

# Advance Python Programming

Go from writing Python to engineering it: clean, fast, production-grade code you can ship.

Duration: 1 month

Certificate: NUEXUS Advance Python completion certificate

Format: Online or onsite, 1-on-1 or batch

NUEXUS certificate on completion

## OVERVIEW

Advance Python Programming takes developers who already know Python basics and turns them into confident, professional Python engineers. Over roughly one month of mentor-led, project-based sessions, you go deep on the language internals, idiomatic patterns, concurrency, testing, and packaging that separate hobby scripts from maintainable, production-ready code. Every module is hands-on and built around real labs and a capstone project, delivered online from anywhere or onsite in Islamabad, in a batch or 1-on-1, with certified trainers reviewing your work. You finish with a portfolio-worthy project and a NUEXUS Advance Python completion certificate.

## SKILLS YOU WILL GAIN

Advanced Python

OOP

Decorators & Generators

Concurrency

File & Data Handling

APIs & Requests

Testing

Packaging

Automation Scripting

Clean Code

## WHAT YOU WILL BE ABLE TO DO

- Write clean, idiomatic, PEP 8-compliant Python that other engineers can read and maintain
- Master advanced language features: comprehensions, generators, decorators, context managers, and iterators
- Apply object-oriented and functional patterns effectively, including dunder methods, dataclasses, and composition
- Write concurrent and parallel code using threading, multiprocessing, and asyncio, and know when to use each
- Test, debug, profile, and optimize your code with pytest, logging, and Python's profiling tools
- Structure, package, and distribute a real Python project with virtual environments, dependencies, and a clean layout

## WHO IT IS FOR

- Developers comfortable with Python basics who want to write cleaner, faster, more idiomatic code
- Backend, data, automation, and DevOps engineers who use Python daily and want to go deeper
- Self-taught programmers who can build scripts but want professional structure, testing, and packaging habits
- Computer science students and graduates preparing for Python-heavy technical roles
- Career changers moving into software, data, or automation who need production-grade Python skills

## PREREQUISITES

- Working knowledge of Python fundamentals: variables, data types, loops, conditionals, and functions
- Comfort writing and running simple Python scripts on your own machine
- A laptop with Python 3 installed (we help you set up the environment on day one)
- Basic command-line familiarity is helpful but not required; we cover what you need

## WHAT'S INCLUDED

- Mentor-led live sessions, online or onsite, in batch or 1-on-1
- Hands-on labs and real-world projects, not slides alone
- Recordings of your own sessions to revisit anytime
- Exam-aligned study material and practice questions
- Doubt-clearing and revision support through the program
- A verifiable NUEXUS completion certificate
- Career and next-step guidance after you finish

## COURSE CURRICULUM

### 1 Pythonic Foundations and Clean Code

- PEP 8, readability, and idiomatic Python conventions
- Deep dive into data structures: lists, dicts, sets, tuples, and when to use each
- Comprehensions and generator expressions for concise, efficient code
- Unpacking, slicing, and truthiness patterns
- Structuring code into clean, reusable functions and modules

### 2 Functions, Closures, and Decorators

- First-class functions, closures, and scope (LEGB rule)
- args, kwargs, and flexible function signatures
- Writing and stacking decorators
- functools: wraps, lru\_cache, partial, and reduce
- Practical use cases: timing, caching, logging, and access control

### 3 Iterators, Generators, and Lazy Evaluation

- The iterator protocol and building custom iterators
- Generator functions and the yield keyword
- Generator pipelines and lazy data processing
- The itertools module for efficient iteration
- Memory-efficient handling of large datasets and streams

### 4 Advanced Object-Oriented Programming

- Classes, inheritance, composition, and MRO
- Dunder (magic) methods and operator overloading
- Properties, class methods, and static methods
- dataclasses and named tuples for clean data models
- Abstract base classes, mixins, and design for extensibility

### 5 Error Handling, Logging, and Debugging

- Exception hierarchy and writing custom exceptions
- try, except, else, finally and context managers with the with statement
- Building your own context managers with contextlib
- Structured logging with the logging module
- Debugging with pdb, breakpoints, and effective tracebacks

### 6 Concurrency and Parallelism

- The GIL and what it means for Python performance
- Threading for I/O-bound work
- Multiprocessing for CPU-bound work
- Asynchronous programming with asyncio, async, and await
- Choosing the right concurrency model for the job

## 7 Testing, Profiling, and Optimization

Unit testing with pytest and fixtures

Test structure, mocking, and coverage

Profiling with cProfile and timeit to find bottlenecks

Common performance patterns and optimization techniques

Type hints and static checking with typing and mypy

## 8 Working with Data and External Systems

File handling, JSON, CSV, and serialization

Interacting with REST APIs using requests

Introduction to database access and SQL from Python

Virtual environments and dependency management with pip and venv

Reading and writing well-structured data pipelines

## 9 Project Structure, Packaging, and Capstone

Organizing a real-world Python project layout

Building an installable package with pyproject.toml

Managing dependencies and versioning

Version control basics for Python projects with Git

Capstone: design, build, test, and package an end-to-end Python project

## CAREER OUTCOMES

Python Developer

Backend Developer

Automation Engineer

Data Engineer

Software Engineer

## CERTIFICATION

On finishing the course and completing the capstone and assessments, you earn the NUEXUS Advance Python completion certificate, which recognizes that you have built and demonstrated production-level Python skills through hands-on work. The assessment is project and practical based rather than a single high-stakes exam; we support you throughout, but as with any honest program, completion depends on your own work and no result is guaranteed.