Here’s a list of **500 Python interview questions** across various categories. These questions cover basic, intermediate, and advanced Python concepts, so you can use them to prepare for interviews at different levels.

**Basic Python Questions:**

**1. What is Python?**

**2. What are the key features of Python?**

**3. Explain the difference between Python 2 and Python 3.**

**4. What is the use of the self keyword in Python?**

**5. What is the difference between a list and a tuple?**

**6. What are Python's mutable and immutable types?**

**7. What is the difference between del, remove(), and pop() in a list?**

**8. How do you copy a list in Python?**

**9. What is a Python dictionary?**

**10. How do you iterate over a list or tuple in Python?**

**11. Explain list comprehensions.**

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

**13. Explain the difference between == and is in Python.**

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

**15. What is the map() function in Python?**

**16. How do you convert a string to an integer in Python?**

**17. What is the difference between str() and repr() in Python?**

**18. How do you remove duplicates from a list in Python?**

**19. What is the zip() function in Python?**

**20. Explain the join() function in Python.**

**Intermediate Python Questions:**

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

**22. Explain the yield keyword in Python.**

**23. What is a decorator in Python?**

**24. What are Python’s built-in data types?**

**25. Explain how \*args and \*\*kwargs work.**

**26. What are Python's memory management techniques?**

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

**28. What is the purpose of the global keyword in Python?**

**29. What is the difference between deepcopy() and copy() in Python?**

**30. How does exception handling work in Python?**

**31. What is the difference between try-except and try-finally?**

**32. What is the purpose of the assert statement in Python?**

**33. Explain the Python classmethod and staticmethod.**

**34. How do you handle file I/O in Python?**

**35. What is the re module in Python?**

**36. What is the difference between range() and xrange()?**

**37. How do you raise exceptions in Python?**

**38. What is the \_\_init\_\_() method in Python?**

**39. What is a Python module?**

**40. Explain how to use import in Python.**

**Advanced Python Questions:**

**41. What are Python's OOP concepts?**

**42. Explain inheritance in Python.**

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

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

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

**46. How do you implement multiple inheritance in Python?**

**47. What is the purpose of super() in Python?**

**48. Explain method resolution order (MRO) in Python.**

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

**50. What are closures in Python?**

**51. Explain the difference between staticmethod and classmethod.**

**52. What are Python's \_\_slots\_\_?**

**53. What is the property function in Python?**

**54. What is a weakref in Python?**

**55. Explain the Python multiprocessing module.**

**56. How does Python handle threading?**

**57. Explain the difference between threading and multiprocessing in Python.**

**58. What is a coroutine in Python?**

**59. What is the asyncio module in Python?**

**60. How does Python's GIL work?**

**Data Structures & Algorithms:**

**61. What is the difference between a stack and a queue?**

**62. What is the difference between a list and a linked list?**

**63. Explain the binary search algorithm.**

**64. What are hash tables and how are they implemented in Python?**

**65. What is a hash function?**

**66. Explain the time complexity of common Python operations.**

**67. What is a tree?**

**68. What is a graph?**

**69. What is the breadth-first search (BFS) algorithm?**

**70. What is depth-first search (DFS)?**

**71. What are heaps and how are they implemented in Python?**

**72. Explain the merge sort algorithm.**

**73. What is quicksort?**

**74. What is dynamic programming?**

**75. Explain memoization in Python.**

**Python Libraries and Frameworks:**

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

**77. Explain the difference between NumPy arrays and Python lists.**

**78. What is Pandas used for in Python?**

**79. What are DataFrames in Pandas?**

**80. Explain how to read/write CSV files using Pandas.**

**81. What is Matplotlib used for?**

**82. What is Seaborn in Python?**

**83. What is Scikit-learn used for in Python?**

**84. What is Flask?**

**85. Explain Django’s architecture.**

**86. What is TensorFlow used for?**

**87. What is PyTorch?**

**88. What is SQLAlchemy?**

**89. What are Python's web scraping tools?**

**90. What is BeautifulSoup?**

**Web Development with Python:**

**91. What is the difference between Flask and Django?**

**92. What are web templates in Flask?**

**93. What is routing in Flask?**

**94. What is a context in Flask?**

**95. How do you handle forms in Flask?**

**96. Explain how to set up a database connection in Flask.**

**97. What is RESTful API in Flask?**

**98. Explain how to deploy a Flask application.**

**99. What are Django's built-in admin tools?**

**100. What is Django ORM?**

**Testing and Debugging:**

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

**102. What is pytest used for?**

**103. How do you write a test case in Python?**

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

**105. Explain debugging in Python.**

**106. How do you use logging in Python?**

**107. What is the difference between traceback and logging in Python?**

**Performance and Optimization:**

**108. What are Python's performance bottlenecks?**

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

**110. What are some techniques for optimizing Python code?**

**111. How does Python handle garbage collection?**

**112. What is gc module in Python?**

**113. Explain the concept of lazy loading in Python.**

**Python Advanced Concepts:**

**114. What is the difference between a class method and a static method?**

**115. What is an iterator in Python?**

**116. What is the difference between \_\_str\_\_() and \_\_repr\_\_()?**

**117. What is a Python descriptor?**

**118. What is a weak reference in Python?**

**119. What is a property in Python?**

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

These questions cover a wide range of topics, from basic syntax to more advanced concepts like OOP, performance optimization, and working with frameworks. To effectively prepare for interviews, I recommend practicing coding exercises and understanding the concepts behind each question.