Advanced Python Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

The **Advanced Python** course is designed for developers and technical professionals who want to move beyond basic Python syntax and build more reliable, efficient, modular, and maintainable applications.

The programme explores widely used Pythonic practices and advanced language features that help developers create cleaner and more scalable solutions.

Participants take a deeper look at advanced data structures, exception handling, metaprogramming, regular expressions, advanced file handling, asynchronous programming, multithreading, multiprocessing, design patterns, debugging, profiling, and unit testing.

This is a highly practical programme, with approximately**50% hands-on labs and 50% instructor-led learning**. The course combines demonstrations, group discussions, coding labs, and project work to reinforce the techniques covered.

By the end of the programme, participants will have a stronger understanding of advanced Python capabilities and will be able to write more robust applications, improve performance, automate tasks, create reusable components, and build effective unit test suites.

Prerequisites

Participants should have:

- Experience writing Python scripts
- User-level knowledge of Unix/Linux, macOS, or Windows

Familiarity with basic Python syntax, functions, loops, collections, and file handling will help participants gain more value from the course.

Who Should Attend

This programme is intended for learners who already have practical Python programming experience and want to explore more advanced features of the language.

It is particularly suitable for:

- Python developers
- Software developers
- Automation engineers
- DevOps professionals
- Technical staff using Python in application or data environments
- Developers who want to automate tasks and improve code quality

Advanced Python Training

Duration: **5 Day**

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By the end of the programme, participants will be able to:

- Extend and enhance Python classes
- Apply advanced metaprogramming concepts
- Write more robust code using exception handling
- Work effectively with Python's advanced data structures
- Search and replace text using regular expressions
- Create easy-to-use and maintainable modules and packages
- Build multithreaded and multi-process applications
- Understand asynchronous programming concepts
- Create and execute unit tests
- Use PyTest and related testing tools
- Analyse and profile Python applications
- Apply debugging and benchmarking techniques
- Write more modular and Pythonic code

Outline

Day 1

Python Refresher

The course begins with a review of essential Python concepts.

Topics include:

- Built-in data types
- Lists
- Tuples
- Dictionaries
- Sets
- Program structure
- Files
- Console I/O
- If statements
- For loops
- While loops

Data Structures and Algorithms

This section introduces common computer science data structures and algorithms in a Python context.

Topics include:

- Linked Lists
- Stacks
- Queues

- Trees
- Graphs
- Sorting Algorithms

Participants review how these structures work and how they can be implemented using Python.

Day 2

Errors and Exception Handling

This module focuses on building more reliable applications through structured error handling.

Topics include:

- Syntax errors
- Exceptions
- try / except / else / finally
- Handling multiple exceptions
- Ignoring selected exceptions
- Designing predictable error-handling behaviour

Implementing Regular Expressions

Participants learn how to perform advanced text searching and manipulation using Python regular expressions.

Topics include:

- RE Objects
- Searching
- Matching
- Searching datasets with regular expressions
- Searching Wireshark trace data
- Working with Python and .pcap files
- Compilation flags
- Groups
- Special groups
- Replacing text
- Splitting strings

Advanced Functional Features of Python

This section explores advanced functional programming techniques.

Topics include:

- Advanced unpacking
- List Comprehensions
- Anonymous functions
- Lambda expressions
- Generator Expressions
- Decorators
- Closures

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Single dispatch
- Multi dispatch
- Relative imports
- Effective use of __init__
- Documentation best practices

Day 3

Metaprogramming

This section examines Python's dynamic programming capabilities.

Topics include:

- OOP conventions
- Class data
- Static data
- Class methods
- Static methods
- Creating classes from dictionary-based information
- The super() method
- Metaclasses
- Abstract Base Classes
- Implementing protocols through special methods
- Context protocols
- Iterator protocols
- Implicit properties
- globals()
- locals()
- Working with object attributes
- The inspect module
- Callable classes
- Monkey patching

Advanced File Handling

Participants learn more sophisticated approaches to working with the file system.

Topics include:

- Paths
- Directories
- Filenames
- Checking for existence
- Permissions
- File attributes
- Walking directory trees
- Creating filters with fileinput
- Using shutil
- Advanced file operations

Day 4

Advanced Python Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Advanced Data Structure Features in Python

This module focuses on advanced collection types and persistence.

Topics include:

- defaultdict
- Counter
- namedtuple
- Data Classes
- pickle
- Pretty printing
- Compressed archives
- ZIP
- GZIP
- TAR
- Persistent data

Multiprogramming and Concurrency

Participants explore different concurrency models available in Python.

Topics include:

- Concurrent programming
- Multithreading
- The threading module
- Sharing variables
- The queue module
- The multiprocessing module
- Creating process pools
- Coroutines
- Asynchronous programming

Python Design Patterns

This section examines reusable software design approaches.

Topics include:

- Why design patterns are useful
- Types of design patterns
- Creational patterns
- Structural patterns
- Behavioral patterns
- Best coding practices

Day 5

Developer Tools

This section focuses on tools used in professional Python development.

Advanced Python Training

Duration: **5 Day**

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com

Topics include:

- Analysing programs with pylint
- Using the debugger
- Profiling code
- Benchmarking
- Measuring execution speed

Unit Testing with PyTest

Participants learn how to write and run automated tests for Python applications.

Topics include:

- What is a unit test?
- Testing with the unit-test framework
- Testing with PyTest
- Testing with doctest
- Writing tests
- Working with fixtures
- Test runners
- Mocking resources
- Testing realistic application behaviour

Practical Application Projects

The final part of the programme gives learners the opportunity to apply the techniques covered during the course.

Command-Line Minesweeper

Build a command-line version of the classic Minesweeper game.

Automated File Renaming

Create a Python application that scans a folder and renames files according to rules defined in the code.

Binary Search

Implement the binary search algorithm and examine its behaviour.

Random Password Generator

Build a Python application that generates random passwords based on selected criteria.

Countdown Timer

Create a countdown timer using Python's `time` module.

Advanced Python Training

Duration: **5 Day**

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com