Advanced Python – Deep Topics Notes

A comprehensive, interview■ready and real■world focused collection of advanced Python concepts. These notes explain what, why, and how each concept works internally.

Duck Typing

- Behavior■based typing
- Method existence over inheritance
- Loose coupling

EAFP vs LBYL

- Exception■driven vs check■driven logic
- Python prefers EAFP

Iterator Protocol

- __iter__ and __next__
- Lazy iteration

Generators & Yield

- Stateful execution
- Memory efficiency

Generator Expressions

- Lazy evaluation
- Low memory usage

Decorators

- Function wrapping
- Cross■cutting concerns

Decorator with Arguments

- Decorator factories
- Configurable behavior

Closures

- Lexical scoping
- State retention

Late Binding

- Loop variable capture

- Default argument fix

Magic Methods

- Operator overloading
- Pythonic objects

Operator Overloading

- Readable APIs
- Custom behavior

Context Managers

- Resource management
- __enter__ / __exit__

contextlib

- Generator■based context managers

Monkey Patching

- Runtime modification
- Testing use cases

Pass by Object Reference

- Mutable vs immutable behavior

Mutable vs Immutable

- Memory & side■effects

Shallow vs Deep Copy

- Reference vs duplication

Descriptors

- Attribute control
- __get__ / __set__

Property Decorator

- Encapsulation

Metaclasses

- Class creation control

MRO

- C3 linearization

Multiple Inheritance

- Diamond problem resolution

__slots__

- Memory optimization

Garbage Collection

- Reference counting
- Cycle detection

Weak References

- Cache patterns

GIL

- Threading limitation

Multithreading

- I/O bound tasks

Multiprocessing

- CPU bound parallelism

Async Programming

- Non■blocking I/O

async / await

- Coroutines

Event Loop

- Task scheduling

Futures

- Deferred results

Thread Safety

- Race conditions

Pickling

- Object serialization

JSON vs Pickle

- Security & portability

Import System

- Module caching

__main__

- Script entry point

Memory Model

- Heap & stack

Bytecode

- CPython execution

eval / exec

- Dynamic execution

Type Hints

- Static analysis

dataclasses

- Boilerplate reduction

NamedTuple

- Lightweight objects

Functional Tools

- map/filter/reduce

Zip Deep Use

- Parallel iteration