Asked in Apple, Google,Amazon

150+ Python Programming Interview Questions

CODING WISE

QUESTIONS:

Certainly! Here's a comprehensive list of 150+ Python programming interview questions covering various topics, including Python basics, data types, control flow, functions, object-oriented programming, file handling, data structures, algorithms, Python libraries, and more:

Python Basics:

1. What is Python? What are the key features of Python?

2. Explain the differences between Python 2 and Python 3.

3. How do you install and set up Python on your computer?

4. How do you write a simple "Hello, World!" program in Python?

5. Explain the concept of Indentation in Python.

6. What are Python docstrings?

7. How do you handle exceptions in Python (try, except, else, finally)?

8. Discuss Python's Global Interpreter Lock (GIL).

Data Types and Variables:

9. What are the basic data types in Python?

10. Explain the differences between mutable and immutable objects in Python.

11. How do you convert one data type to another in Python?

12. What is variable scope in Python? Discuss global and local variables.

Control Flow:

13. Explain the if...else statement in Python.

14. Discuss the usage of loops (for loop and while loop) in Python.

15. How do you iterate over a dictionary and its keys/values in Python?

16. What are list comprehensions and how do they work?

Functions and Modules:

17. How do you define and call functions in Python?

18. What are lambda functions (anonymous functions) in Python?

19. Discuss recursion and how it can be used in Python functions.

20. How do you create and use modules in Python?

Object-Oriented Programming (OOP):

21. Explain the concepts of classes and objects in Python.

22. What are constructors in Python? How are they defined?

23. Discuss inheritance, encapsulation, and polymorphism in Python.

24. How do you handle method overriding and overloading in Python?

25. What is the difference between class and instance variables?

File Handling:

26. How do you open, read, and write to files in Python?

27. What are the different file modes in Python?

Strings:

28. How do you manipulate strings in Python? Discuss string methods.

29. Explain the concept of string formatting in Python.

Lists, Tuples, and Sets:

30. Discuss the differences between lists, tuples, and sets in Python.

31. How do you perform common operations on lists (e.g., slicing, concatenation)?

32. Explain the concept of list comprehension.

Dictionaries:

33. How do you create, access, and modify dictionaries in Python?

34. Discuss the differences between dictionaries and sets.

Recursion:

35. Explain recursion and discuss situations where it is suitable.

36. Demonstrate writing a recursive function in Python.

Sorting and Searching:

37. How do you sort a list in Python? Discuss various sorting algorithms.

38. Explain binary search and its implementation in Python.

Data Structures:

39. Discuss the usage and implementation of stacks and queues in Python.

40. How do you implement a linked list in Python?

Algorithmic Complexity:

41. Explain Big O notation and its significance.

42. Discuss the time and space complexity of common Python operations.

Python Libraries:

43. What are the uses of NumPy and how do you use it?

44. Discuss the purpose and usage of Pandas.

45. How do you handle JSON data in Python (using the `json` module)?

Exception Handling:

46. How do you raise custom exceptions in Python?

47. Discuss the differences between the `raise` and `assert` statements.

Generators:

48. What are Python generators, and how are they different from regular functions?

Decorators:

49. Explain the concept of decorators in Python and how to use them.

Threading and Concurrency:

50. How do you create and manage threads in Python?

51. Discuss the Global Interpreter Lock (GIL) and its impact on threading.

Regular Expressions:

52. What are regular expressions, and how do you use them in Python?

Python Virtual Environments:

53. What is a virtual environment, and why would you use one?

Database Connectivity:

54. How do you connect to a database in Python?

55. Discuss the usage of Python with SQLite.

Web Scraping:

56. How do you perform web scraping in Python using libraries like BeautifulSoup?

Testing:

57. Discuss the purpose and usage of the `unittest` library for testing in Python.

58. Explain the differences between unit testing and integration testing.

Flask Framework:

59. What is Flask, and how do you create a web application using it?

Django Framework:

60. What is Django, and how do you create a web application using it?

API Usage:

61. How do you interact with RESTful APIs in Python?

Data Serialization:

62. Discuss the differences between JSON and Pickle for data serialization.

Multi-threading and Multi-processing:

63. How do you achieve multi-threading and multi-processing in Python?

Python Packaging:

64. What is `pip`, and how do you create and distribute Python packages?

Memory Management:

65. Discuss Python's memory management and garbage collection mechanism.

Python Best Practices:

66. Explain PEP 8 and its significance in Python development.

67. Discuss best practices for writing efficient and clean Python code.

Design Patterns:

68. Explain common design patterns used in Python (e.g., Singleton, Factory, Observer).

Networking:

69. How do you work with sockets in Python?

Asynchronous Programming:

70. Discuss asynchronous programming in Python using `asyncio`.

Closures:

71. Explain closures in Python and their applications.

Data Encryption:

72. How do you perform data encryption and decryption in Python?

Web Development:

73. Discuss different frameworks and libraries for web development in Python.

Memory Profiling:

74. How do you profile memory usage in Python applications?

Performance Optimization:

75. Discuss strategies for optimizing Python code for better performance.

Operator Overloading:

76. Explain operator overloading in Python classes.

Context Managers:

77. What are context managers, and how do you use them with the `with` statement?

Data Validation:

78. How do you validate user input and data in Python applications?

Internationalization:

79. Discuss internationalization (i18n) and localization (l10n) in Python.

Data Compression:

80. How do you compress and decompress data in Python?

Debugging:

81. What tools and techniques do you use for debugging Python code?

Code Profiling:

82. How do you profile Python code for performance analysis?

API Authentication:

83. Discuss methods for securing APIs in Python.

Websockets:

84. How do you work with WebSockets in Python applications?

JSON Web Tokens (JWT):

85. Explain JWT and how to use it for authentication in Python.

Python and Machine Learning:

86. Discuss the usage of Python in Machine Learning and AI.

Image Processing:

87. How do you process images in Python using libraries like OpenCV?

Sentiment Analysis:

88. Discuss sentiment analysis and its implementation in Python.

Natural Language Processing (NLP):

89. How do you perform NLP tasks in Python using libraries like NLTK or spaCy?

Neural Networks and Deep Learning:

90. Discuss Python libraries for building and training neural networks

Sign up for answers

THANKS