

# Python Important Questions

## Basic Concepts in Python

- What is Python? Who developed it and when?
- What are the key features and benefits of using Python?
- What is Python interpreter? How does it work?
- What are the main differences between Python 2 and Python 3?
- Is Python a compiled or interpreted language? Explain.

## Frequently Asked Questions

- How do you concatenate two lists in Python?
- Explain the difference between a list and a tuple.
- What is the difference between / and // operators?
- How does Python handle errors and exceptions?
- What are the built-in data types in Python (with examples: int, float, str, list, dict, set, tuple, bool).

## Common Interview Questions

- How are arguments passed in Python—by value or by reference?
- What are \*args and \*\*kwargs in function definitions?
- What is a lambda function? When would you use one?
- What is a docstring? How is it used?
- How do you generate random numbers in Python?
- What is the difference between deep copy and shallow copy?
- Describe data abstraction, encapsulation, inheritance, and polymorphism in Python.

## Conceptual and Long Questions

- Explain Python memory management. How does garbage collection work?
- How would you make a Python script executable on Unix?
- Discuss the Global Interpreter Lock (GIL) in Python.
- What are variable scopes and namespaces in Python?
- How do modules and packages work in Python?

## Coding and Practical Questions

- How do you delete a file using Python?

- Write a code to display the current time.
- Explain list comprehension and dictionary comprehension with examples.
- How do you debug a Python program? Mention pdb and logging.
- What is slicing? Provide syntax and some examples.

### **Advanced and Full Stack Development Questions**

- What is a decorator in Python? Where is it used?
- Explain the difference between @classmethod, @staticmethod, and regular instance methods.
- How do you manage front-end and back-end in Python full stack development?
- How do you handle databases and web APIs in Python?

### **Additional Important Questions**

- What is the Walrus operator?
- What is structural pattern matching (match-case statement)?
- How does Python's unit testing framework work?
- What are function annotations and exception groups in Python?

### **Best Practices and Design Principles**

- What are SOLID principles and how are they applied in Python?
- How do type hinting and logging improve code quality?
- When should you use lists vs. sets vs. dicts?
- How can you optimize code for large data processing (e.g., finding first non-repeating character in a 1 GB file)?

### **General and Basic Python Questions**

- What is Python? Who developed it and when?
- List Python's key features and advantages.
- What are the differences between Python 2 and Python 3?
- Explain interpreted vs. compiled languages.
- What is the scope and lifetime of a variable in Python?
- List mutable and immutable types in Python.
- What are membership operators? Give examples.

- What are the features of list data structure?
- What is the difference between lists and tuples?
- How do + and \* operators work with strings? Provide examples.

### **Frequently Asked and Conceptual Questions**

- What are built-in data types in Python?
- What are arrays vs. lists in Python?
- What does docstring mean?
- What is unit testing and why is it important?
- Explain break, continue, and pass statements.
- What is the use of self in Python classes?
- Explain global, protected, and private attributes in Python.
- How are modules and packages used?
- What is a constructor? Describe various constructors for type conversion.
- What is PEP 8 and why is it important?
- What is a dynamically typed language? How does Python achieve dynamic typing?

### **Syntax, Operators, and Functions**

- How to declare and call functions in Python?
- Explain \*args and \*\*kwargs with examples.
- How does try-except statement work? Example code.
- How do you import module attributes in Python scripts?
- Summarize various operators, built-in functions, and standard library modules for numeric types.
- What is slicing in Python? Give syntax and examples.
- Explain the split() and join() string functions.
- What is the difference between .py and .pyc files?

### **Advanced Interview Questions**

- Explain comprehensions (list/dict).
- What is a lambda function? Why use it?
- What are decorators? Show practical use.
- What are negative indexes and their purpose?
- Explain iterators and generators in Python.

- Discuss pickling and unpickling.
- What is the difference between range and xrange? When to use each?
- How can you copy objects (deep vs. shallow)?
- What is the Global Interpreter Lock (GIL) in Python?
- What is the motivation behind parallelism? How is it achieved?

### **Python Libraries and Modules**

- What are common Python libraries (numpy, pandas, matplotlib, SQLAlchemy, tkinter, re, os)?
- How does Python support regular expressions? Demonstrate core methods.
- Briefly explain about other GUI modules in Python (besides Tkinter).

### **Object-Oriented Programming (OOP)**

- What is inheritance? Give examples.
- How will you check if a class is a child of another class?
- What is init and finalize used for?
- Describe new and override modifiers.
- Are access specifiers used in Python? Demonstrate protected/private/public.
- How to access parent members in the child class (using Parent name and super())?
- How do you create an empty class?
- What is Object Relational Mapping (ORM)? Case study using SQLAlchemy.

### **Practical Coding Questions and Examples**

- How do you delete a file in Python?
- Write code to print intersection of two lists (without sets).
- Convert seconds to days, hours, and minutes.
- Convert height in feet/inches to cm.
- Print all prime numbers less than 256.
- Interchange first and last characters of a string.
- Create and use file objects, methods, and built-in attributes.
- Use persistent storage in Python.
- Importing modules and attributes.

### **GUI and Web Programming**

- What is the Tkinter module? How to create Label, Button, Entry, and ListBox widgets?

- Explain geometry managers (pack, grid, place) with examples.
- How to build a basic web browser with Tkinter?
- Discuss web applications using Python's CGI module.

### **Database Programming**

- What is a cursor object? List its methods and attributes.
- Show database connectivity and insert/query data using DB-API (example code).
- Create a Product Table with ORM (SQLAlchemy); insert/select data.
- Write queries for an Item table (display, filter, delete, etc.).
- List functions to retrieve rows from a table.

### **Multithreading and Parallelism**

- What is multithreading? How to achieve it?
- When should multithreading be used in Python?
- Discuss synchronization and the GIL.
- Compare processes versus threads.
- Discuss Thread and Threading modules: starting new threads, synchronization, examples.

### **Miscellaneous and Applied Questions**

- State four popular application domains for Python.
- Explain basic file handling (open, read, seek, tell).
- Give overview and demonstration of building web applications with CGI module.
- Describe SQL commands/DB operations for create/use/drop database.
- How can you optimize code for large data processing (e.g., non-repeating character in large file)?
- What is debugging in Python? Mention pdb and logging.