClass()

obj.function()

**46. What is \_\_init\_\_() in Python?**

Equivalent to constructors in OOP terminology, \_\_init\_\_ is a reserved method in Python classes. The \_\_init\_\_ method is called automatically whenever a new object is initiated. This method allocates memory to the new object as soon as it is created. This method can also be used to initialize variables.

**47. Write a code to display the current time?**

import time

currenttime= time.localtime(time.time())

print (“Current time is”, currenttime)

**48. What are Access Specifiers in Python?**

Python uses the ‘\_’ symbol to determine the access control for a specific data member or a member function of a class. A Class in Python has three types of [Python access modifiers](https://www.geeksforgeeks.org/access-modifiers-in-python-public-private-and-protected/):

* **Public Access Modifier:** 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.
* **Protected Access Modifier:** The members of a class that are declared protected are only accessible to a class derived from it. All data members of a class are declared protected by adding a single underscore ‘\_’ symbol before the data members of that class.
* **Private Access Modifier:** The members of a class that are declared private are accessible within the class only, the 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.

**49. What are unit tests in Python?**

Unit Testing is the first level of software testing where the smallest testable parts of the software are tested. This is used to validate that each unit of the software performs as designed. The unit test framework is Python’s xUnit style framework. The White Box Testing method is used for Unit testing.

**50. Python Global Interpreter Lock (GIL)?**

[Python Global Interpreter Lock](https://www.geeksforgeeks.org/what-is-the-python-global-interpreter-lock-gil/) (GIL) is a type of process lock that is used by Python whenever it deals with processes. Generally, Python only uses only one thread to execute the set of written statements. The performance of the single-threaded process and the multi-threaded process will be the same in Python and this is because of GIL in Python. We can not achieve multithreading in Python because we have a global interpreter lock that restricts the threads and works as a single thread.

**51. What are Function Annotations in Python?**

[Function Annotation](https://www.geeksforgeeks.org/function-annotations-python/) is a feature that allows you to add metadata to function parameters and return values. This way you can specify the input type of the function parameters and the return type of the value the function returns.

Function annotations are arbitrary Python expressions that are associated with various parts of functions. These expressions are evaluated at compile time and have no life in Python’s runtime environment. Python does not attach any meaning to these annotations. They take life when interpreted by third-party libraries, for example, mypy.

**52. What are Exception Groups in Python?**

The latest feature of Python 3.11, [Exception Groups](https://www.geeksforgeeks.org/exception-groups-in-python/). The ExceptionGroup can be handled using a new except\* syntax. The \* symbol indicates that multiple exceptions can be handled by each except\* clause.

ExceptionGroup is a collection/group of different kinds of Exception. Without creating Multiple Exceptions we can group together different Exceptions which we can later fetch one by one whenever necessary, the order in which the Exceptions are stored in the Exception Group doesn’t matter while calling them.

Python

**try**:

**raise** ExceptionGroup('Example ExceptionGroup', (

**TypeError**('Example TypeError'),

**ValueError**('Example ValueError'),

**KeyError**('Example KeyError'),

**AttributeError**('Example AttributeError')

))

**except**\* **TypeError**:

...

**except**\* **ValueError** **as** e:

...

**except**\* (**KeyError**, **AttributeError**) **as** e:

…

**53. What is Python Switch Statement**

From version 3.10 upward, Python has implemented a switch case feature called “structural pattern matching”. You can implement this feature with the match and case keywords. Note that the underscore symbol is what you use to define a default case for the switch statement in Python.

**Note**: Before Python 3.10 Python doesn’t support match Statements.

Python

**match** term:

**case** pattern-1:

action-1

**case** pattern-2:

action-2

**case** pattern-3:

action-3

**case** **\_**:

Action-default

**54. What is Walrus Operator?**

[The Walrus Operator](https://docs.python.org/3/whatsnew/3.8.html) allows you to assign a value to a variable within an expression. This can be useful when you need to use a value multiple times in a loop, but don’t want to repeat the calculation.

The Walrus Operator is represented by the `:=` syntax and can be used in a variety of contexts including while loops and if statements.

**Note:** Python versions before 3.8 doesn’t support Walrus Operator.

Python

names = ["Jacob", "Joe", "Jim"]

**if** (name := input("Enter a name: ")) **in** names:

print(f"Hello, **{**name**}**!")

**else**:

print("Name not found.")

**55.What is the difference between == and is in Python?**

* + == checks for value equality, whereas is checks for reference equality (i.e., whether the two variables point to the same object).

**56.How do you convert a string to a number in Python?**

* + Use int() for integers and float() for floating-point numbers. Example: int("10") or float("10.5").

**57.What is the purpose of the with statement in Python?**

* + It is used to wrap the execution of a block of code within methods defined by a context manager, ensuring proper acquisition and release of resources.

**58.How do you create a virtual environment in Python?**

* + Use python -m venv env\_name to create a virtual environment.

**59.What is the super() function in Python?**

* + super() is used to call a method from the parent class in a subclass.

**60.How do you handle missing keys in a dictionary?**

* + Use the get() method or collections.defaultdict.

**61.What is a list in Python?**

* + A list is a mutable, ordered sequence of elements. Example: [1, 2, 3].

**61.What is a tuple in Python?**

* + A tuple is an immutable, ordered sequence of elements. Example: (1, 2, 3).

**62.What is a set in Python?**

* + A set is an unordered collection of unique elements. Example: {1, 2, 3}.

**63.What is the difference between remove() and pop() in Python?**

* + remove() deletes the first occurrence of a value, while pop() removes an element at a specified index or the last element if the index is not provided.

**64.What are \*args and \*\*kwargs used for?**

* + \*args allows a function to accept any number of positional arguments, and \*\*kwargs allows it to accept any number of keyword arguments.

**65.How can you copy an object in Python?**

* + Use the copy module with copy.copy() for a shallow copy and copy.deepcopy() for a deep copy.

**66.What are list comprehensions in Python?**

* + A concise way to create lists. Example: [x for x in range(5)].

**67.What is a generator in Python?**

* + A function that returns an iterator and uses yield instead of return.

**68.How do you merge two dictionaries in Python?**

* + Use the {\*\*dict1, \*\*dict2} syntax or dict1.update(dict2).

**69.What is a lambda function in Python?**

* + An anonymous function defined using the lambda keyword. Example: lambda x: x + 1.

**70.What is the @staticmethod decorator in Python?**

* + It defines a method that doesn't operate on an instance or class but still belongs to the class's namespace.

**71.What is the difference between \_\_str\_\_ and \_\_repr\_\_?**

* + \_\_str\_\_ is meant for end-user output, while \_\_repr\_\_ is meant for developer/debugging output.

**72.What is the difference between sort() and sorted()?**

* + sort() sorts a list in place, whereas sorted() returns a new sorted list.

**73.How can you handle exceptions in Python?**

* + Use try, except, else, and finally blocks.

**74.What is a metaclass in Python?**

* + A class of a class that defines how a class behaves.

**75.What is the Global Interpreter Lock (GIL)?**

* + A mutex that protects access to Python objects, preventing multiple threads from executing Python bytecodes simultaneously.

**76.How do you profile a Python script?**

* + Use the cProfile or profile modules.

**77.What is the @classmethod decorator in Python?**

* + It defines a method that operates on the class rather than instances of the class.

**78.How do you create a singleton class in Python?**

* + By overriding the \_\_new\_\_ method or using a metaclass.

**79.What is monkey patching in Python?**

* + The practice of dynamically modifying or extending a module or class at runtime.

**80.What is the inspect module used for?**

* + For introspection, to get information about live objects such as modules, classes, and functions.

**81.How can you optimize Python code for performance?**

* + Use built-in functions, list comprehensions, and modules like NumPy, and avoid unnecessary loops and global variables.

**82.What is the difference between \_\_new\_\_ and \_\_init\_\_?**

* + \_\_new\_\_ creates a new instance, while \_\_init\_\_ initializes the instance.

**83.How do you handle circular imports in Python?**

* + By using local imports, restructuring the code, or using import guards.

**84.What is a context manager in Python?**

* + A way to allocate and release resources precisely when you want to. Implemented using with statement and \_\_enter\_\_ and \_\_exit\_\_ methods.

**85.What are coroutines in Python?**

* + Functions that can pause and resume their execution, used for asynchronous programming with async and await keywords.

**86.What is memoization in Python?**

* + An optimization technique to cache the results of expensive function calls to avoid redundant computations.

**87.How do you create a module in Python?**

* + By writing Python code in a .py file and importing it using the import statement.

**88.What is the difference between staticmethod and classmethod?**

* + staticmethod doesn't take any parameters related to the class or instance, while classmethod takes the class itself as the first parameter.

**89.What is the \_\_slots\_\_ attribute in Python?**

* + It is used to declare a fixed set of attributes, optimizing memory usage by preventing the creation of \_\_dict\_\_ for each instance.

**90.What is the purpose of the heapq module?**

* + It provides an implementation of the heap queue algorithm, also known as the priority queue algorithm.

**91.How do you create a thread-safe program in Python?**

* + By using threading locks, semaphores, or other synchronization primitives provided by the threading module.

**92.What is the uuid module used for?**

* + To generate universally unique identifiers (UUIDs).

**93.How do you serialize and deserialize objects in Python?**

* + By using the pickle module for binary serialization or json module for text serialization.

**94.What are decorators in Python?**

* + Functions that modify the behavior of other functions or methods. Defined using the @decorator syntax.

**95.What is the \_\_name\_\_ variable used for?**

* + It determines if the module is being run directly or imported elsewhere. If \_\_name\_\_ == '\_\_main\_\_', the module is being run directly.

**96.How do you handle command-line arguments in Python?**

* + By using the argparse module.

**97.What is the enum module used for?**

* + To create enumerations, which are a set of symbolic names bound to unique, constant integer values.

**98.What is the purpose of the assert statement?**

* + To test if a condition in your code returns True. If not, the program will raise an AssertionError.

**99.What is the difference between synchronous and asynchronous programming?**

* + Synchronous programming waits for each operation to complete before moving on, while asynchronous programming allows the execution of other operations while waiting for the completion of the current operation.

**100.How do you handle multiple exceptions in a single block?**

* + By using multiple except clauses or combining exceptions in a tuple. Example: except (TypeError, ValueError):.

**101.What is the itertools module used for?**

* + To provide functions that create iterators for efficient looping.

**102.How do you perform matrix operations in Python?**

* + By using libraries like NumPy.

**103.What is the abc module used for?**

* + To define Abstract Base Classes (ABCs).

**104.How do you create custom exceptions in Python?**

* + By defining a new class that inherits from the built-in Exception class.

**105.What is the logging module used for?**

* + For tracking events that happen when some software runs, typically used for debugging and monitoring.

**106.What is the difference between isinstance() and issubclass()?**

* + isinstance() checks if an object is an instance of a class or a tuple of classes, whereas issubclass() checks if a class is a subclass of another class.

**107.How do you implement a binary search algorithm in Python?**

* + By writing a function that repeatedly divides the search interval in half.

**108.What is the shutil module used for?**

* + To perform high-level file operations like copying and removing files.

**109.What is the purpose of the \_\_main\_\_.py file in a Python package?**

* + To define the entry point of a package when it is executed as a script.

**110.How do you create an iterator in Python?**

* + By defining a class with \_\_iter\_\_() and \_\_next\_\_() methods.

**111.What is a mixin in Python?**

* + A class that provides methods to other classes through inheritance but is not meant to stand on its own.

**112.What is the pathlib module used for?**

* + To handle filesystem paths in an object-oriented way.

**113.How do you schedule tasks in Python?**

* + By using libraries like schedule or APScheduler.

**114.What is the weakref module used for?**

* + To create weak references to objects, allowing them to be garbage-collected.

**115.How do you convert a list of tuples into a dictionary?**

* + By using the dict() constructor. Example: dict([('a', 1), ('b', 2)]).

**116.What is the collections module used for?**

* + To provide alternatives to Python's general-purpose built-in containers like dict, list, set, and tuple.

**117.What is the purpose of the type() function?**

* + To return the type of an object or create a new type.

**118.How do you merge two sets in Python?**

* + By using the union() method or the | operator.

**119.What is the difference between @staticmethod and @classmethod?**

* + @staticmethod defines a static method that doesn't receive the class or instance as the first argument, while @classmethod receives the class as the first argument.

**120.What is the os module used for?**

* + To interact with the operating system, such as reading or writing to the filesystem.

**121.How do you reverse a string in Python?**

* + By using slicing: s[::-1].

**122.What is the difference between None and 0 in Python?**

* + None is a special constant representing the absence of a value, while 0 is an integer value.

**123.What is the collections module used for?**

* + It provides specialized container datatypes like namedtuple, deque, Counter, OrderedDict, defaultdict, and ChainMap.

**124.What is the difference between classmethod() and staticmethod()?**

* + classmethod() can modify class state that applies across all instances of the class, while staticmethod() cannot access or modify class state.

**125.What are f-strings in Python?**

* + A way to embed expressions inside string literals, using curly braces {}. Example: f"Hello, {name}".

**126.How do you handle large datasets in Python?**

* + By using libraries like pandas for data manipulation, numpy for numerical operations, and dask for parallel computing.

**127.What is the re module used for?**

* + For working with regular expressions.

**128.How do you reverse a string in Python?**

* + Using slicing: my\_string[::-1].

**129.What are the different types of arguments in Python?**

* + Positional arguments, keyword arguments, default arguments, variable-length arguments (\*args and \*\*kwargs).

**130.What is the subprocess module used for?**

* + To spawn new processes, connect to their input/output/error pipes, and obtain their return codes.

**131.What is the multiprocessing module used for?**

* + For running parallel processes, allowing the efficient use of multiple CPUs.

**132.What is the contextlib module used for?**

* + To provide utilities for working with context managers and the with statement.

**133.How do you sort a list of dictionaries by a specific key?**

* + Using the sorted() function with a lambda function as the key. Example: sorted(list\_of\_dicts, key=lambda x: x['key']).

**134.What is a descriptor in Python?**

* + An object attribute with binding behavior, whose access has been overridden by methods in the descriptor protocol (\_\_get\_\_, \_\_set\_\_, \_\_delete\_\_).

**135.How do you handle circular references in Python?**

* + By using weak references from the weakref module or by careful design to avoid circular dependencies.

**136.What is the difference between @abstractmethod and @staticmethod?**

* + @abstractmethod is used to declare abstract methods in abstract base classes, requiring subclasses to implement them, while @staticmethod is used to define static methods that do not operate on class or instance data.

**137.What is the bisect module used for?**

* + To maintain a list in sorted order without having to sort it after each insertion, using functions like bisect() and insort().

**138.What is the dataclasses module used for?**

* To provide a decorator and functions for automatically adding special methods to user-defined classes, making it easier to create classes that store data.

**139.How do you create a custom iterator in Python?**

* + By defining a class with \_\_iter\_\_() and \_\_next\_\_() methods.

**140.What is the functools module used for?**

* + To provide higher-order functions that act on or return other functions, such as reduce(), partial(), lru\_cache().

**141.How do you check for memory leaks in a Python application?**

* + By using profiling tools like objgraph, pympler, or tracemalloc.

**142.What is duck typing in Python?**

* + A programming style that determines an object's suitability for use by the presence of certain methods and properties, rather than the object's type itself.

**143.What are property decorators in Python?**

* + @property is used to create managed attributes, turning class methods into properties.

**144.How do you merge multiple dictionaries in Python 3.9+?**

* + Using the union operator (|). Example: dict3 = dict1 | dict2.

**145.What is the difference between remove(), pop(), and del?**

* + remove() deletes the first occurrence of a value, pop() removes and returns an element at a given index, and del deletes an element or a slice from a list by index.

**146.How do you implement method overloading in Python?**

* + Python does not support method overloading directly. However, you can achieve similar functionality using default arguments, \*args and \*\*kwargs.

**147.What is the struct module used for?**

* + To convert between Python values and C structs represented as Python bytes objects.

**148.How do you handle database connections in Python?**

* + By using libraries like sqlite3, SQLAlchemy, psycopg2 (for PostgreSQL), or PyMySQL (for MySQL).

**149.What is the \_\_call\_\_ method in Python?**

* + It allows an instance of a class to be called as a function.

**150.What are slots in Python?**

* + \_\_slots\_\_ is used to limit the attributes of an instance, saving memory by preventing the creation of \_\_dict\_\_.

**151.What is the pygments module used for?**

* + For syntax highlighting.

**152.What is the difference between extend() and append() in a list?**

* + extend() adds elements from an iterable to the end of the list, whereas append() adds its argument as a single element to the end of the list.

**153.How do you find the intersection of two lists in Python?**

* + Using list comprehensions or the set intersection method. Example: list(set(list1) & set(list2)).

**154.What is the difference between \_\_getattr\_\_ and \_\_getattribute\_\_?**

* + \_\_getattr\_\_ is called only when an attribute is not found, whereas \_\_getattribute\_\_ is called for every attribute access.

**155.What is the traceback module used for?**

* + To extract, format, and print stack traces of a Python program.

**156.How do you create a read-only attribute in Python?**

* + By using the property decorator with only a getter method.

**157.What is the difference between flush() and close() in file handling?**

* + flush() writes the buffer's contents to disk without closing the file, while close() flushes and then closes the file, making it unusable.

**158.How do you handle file paths in a platform-independent way?**

* + By using the os.path module or pathlib module.

**159.What is the mmap module used for?**

* + To create memory-mapped file objects, allowing files to be manipulated as if they were arrays of memory.

**160.How do you unit test a Python application?**

* + By using the unittest module or other testing frameworks like pytest or nose.

**161.What is the difference between repr() and str()?**

* + repr() returns a string that would produce the same object when passed to eval(), while str() returns a readable, human-friendly string.

**162.What is the yield from expression used for?**

* + To delegate part of a generator's operations to another generator.

**163.What is the sys module used for?**

* + To provide access to some variables used or maintained by the Python interpreter and to functions that interact with the interpreter.

**164.How do you create an immutable class in Python?**

* + By defining \_\_setattr\_\_ and \_\_delattr\_\_ methods to raise exceptions or using namedtuple from the collections module.

**165.What is the atexit module used for?**

* + To register functions to be executed upon normal program termination.

**166.What is the unittest.mock module used for?** - For testing by replacing parts of the system under test with mock objects and making assertions about how they were used.

**167.What is the shutil module used for?** - To perform high-level file operations like copying, moving, and removing files and directories.

**168.How do you create a daemon thread in Python?** - By setting the daemon attribute to True before starting the thread. Example: thread.daemon = True.

**169.What is the purpose of the \_\_main\_\_ block?** - It allows code to be executed when the script is run directly, but not when it is imported as a module. Example:

python

Copy code

if \_\_name\_\_ == "\_\_main\_\_":

main()

**170.What is the argparse module used for?** - To parse command-line arguments.

**171.What is the textwrap module used for?** - To format and wrap plain text.

**172.How do you perform HTTP requests in Python?** - By using the requests library. Example:

python

Copy code

import requests

response = requests.get('https://example.com')

**173.What is the difference between Thread and Process in Python?** - Thread is for concurrent execution within the same process, while Process is for parallel execution in separate processes.

**174.How do you achieve reflection in Python?** - By using the getattr(), setattr(), and hasattr() functions to inspect and modify object attributes at runtime.

**175.What is the abc module used for?** - To define Abstract Base Classes, which can contain abstract methods that must be implemented by subclasses.

**176.How do you create a read-only property in Python?** - By using the property decorator with only a getter method.

**177.What is the difference between isinstance() and type()?** - isinstance() checks for inheritance and class, whereas type() checks for the exact type of an object.

**178.What is the purpose of the with statement in file handling?** - It ensures that resources are properly acquired and released, handling exceptions and cleaning up after the block is executed.

**179.How do you convert a list of tuples to a dictionary?** - By using the dict() constructor. Example:

python

Copy code

dict\_of\_tuples = dict([('a', 1), ('b', 2)])

**180.What is the heapq module used for?** - To implement a heap queue (priority queue) algorithm.

**181.What is the difference between deepcopy() and copy()?** - deepcopy() creates a new object and recursively adds copies of nested objects, whereas copy() creates a shallow copy with references to the same objects.

**182.How do you handle JSON data in Python?** - By using the json module to parse and serialize JSON data.

**183.What is the secrets module used for?** - To generate cryptographically strong random numbers suitable for managing data such as passwords and tokens.

**184.How do you count the occurrences of elements in a list?** - By using the Counter class from the collections module. Example:

python

Copy code

from collections import Counter

count = Counter(my\_list)

**185.What is the difference between a shallow copy and a deep copy?** - A shallow copy copies the object but not the nested objects, whereas a deep copy copies the object and all nested objects.

**186.What is the pandas library used for?** - For data manipulation and analysis, providing data structures and functions for working with structured data.

**187.How do you create a Pandas DataFrame?** - By using the pd.DataFrame() constructor. Example:

python

Copy code

import pandas as pd

df = pd.DataFrame({'col1': [1, 2], 'col2': [3, 4]})

**188.What is the openpyxl library used for?** - To read and write Excel 2010 xlsx/xlsm/xltx/xltm files.

**189.How do you handle missing data in Pandas?** - By using methods like dropna(), fillna(), and isna().

**190.What is the matplotlib library used for?** - For creating static, animated, and interactive visualizations in Python.

**191.How do you create a plot in Matplotlib?** - By using the pyplot module. Example:

python

Copy code

import matplotlib.pyplot as plt

plt.plot([1, 2, 3], [4, 5, 6])

plt.show()

**192.What is the seaborn library used for?** - For statistical data visualization, built on top of Matplotlib.

**193.How do you create a seaborn plot?** - By using functions like sns.scatterplot() and sns.barplot(). Example:

python

Copy code

import seaborn as sns

sns.scatterplot(x=[1, 2, 3], y=[4, 5, 6])

**194.What is the scikit-learn library used for?** - For machine learning, providing simple and efficient tools for data mining and data analysis.

**195.How do you train a machine learning model in scikit-learn?** - By using the fit() method on a model object. Example:

python

Copy code

from sklearn.linear\_model import LinearRegression

model = LinearRegression()

model.fit(X\_train, y\_train)

**196.What is the numpy library used for?** - For numerical computing, providing support for large, multi-dimensional arrays and matrices.

**197.How do you create a NumPy array?** - By using the np.array() function. Example:

python

Copy code

import numpy as np

arr = np.array([1, 2, 3])

**198.What is the scipy library used for?** - For scientific and technical computing, providing modules for optimization, integration, interpolation, eigenvalue problems, algebraic equations, and more.

**199.How do you perform linear algebra operations in SciPy?** - By using the scipy.linalg module. Example:

python

Copy code

from scipy.linalg import inv

matrix\_inv = inv(matrix)

**200.What is the tensorflow library used for?** - For machine learning and artificial intelligence, particularly for building and training neural networks.

**201.How do you create a neural network in TensorFlow?** - By using the tf.keras API. Example:

python

Copy code

import tensorflow as tf

model = tf.keras.models.Sequential([

tf.keras.layers.Dense(128, activation='relu'),

tf.keras.layers.Dense(10, activation='softmax')

])

**202.What is the flask framework used for?** - For building lightweight web applications.

**203.How do you create a simple Flask app?** - By using the Flask class. Example:

python

Copy code

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello\_world():

return 'Hello, World!'

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

app.run()

**204.What is the django framework used for?** - For building high-level web applications with an emphasis on rapid development and clean, pragmatic design.

**205.How do you create a Django project?** - By using the django-admin startproject command. Example:

bash

Copy code

django-admin startproject myproject

**206.What is the sqlalchemy library used for?** - For SQL and database management using the Object Relational Mapping (ORM) technique.

**207.How do you create a SQLAlchemy engine?** - By using the create\_engine() function. Example:

python

Copy code

from sqlalchemy import create\_engine

engine = create\_engine('sqlite:///mydatabase.db')

**208.What is the beautifulsoup4 library used for?** - For web scraping, parsing HTML and XML documents.

**209.How do you parse HTML with BeautifulSoup?** - By using the BeautifulSoup class. Example:

python

Copy code

from bs4 import BeautifulSoup

soup = BeautifulSoup(html\_doc, 'html.parser')

**210.What is the pytest framework used for?** - For writing and running test cases in Python.