

📖 LEVEL 1: Python Fundamentals (Beginner)

Goal: Syntax clarity, basic programming logic

1. Python history & features
2. Python installation & execution
3. Variables & naming rules
4. Data types
 - int, float, complex
 - bool, str
5. Type conversion & casting
6. Input / Output (input(), print())
7. Operators
 - Arithmetic
 - Relational
 - Logical
 - Assignment
8. Keywords & identifiers
9. Comments & indentation
10. Basic built-in functions
(type(), id(), len())

📌 Interview focus:

Syntax correctness, beginner traps, output prediction

📖 LEVEL 2: Control Flow & Core Data Structures

Goal: Logical thinking + data handling

1. Conditional statements
 - if, elif, else
2. Loops
 - for, while
3. Loop control
 - break, continue, pass
4. range() function
5. Strings
 - slicing
 - methods (upper, lower, find, split)
6. Lists
 - indexing & slicing
 - list methods
7. Tuples & immutability
8. Sets & set operations
9. Dictionaries

- keys, values, methods
10. Truthy & falsy values

📌 **Interview focus:**

Output-based questions, logic flow

🧠 **LEVEL 3: Functions & Intermediate Python**

Goal: Reusability, scope, functional concepts

1. Function definition & calling
2. Arguments
 - positional
 - keyword
 - default
 - variable length (*args, **kwargs)
3. Return statement
4. Variable scope
 - local
 - global
 - nonlocal
5. Lambda functions
6. Recursion
7. List / Dict / Set comprehensions
8. map(), filter(), reduce()
9. Docstrings
10. Python modules & imports

📌 **Interview focus:**

Code comprehension, functional patterns

⚙️ **LEVEL 4: OOP & Exception Handling**

Goal: Object-oriented design & robustness

1. Classes & objects
2. `__init__()` constructor
3. Instance vs class variables
4. Methods (instance, class, static)
5. Encapsulation
6. Inheritance
 - single
 - multilevel
 - multiple
7. Polymorphism
8. Method overriding

9. Special (dunder) methods
10. Exception handling
 - try, except, else, finally
11. Custom exceptions

 **Interview focus:**

OOP concepts + real-world scenarios

 **LEVEL 5: Advanced Python & Standard Library**

Goal: Deep language understanding

1. Iterators & generators
2. yield keyword
3. Decorators
4. Context managers (with)
5. File handling
6. datetime, math, random
7. collections module
8. itertools
9. Regular expressions (re)
10. Memory management
11. Garbage collection
12. Shallow vs deep copy
13. Multithreading vs multiprocessing
14. GIL (Global Interpreter Lock)

 **Interview focus:**

Performance, internals, tricky concepts

 **LEVEL 6: Professional & Interview-Grade Python**

Goal: Industry-level mastery

1. Python internals
2. CPython vs PyPy
3. Bytecode & interpreter
4. Virtual environments
5. Packaging (pip, venv)
6. Testing
 - unittest
 - pytest
7. Logging
8. Debugging techniques
9. Code optimization
10. PEP 8 & best practices

11. Design patterns in Python
12. Async programming
 - async / await
13. Security best practices
14. Real interview trick questions
15. Output-based complex code snippets

 **Interview focus:**

Senior-level reasoning & system thinking

REMOVIZ.COM