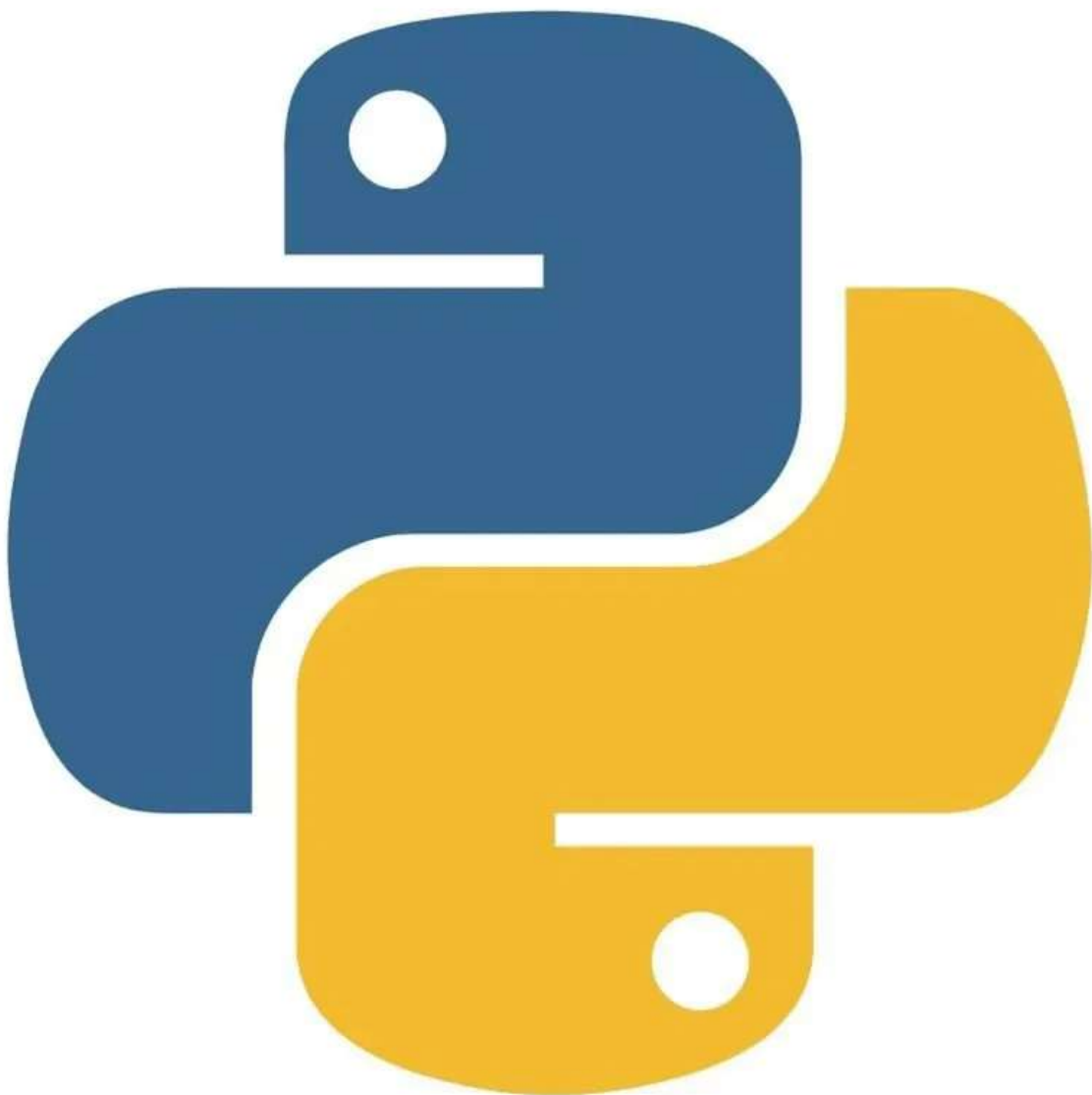


# 90+ Most Asked *Interview Questions* on **Python**



- What is Python and why is it popular?
- Explain the differences between Python 2 and Python 3.
- What are the key features of Python?
- How is memory managed in Python?
- What are Python's built-in data types?
- Explain the difference between lists and tuples.
- How does Python handle error and exceptions?
- What is the difference between break, continue, and pass?
- Explain the concept of duck typing in Python.
- What are iterators and generators?
- How are arguments passed in Python: by value or by reference?
- What is a lambda function?
- What is the purpose of the self keyword?
- How do you manage Python packages?
- Explain scope and namespace in Python.
- What is a decorator?



- How do you implement a stack and a queue in Python?
- Explain the difference between array and list.
- What are dictionaries and how are they used?
- How do you handle multidimensional arrays?
- What are sets and how are they different from lists?
- Explain the concept of list comprehensions.
- How do you implement a binary tree in Python?
- What are the use cases for a linked list?
- How would you sort a dictionary by value?
- How do you reverse a string?
- Write an algorithm to check if a string is a palindrome.
- Explain how to find the smallest/largest number in a list.



- Describe how to implement the bubble sort algorithm.
- What is recursion and how would you use it?
- How do you implement a binary search?
- Explain the merge sort algorithm.
- Describe an algorithm to find the common elements in two lists.
- What are the differences between depth-first and breadth-first search?
- How would you implement a Fibonacci sequence?
- What are some popular libraries in Python?
- How do you work with dates and times in Python?
- Explain how to use NumPy for numerical computations.
- What is Pandas and how is it used?
- How do you use Matplotlib for data visualization?





- Explain the use of the requests library for HTTP requests.
- What is Flask/Django and when would you use it?
- How do you work with databases in Python?
- What are the uses of the SciPy library?
- Explain how TensorFlow or PyTorch is used in machine learning.
- How do you debug a Python program?
- What are some common Python errors?
- Explain the use of logging in Python.
- How do you write unit tests in Python?
- What is the purpose of a docstring?
- How do you handle deprecation warnings?
- What tools do you use for static code analysis?
- Explain the use of profiling in Python.
- How do you optimize Python code for performance?
- What is the GIL (Global Interpreter Lock)?



- Explain the concept of metaclasses.
- How does Python support functional programming?
- What are descriptors?
- How do you manage memory in Python?
- Explain the concept of context managers and the with statement.
- What is monkey patching?
- How does multithreading work in Python?
- Explain multiprocessing and how it differs from threading.
- How do you handle asynchronous programming in Python?
- Explain the use of coroutines and async/await.
- How do you handle missing data in a dataset?
- Explain how to perform data aggregation in Pandas.
- What are the ways to manipulate text data in Python?
- Describe how to perform a merge operation on two dataframes.



- Explain the concept of data normalization and standardization.
- How do you handle form submissions in Flask or Django?
- Explain the MVC architecture in web development.
- How do you ensure web application security in Python?
- What is ORM and how do you use it in Python?
- How do you deploy a Python web application?
- How do you handle overfitting in machine learning models?
- Explain the difference between supervised and unsupervised learning.
- What is cross-validation in machine learning?
- How do you evaluate a machine learning model?
- What are the common challenges in machine learning?





- How do you work with JSON/XML in Python?
- Explain how to use Python for network programming.
- What are decorators and how are they used?
- How do you manage state in a Python application?
- What is a virtual environment and why is it useful?
- How do you manage large codebases in Python?
- Explain how Python can be used in data scraping.
- What are the best practices for Python coding?
- How do you keep up with Python updates and changes?
- What is your experience with Python in production environments?





- How do you stay informed about new Python libraries?
- What are your favorite resources for learning Python?
- What is the `__init__.py` file and how is it used?
- Explain the use of the `__str__` and `__repr__` methods in a Python class.
- How do you identify and improve the performance bottlenecks in a Python application?
- What is Cython, and how can it be used to improve Python code performance?
- Explain how to handle large datasets in Python without running out of memory.
- How do you use Python to interact with a NoSQL database like MongoDB?
- How would you implement a simple TCP/IP server in Python?