

Python Interview Questions

1. Basic Python Interview Questions

1. What are the key features of Python?
 2. How is Python an interpreted language?
 3. What are Python's built-in data types?
 4. What is the difference between list, tuple, set, and dictionary?
 5. How does Python handle memory management?
 6. What is the difference between `is` and `==` in Python?
 7. What are *mutable* and *immutable* data types?
 8. What is PEP 8, and why is it important?
 9. How do you swap two variables in Python without using a temporary variable?
 10. What are Python's indentation rules?
-

2. Python Data Structures & Operations

11. How do you remove duplicates from a list?
 12. What is the difference between `append()` and `extend()` in a list?
 13. How does slicing work in Python?
 14. What are list comprehensions, and why are they useful?
 15. How do you merge two dictionaries in Python?
 16. What is the difference between `deepcopy()` and `copy()`?
 17. How do you iterate over a dictionary?
 18. How do you sort a dictionary by keys or values?
 19. What is the difference between `pop()` and `remove()` in lists?
 20. How do you count occurrences of an element in a list?
-

3. Functions & Lambda Expressions

21. What is the difference between `args` and `kwargs`?
22. What are lambda functions, and where are they used?
23. What is a higher-order function in Python?
24. What are Python's `map()`, `filter()`, and `reduce()` functions?
25. What is recursion, and can Python handle deep recursion?

Python Interview Questions

- 26. What is the difference between return and yield in Python?
 - 27. What is a closure in Python?
 - 28. What are decorators, and how are they used?
 - 29. What is the purpose of `functools.lru_cache()`?
 - 30. What are generator functions, and how are they different from normal functions?
-

4. Object-Oriented Programming (OOP) in Python

- 31. What is a class and an object in Python?
 - 32. What is the difference between instance methods, class methods, and static methods?
 - 33. What is inheritance, and what are its types?
 - 34. What is method overriding in Python?
 - 35. What are dunder (magic) methods in Python?
 - 36. What is the purpose of the `super()` function?
 - 37. What is the difference between `__str__` and `__repr__`?
 - 38. What is multiple inheritance, and how does Python handle it?
 - 39. What is polymorphism in Python?
 - 40. What is composition, and how is it different from inheritance?
-

5. Exception Handling

- 41. What are try, except, finally in Python?
- 42. How do you handle multiple exceptions in a single block?
- 43. What is a custom exception, and how do you create one?
- 44. What is the difference between raise and assert?
- 45. What happens if an exception is not handled?
- 46. What is the else block in an exception handling structure?
- 47. How do you re-raise an exception in Python?
- 48. How does Python's logging module work?
- 49. What are context managers, and how do they help in exception handling?
- 50. What are some best practices for handling exceptions?

Python Interview Questions

6. Python File Handling

- 51. How do you open and close a file in Python?
 - 52. What are the different file modes in Python?
 - 53. What is the difference between read(), readline(), and readlines()?
 - 54. How do you write data to a file?
 - 55. What is with statement, and why is it used for file handling?
 - 56. How do you check if a file exists in Python?
 - 57. How do you delete a file using Python?
 - 58. What is os.path module used for?
 - 59. What are the differences between shutil and os modules?
 - 60. How do you copy files in Python?
-

7. Python Modules & Packages

- 61. What is the difference between module and package?
 - 62. How do you import a module in Python?
 - 63. What are __init__.py files in Python?
 - 64. How do you install third-party modules in Python?
 - 65. What is the difference between import module and from module import *?
 - 66. What is virtual environment, and why is it important?
 - 67. How do you create and activate a virtual environment?
 - 68. What is pip, and how is it used?
 - 69. What are some commonly used built-in modules in Python?
 - 70. What is sys.path, and how does it affect module import?
-

8. Multithreading & Multiprocessing

- 71. What is the difference between multithreading and multiprocessing in Python?
- 72. What is the GIL (Global Interpreter Lock)?
- 73. How do you create and start a thread in Python?
- 74. What is the threading module in Python?

Python Interview Questions

- 75. What are locks and semaphores in threading?
 - 76. How does Python handle concurrent execution?
 - 77. What is the purpose of asyncio in Python?
 - 78. What is the difference between synchronous and asynchronous programming?
 - 79. How do you implement parallel processing using Python?
 - 80. How do you use concurrent.futures for parallel execution?
-

9. Django Framework

- 81. What is Django, and why is it used?
- 82. What are the main features of Django?
- 83. What is Django's MTV (Model-Template-View) architecture?
- 84. What is the difference between Flask and Django?
- 85. How do you install Django and start a project?
- 86. What are Django models, and how do they work?
- 87. How do you create a database in Django?
- 88. What is Django ORM (Object-Relational Mapping)?
- 89. What are migrations in Django?
- 90. What is the purpose of manage.py?
- 91. How do you define URL patterns in Django?
- 92. What is the difference between function-based views and class-based views?
- 93. What is Django's middleware, and how does it work?
- 94. What are static files in Django?
- 95. How does Django handle form validation?
- 96. What is the Django admin panel, and how do you customize it?
- 97. What is Django's authentication system?
- 98. What are Django signals, and how are they used?
- 99. What is Django REST Framework (DRF), and why is it used?
- 100. How do you deploy a Django project to production?