TOP 100 PYTHON INTERVIEW QUESTIONS

Basic Python Concepts

1. **What is Python?**

   Answer: Python is a high-level, interpreted, and general-purpose programming language, known for its readability and simplicity.
2. **What are the key features of Python?**

   Answer: Easy syntax, interpreted, dynamic typing, large standard library, and support for object-oriented, imperative, and functional programming.
3. **What are Python's data types?**

   Answer: Integer, float, string, list, tuple, dictionary, set, and boolean.
4. **What is the difference between a list and a tuple in Python?**

   Answer: Lists are mutable (can be modified), while tuples are immutable (cannot be changed once created).
5. **How does Python handle memory management?**

   Answer: Python uses an automatic memory management system with a built-in garbage collector for memory cleanup.
6. **What is the purpose of self in Python?**

   Answer: self refers to the instance of the class and is used to access instance variables and methods.
7. **What is the difference between del, remove(), and pop() in Python?**

   Answer: del removes a variable, remove() removes a value from a list, and pop() removes an element at a specific index and returns it.
8. **What are Python decorators?**

   Answer: Decorators are functions that modify the functionality of other functions or methods.
9. **What is the difference between is and = = in Python?**

   Answer: = = compares values, while is compares the memory address of two objects.
10. **How do you handle exceptions in Python?**

    Answer: Exceptions are handled using the try, except, else, and finally blocks.

Control Flow and Loops

11. **What is the difference between for and while loops in Python?**

    Answer: A for loop iterates over a sequence, while a while loop continues as long as a condition is True.

12. **What is a break statement in Python?**

    Answer: The break statement is used to exit a loop prematurely.

13. **What is a continue statement in Python?**

    Answer: The continue statement skips the current iteration and moves to the next iteration of a loop.

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

    Answer: List comprehensions provide a concise way to create lists using a single line of code.

15. **Explain the use of the range() function in Python.**

    Answer: range() is used to generate a sequence of numbers, commonly used in loops.

16. **What are lambda functions in Python?**

    Answer: lambda functions are anonymous, small, one-line functions defined using the lambda keyword.

17. **What is the use of else with loops in Python?**

    Answer: The else block in a loop is executed when the loop completes normally (without a break).

18. **How do you handle multiple conditions in Python?**

    Answer: Using logical operators like and, or, and not.

19. **What is a pass statement in Python?**

    Answer: pass is a placeholder that allows you to write syntactically correct but empty code blocks.

20. **What is a try-except-finally block in Python?**

    Answer: It’s used for exception handling where try contains code that might raise an exception, except catches the exception, and finally always executes.

Functions

21. **What are functions in Python?**

    Answer: Functions are blocks of reusable code designed to perform a specific task.

22. **What is the difference between global and local variables in Python?**

    Answer: A global variable is defined outside any function and can be accessed anywhere in the program, while a local variable is defined inside a function and is accessible only within that function.

23. **How can you return multiple values from a function in Python?**

    Answer: By returning them as a tuple or using a list/dictionary.

24. **What are default arguments in Python functions?**

    Answer: Default arguments allow a function to be called with fewer arguments by providing default values for some parameters.

25. **What is the purpose of *args and **kwargs in Python?**

    Answer: *args allows passing a variable number of positional arguments, and **kwargs allows passing a variable number of keyword arguments.

26. **What is a yield statement in Python?**

    Answer: The yield statement is used to return an iterator (generator) from a function, which can be iterated lazily.

27. **How does Python handle recursion?**

    Answer: Python supports recursion, but it has a default recursion limit (usually 1000), which can be adjusted using sys.setrecursionlimit().

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

    Answer: A generator is an iterator created using a function with the yield keyword, which allows iterating over a sequence lazily.

29. **What is the difference between map(), filter(), and reduce() in Python?**

    Answer: map() applies a function to all items in an input list, filter() filters items based on a condition, and reduce() accumulates items into a single value.

30. **What is the use of the assert statement in Python?**

    Answer: assert is used for debugging purposes, ensuring that a condition holds true, otherwise raises an AssertionError.

Object-Oriented Programming (OOP) Concepts

31. **What is object-oriented programming in Python?**

    Answer: OOP is a programming paradigm that uses objects and classes to model real-world concepts.

32. **What are classes and objects in Python?**

    Answer: A class is a blueprint for creating objects, and an object is an instance of a class.

33. **What is inheritance in Python?**

    Answer: Inheritance allows a class to inherit attributes and methods from another class.

34. **What is method overloading and method overriding in Python?**

    Answer: Python does not support method overloading, but method overriding allows a subclass to provide its own implementation of a method defined in the parent class.

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

    Answer: super() is used to call methods from a superclass in a subclass.

36. **What are class variables and instance variables in Python?**

    Answer: Class variables are shared among all instances of a class, while instance variables are unique to each instance.

37. **What is encapsulation in Python?**

    Answer: Encapsulation is the concept of hiding the internal state of an object and only exposing methods to interact with that state.

38. **What is polymorphism in Python?**

    Answer: Polymorphism allows methods to have the same name but behave differently depending on the object type.

39. **What is the __init__() method in Python?**

    Answer: __init__() is a special method called when an object is instantiated, used to initialize the object's state.

40. **What is a static method in Python?**

    Answer: A static method is bound to the class rather than the instance and doesn't require a reference to the object (self).

Data Structures in Python

41. **What is the difference between a list and a dictionary in Python?**

    Answer: A list is an ordered collection of elements, while a dictionary is an unordered collection of key-value pairs.

42. **What is the use of sets in Python?**

    Answer: Sets are unordered collections of unique elements, useful for operations like union, intersection, and difference.

43. **How do you merge two lists in Python?**

    Answer: You can use the + operator or the extend() method to merge two lists.

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

    Answer: Slicing allows accessing a portion of a list, string, or tuple using the start:stop:step syntax.

45. **What are the built-in data structures in Python?**

    Answer: Lists, tuples, dictionaries, sets, and strings.

46. **How do you convert a list to a set in Python?**

    Answer: Use the set() function to convert a list to a set.

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

    Answer: pop() removes and returns an item at a specified index, while remove() removes the first occurrence of a specified value.

48. **What are defaultdict and OrderedDict in Python?**

    Answer: defaultdict returns a default value if the key does not exist, while OrderedDict maintains the order of keys based on insertion.

49. **How do you iterate over a dictionary in Python?**

    Answer: Use the for loop with items(), keys(), or values() to iterate over a dictionary.

50. **What is a deque in Python?**

    Answer: A deque (double-ended queue) allows fast appends and pops from both ends of the list.

File Handling

51. **How do you open and read a file in Python?**

    Answer: Use the open() function to open a file, and read() to read its contents.

52. **How do you write to a file in Python?**

    Answer: Use the write() or writelines() method after opening the file in write (w) or append (a) mode.

53. **What is the difference between read() and readlines() in Python?**

    Answer: read() reads the entire file content, while readlines() reads the file line by line and returns a list of lines.

54. **What is the with statement used for in file handling?**

    Answer: The with statement ensures proper resource management, automatically closing the file when the block is exited.

55. **How do you handle file exceptions in Python?**

    Answer: Use a try-except block to handle exceptions like FileNotFoundError.

56. **How do you check if a file exists in Python?**

    Answer: Use the os.path.exists() method to check if a file exists.

57. **What is the os module in Python?**

    Answer: The os module provides functions for interacting with the operating system, like file operations and environment variables.

58. **How do you rename a file in Python?**

    Answer: Use the os.rename() method to rename a file.

59. **What is the difference between os.remove() and os.rmdir()?**

    Answer: os.remove() deletes a file, while os.rmdir() deletes an empty directory.

60. **How do you work with CSV files in Python?**

    Answer: Use the csv module to read from and write to CSV files.

Libraries and Frameworks

61. **What are Python's most commonly used libraries?**

    Answer: NumPy, Pandas, Matplotlib, TensorFlow, Keras, Django, Flask, etc.

62. **What is NumPy used for in Python?**

    Answer: NumPy is a library for numerical computing, providing support for large multi-dimensional arrays and matrices, along with mathematical functions.

63. **What is the purpose of Pandas in Python?**

    Answer: Pandas is a data manipulation library used for working with structured data like CSV files, databases, and Excel files.

64. **What is Flask used for in Python?**

    Answer: Flask is a lightweight web framework for building web applications in Python.

65. **What is Django used for in Python?**

    Answer: Django is a high-level web framework for building complex web applications quickly.

66. **What is Matplotlib used for in Python?**

    Answer: Matplotlib is a plotting library used for creating static, animated, and interactive visualizations.

67. **What is SciPy used for in Python?**

    Answer: SciPy is used for scientific and technical computing, building on NumPy with additional functions for optimization, integration, and linear algebra.

68. **What is TensorFlow?**

    Answer: TensorFlow is a popular open-source library for machine learning and deep learning.

69. **What is Keras?**

    Answer: Keras is an API for building and training deep learning models, often used with TensorFlow.

70. **What is OpenCV used for in Python?**

    Answer: OpenCV is used for image and video processing tasks, including computer vision applications.

Advanced Topics and Performance

71. **What is the difference between shallow copy and deep copy in Python?**

    Answer: A shallow copy copies references to objects, while a deep copy copies the actual objects.

72. **What are Python generators and iterators?**

Answer: A generator is a function that produces items one at a time using the yield statement, and an iterator is an object that implements the __iter__() and __next__() methods.

73. **How do you optimize Python code for performance?**

Answer: Use built-in functions, avoid global variables, use list comprehensions, and profile code with cProfile.

74. **What is the Global Interpreter Lock (GIL) in Python?**

Answer: GIL is a mutex that prevents multiple native threads from executing Python bytecodes at once, limiting multi-threading performance.

75. **What is multithreading in Python?**

Answer: Multithreading allows concurrent execution of tasks, but due to GIL, it is best used for I/O-bound tasks.

76. **What is multiprocessing in Python?**

Answer: Multiprocessing allows concurrent execution of processes, bypassing the GIL and using separate memory space for each process.

77. **What are context managers in Python?**

Answer: Context managers are used to allocate and release resources, commonly used with the with statement.

78. **What is the difference between join() and threading in Python?**

Answer: join() is used to wait for a thread to finish, while threading is used to create and manage threads.

79. **What is the time module used for in Python?**

Answer: The time module provides functions for working with time-related tasks like measuring performance and delays.

80. **How do you profile Python code?**

Answer: Use the cProfile module to profile code and analyze performance bottlenecks.

Testing and Debugging

81. **What is unit testing in Python?**

    Answer: Unit testing involves testing individual units of code using a testing framework like unittest.

82. **What is the unittest module in Python?**

    Answer: The unittest module provides a framework for writing and running tests.

83. **What is the purpose of assert in Python?**

    Answer: assert is used to check if a condition is True, and if it's not, it raises an AssertionError.

84. **What are Python's built-in testing frameworks?**

    Answer: unittest, pytest, and nose.

85. **How do you handle logging in Python?**

    Answer: The logging module allows for logging messages, errors, and other important events in Python programs.

86. **What is the difference between debug and info in logging?**

    Answer: debug is used for detailed debugging information, while info is used for general information about program execution.

87. **What is mocking in Python testing?**

    Answer: Mocking involves simulating behavior for external dependencies in tests using the unittest.mock module.

88. **How do you test for exceptions in Python?**

    Answer: Use the assertRaises method in unittest to test whether an exception is raised.

89. **What is the difference between assert and raise in Python?**

    Answer: assert checks conditions and raises an AssertionError if the condition is false, while raise manually raises exceptions.

90. **What is the purpose of the nose testing framework?**

    Answer: nose is used for test discovery, running tests, and generating test reports.

Other Python Questions

91. **What is the difference between __str__() and __repr__() in Python?**

    Answer: __str__() is used for user-friendly string representations, while __repr__() is used for developer-friendly representations.

92. **What is the __del__() method in Python?**

    Answer: __del__() is a destructor method called when an object is about to be destroyed.

93. **What is the purpose of isinstance() in Python?**

    Answer: isinstance() checks whether an object is an instance of a specified class or tuple of classes.

94. **What is the difference between open() and os.open()?**

    Answer: open() is a higher-level function for file handling, while os.open() is a lower-level function with more control over file handling options.

95. **What is an abstract class in Python?**

    Answer: An abstract class is a class that cannot be instantiated directly and must be subclassed.

96. **What are try, except, else, and finally blocks used for in Python?**

    Answer: They are used for handling exceptions, where finally always executes after try and except.

97. **What is the purpose of with in Python?**

    Answer: with simplifies resource management (e.g., file handling) by automatically cleaning up resources when done.

98. **What is the os module in Python?**

    Answer: The os module provides functions for interacting with the operating system, such as file and directory manipulation.

99. **What is the difference between json and pickle in Python?**

    Answer: json is used for serializing data into a human-readable format, while pickle is used for serializing Python objects into a binary format.

100. **How do you update Python to the latest version?**

    Answer: You can update Python using a package manager like apt, brew, or by downloading the latest version from the official Python website.