Faysal Aziz

Foundations & Intro to Python

Build a rock-solid grounding in Python, from first installation to writing real code with confidence

FA-0211 · two-day

Software Development

Introduction

Python continues to be one of the most widely used programming languages in technology today because of its clarity, readability, and versatility. Whether learners aim to build simple scripts, automate tasks, or eventually explore data science, web development, or AI, understanding Python's core concepts is essential. This two-day course focuses on foundational skills: understanding syntax, data structures, control flow, functions, and practical coding patterns that prepare participants to continue their learning independently or begin real projects. The instructor brings over 30 years of industry experience and prioritizes real-world relevance over academic theory, grounding lessons in patterns and practices seen in professional software development.

Learning outcomes

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

- Install and set up Python and a development environment.
- Write and execute Python scripts using correct syntax.
- Use core Python data types, variables, and operators to solve problems.
- Implement control flow with conditionals and loops.
- Define and call functions, understand scope, and modularize code.
- Work with Python built-in data structures like lists, tuples, dictionaries, and sets.
- Read from and write to files.
- Apply simple debugging techniques and interpret runtime errors. (

Prerequisites

To get the most from this course, participants should have:

- Basic familiarity with using a computer (files, folders, and text editors).
- Understanding of programming concepts.
- Curiosity and willingness to write code hands-on.

Training outline

1. 1. Getting Started with Python

Introduction to Python

- What Python is and where it's used (AI, web, automation, data science).
- Why Python's simplicity and readability matter.
- Overview of Python versions (Python 3.14 as the current major release).

Setting Up the Environment

- Downloading and installing Python.
- Choosing an editor or IDE (VS Code, PyCharm, Thonny).
- Running Python code interactively vs. script mode.

First Python Program

- Writing and running a "Hello World".
- Understanding errors and error messages.

2. 2. Python Syntax & Basics

- Comments and Code Style
- Variables and Assignment

  Basic Data Types

  - Numbers (integers, floats)
  - Strings and string methods
  - Booleans

  Operators

  - Arithmetic, comparison, logical operators
  - Order of operations

3. 3. Control Flow

Conditional Statements

- if, elif, else
- Nested and chained conditionals

Loops

- for loops and iteration
- while loops
- Loop control: break, continue, pass

4. 4. Core Data Structures

Sequences and Collections

- Lists — creation, indexing, slicing
- Tuples — immutability implications
- Sets — uniqueness and set operations
- Dictionaries — key/value access and iteration

Common Patterns

- Iterating over collections
- Comprehensions (list, dict)

5. 5. Functions and Modular Code

Defining Functions

- Parameters and return values
- Positional vs. keyword arguments

Variable Scope

- Local and global variables

- Built-in functions and creating reusable code

  Modules and imports

  - Using the standard library
  - Simple package organization

6. 6. Working with Files

- Opening and Closing Files
- Reading and Writing Text

Handling Exceptions

- Try/except blocks
- Cleaning up with finally

7. 7. Practical Debugging & Best Practices

Debugging Techniques

- Reading tracebacks
- Using print-based debugging

Code Style Basics

- Naming conventions
- Structuring small projects

© 2026 Faysal Aziz · Kuala Lumpur