Core Python Detailed Notes (Interview Ready)

1. Data Types & Variables

Python is dynamically typed, meaning you don’t need to declare variable types explicitly. Variables are references to objects stored in memory. - **Basic types**: int (whole numbers), float (decimal numbers), str (text), bool (True/False), NoneType (represents null). - **Collections**: * List → Ordered, mutable collection. * Tuple → Ordered, immutable collection. * Set → Unordered, unique elements only. * Dict → Key-value pairs, unordered (insertion ordered from Python 3.7+). - **Type Casting**: Use int(), float(), str(), list(), tuple(), set(), dict(). - **Interview Tip**: Be clear about mutable (list, dict, set) vs immutable (tuple, str, int). This often comes up in interviews.

2. Control Flow

Control flow structures dictate how code executes: - **Conditional statements**: if, elif, else. Python uses indentation instead of braces. - **Loops**: * for loops iterate over sequences (lists, tuples, strings, dicts). * while loops run until a condition is False. - **Comprehensions**: Provide concise ways to create sequences. Example: [x**2 for x in range(5)]. - **Loop control**: break (exit loop), continue (skip current iteration), pass (do nothing placeholder). - **Interview Tip**: Be ready to explain how list comprehension differs from generator expressions (memory efficiency).

3. Functions

Functions group reusable blocks of code: - Defined with def keyword. Functions must be defined before use. - **Arguments**: * Positional → matched by order. * Keyword → explicitly named. * Default → parameters with default values. * *args → variable length positional arguments. * **kwargs → variable length keyword arguments. - **Lambda functions**: Anonymous functions using lambda keyword. Example: lambda x: x*2. - **Docstrings**: Use triple quotes to describe a function. Good practice for interviews. - **Interview Tip**: Be prepared to compare pass-by-value vs pass-by-reference. In Python, arguments are passed by object reference.

4. Object-Oriented Programming (OOP)

Python supports OOP for code organization: - **Class**: Blueprint created using class keyword. - **Objects**: Instances of a class. - **Attributes**: Variables inside class. **Methods**: Functions inside class. - **Constructor**: __init__ method initializes object attributes. - **Encapsulation**: Restrict access using _var (protected) or __var (private convention). - **Inheritance**: Child class inherits properties from parent. Multiple inheritance is allowed in Python. - **Polymorphism**: Same method name, different implementation (method overriding). Operator overloading is possible (e.g. __add__, __len__). - **Special Methods**: __str__, __repr__, __eq__ provide readable representations and custom behavior. - **Interview Tip**: Be ready to explain differences between __str__ vs __repr__ and multiple inheritance method resolution order (MRO).

5. Exception Handling

Exceptions handle runtime errors gracefully: - **try-except-finally**: try block executes code, except handles errors, finally always runs. - **Multiple except**: Catch specific exceptions like ValueError, TypeError separately. - **Raise keyword**: Manually throw exceptions. - **Custom Exceptions**: Define a new class inheriting from Exception. - **Interview Tip**: Often asked why exception handling is better than checking return codes. Answer: improves readability, separates error handling from logic.

6. File Handling

Python makes file operations easy: - Open files with open(filename, mode). Always close after use. - **Modes**: 'r' (read), 'w' (write, overwrite), 'a' (append), 'b' (binary mode). - **Read methods**: read(), readline(), readlines(). - **Write methods**: write(), writelines(). - Use 'with open(...) as f' to auto-close (context manager). - **Interview Tip**: Be clear on difference between text vs binary mode. For example, images use binary mode.

7. Modules & Packages

Modules and packages organize reusable code: - Import with import module or from module import func. - Built-in modules: math, os, sys, datetime, random. - Custom modules: Any .py file can be imported as module. - Packages: Directories containing __init__.py. - **Interview Tip**: Be prepared to explain difference between module and package. Module = single file. Package = collection of modules.

8. Important Interview Concepts

- Mutable vs Immutable: Lists, dicts, sets are mutable; strings, tuples, ints are immutable. - Deep copy vs Shallow copy: copy.copy() (shallow), copy.deepcopy() (deep). - Python Memory Management: Reference counting + garbage collector. - Global Interpreter Lock (GIL): Only one thread executes Python bytecode at a time. Important for multithreading questions. - Difference between is vs ==: 'is' checks identity (same object in memory), '==' checks equality (same value).