

Comprehensive Python Programming

Applied Technology Academy

A five-day, hands-on Python course that takes participants from writing and running their first scripts to file operations, regular expressions, binary data, networking and the standard library's work-saving modules. Extra emphasis falls on the features that are distinctly Python - tuples, slices, comprehensions, generators and output formatting. This is an in-depth exploration of working with the language rather than an academic tour of syntax, so participants can use Python on real tasks immediately.

LEVEL Introductory to Intermediate	DURATION 5 Days	DELIVERY Instructor-led
---	----------------------------------	--

Course Overview

- Five days, roughly 50% hands-on lab and 50% instruction, demonstration and discussion.
- Introductory-and-beyond: starts at the basics and continues well past them.
- Emphasis on features unique to Python: tuples, array slices, comprehensions and output formatting.
- Materials: a 500+ page course manual and a double-sided Python 3 quick reference.

Course Outline

- 1. The Python Environment
 - Starting Python, using the interpreter, running a script, editors and IDEs.
- 2. Variables and Values
 - Using variables, builtin functions, string data, numeric data, converting types.
- 3. Basic Input and Output
 - Writing to the screen, string formatting, command-line arguments, reading the keyboard.
- 4. Flow Control
 - The if statement, relational and Boolean values, while loops, exiting from loops.
- 5. Array Types
 - Sequence types, lists and list methods, tuples, indexing and slicing, iteration,
 - sequence functions and operators, list comprehensions and generators.
- 6. Working with Files
 - File I/O overview, opening, reading and writing text files.
- 7. Dictionaries and Sets
 - Creating dictionaries, getting values, iteration, creating and working with sets.
- 8. Functions
 - Defining functions, returning values, parameters and arguments, variable scope.

- 9. Sorting
 - The sorted() function, custom sort keys, lambda functions, reverse sorting, min() and max().
- 10. Exception Handling and Logging
 - Exceptions, try/except/else/finally, handling multiple exceptions, logging setup and basic logging.
- 11. Modules and Packages
 - Creating modules, the import statement, module search path, using packages, function and module aliases.
- 12. Introduction to Classes
 - Object-oriented programming, defining classes, constructors, self, properties, instance and class methods and data, inheritance.
- 13. Regular Expressions
 - RE syntax and objects, searching and matching, compilation flags, groups, search-and-replace, splitting strings.
- 14. Dates and Times
 - Representations, parsing dates from text, formatting, converting, calendar data, time zones.
- 15. Working with the File System
 - Paths, directories and filenames, checking existence, permissions and attributes, walking directory trees, shutil operations.
- 16. Advanced Data Handling
 - defaultdict and Counter, pretty-printing data structures, compressed archives, persistent data.
- 17. Network Programming
 - Using requests, grabbing web content, sending email, SSH for remote access, FTP.
- 18. Effective Scripts
 - Reading input files Unix-style, parsing command-line options, detecting the platform, logging.
- 19. Virtual Environments
 - Why they are needed, creating one, replicating an environment, common issues.

Prerequisites

- Basic familiarity with any programming or scripting language is helpful.
- A working, user-level knowledge of Linux, macOS or Windows.
- No prior Python experience is required.

Who Should Attend

- Developers who want to use Python across a wide range of work.
- Anyone who wants to automate or simplify common tasks with Python scripts.
- Technical staff moving to Python from another language.

What You'll Learn

By the end of this course, participants will be able to:

- create working Python scripts that follow current best practice
- use Python data types appropriately, including the lesser known but powerful ones
- read and write files containing both text and binary data
- search and replace text with regular expressions
- work productively with the standard library and its work-saving modules
- know when to reach for lists, tuples, dictionaries or sets
- use Pythonic features such as list comprehensions and generators
- write robust code with exception handling and logging
- work with dates, times and calendars
- create and replicate virtual environments

Environment and Tools

- Any operating system.
- Python 3 (python.org or Anaconda).
- A Python-aware IDE - Visual Studio Code recommended; PyCharm Community Edition or Eclipse also work.
- Detailed setup instructions are supplied before class.

Ready to enroll? Call 800.674.3550 or email Info@AppliedTechAc.com — private team cohorts, GSA MAS purchasing, military credentialing funding, VA GI Bill and VR&E and student financing available.