

Here are **50 Python Developer Interview Questions**, divided into categories to cover a broad range of topics:

Core Python (Basics)

1. What are Python's key features?
2. What is the difference between `is` and `==` in Python?
3. What are Python's data types?
4. What is the difference between a list and a tuple?
5. What is a Python dictionary? How is it different from a list?
6. What is the use of `self` in Python classes?
7. What are Python's mutable and immutable types?
8. What are `*args` and `*kwargs`?
9. How is memory managed in Python?
10. What is the use of `pass`, `continue`, and `break`?

Object-Oriented Programming

11. What are the four pillars of OOP in Python?
12. How does Python implement inheritance?
13. What is method overloading and overriding in Python?
14. What is the difference between a class method, instance method, and static method?
15. What is the purpose of `super()`?

Data Structures & Algorithms

16. How do you implement a stack/queue in Python?
17. How do you reverse a linked list in Python?
18. What is the time complexity of searching in a dictionary?
19. How would you sort a list of tuples by the second element?
20. Write a Python function to check if a string is a palindrome.

Python Libraries

21. What is NumPy and how is it used?
22. What is Pandas and how does it differ from NumPy?
23. Explain the use of `matplotlib.pyplot`.
24. How do you use `scikit-learn` for a simple linear regression?
25. What are the key features of TensorFlow?

Web Development (Django/Flask)

26. What are the differences between Django and Flask?
27. How does Django handle security?
28. What is middleware in Django?
29. How does routing work in Flask?
30. What is a context processor in Django?

Error Handling & Debugging

31. What is exception handling in Python? Give an example.
32. What is the difference between `try-except` and `try-finally`?
33. What are common built-in exceptions in Python?
34. How do you raise a custom exception?
35. What are some debugging tools available in Python?

Modules and Packages

36. How do you import a module in Python?
37. What is the difference between a Python module and a package?
38. What is `__init__.py` used for?
39. How can you create a virtual environment?
40. What is pip and how do you use it?

💡 **Advanced Python & Deployment**

41. What is a generator? How is it different from a list?
42. What are decorators in Python?
43. What is a lambda function and where would you use it?
44. What is multithreading and multiprocessing in Python?
45. How do you deploy a Python web application to the cloud (e.g., Heroku, AWS)?

**Data Science & Machine Learning **

46. How do you handle missing data in Pandas?
47. What is the difference between classification and regression?
48. How do you evaluate the performance of a machine learning model?
49. What is the difference between `fit()` and `transform()` in scikit-learn?
50. What is overfitting, and how can it be prevented?