Faysal Aziz

Python Prerequisites

Your Launchpad to Python-Driven Learning - 2 Days

FA-0393 · 2 Days

Software Development

Introduction

Python has become the universal language for problem-solving across domains like data science, artificial intelligence, web development, and beyond. Whether you're analyzing data, building machine learning models, or automating workflows, Python is often the go-to tool.

This course is designed to provide you with the foundational knowledge of Python so you can confidently transition into more advanced, domain-specific Python courses. With over 30 years of industry expertise, our instructor will teach you the most relevant and practical skills needed to thrive in today’s fast-paced learning environments.

Learning outcomes

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

- Understand Python’s basic syntax and structure.
- Write and execute Python scripts.
- Utilize essential Python data types, operators, and functions.
- Implement control flow structures like loops and conditional statements.
- Create and use Python functions effectively.
- Work with lists, dictionaries, and other collection data types.
- Debug simple Python programs.
- Prepare to learn more advanced topics in Python through real-world exercises.

Prerequisites

- Basic understanding of programming concepts (helpful but not mandatory).
- A laptop with Python 3 installed (instructions for setup will be provided).
- A Google Account (as we shall be using Google COlab as the IDE)
- Willingness to learn and experiment with hands-on exercises.

Training outline

1. 1. Introduction to Python and Programming Fundamentals

Overview of Python: Why Python is Popular

- Applications in industry (AI, web, data science, automation).

Setting Up the Python Environment

- Initializing Python and an IDE.
- Introduction to writing and running Python scripts.

Python Syntax Basics

- Writing clean and readable code (PEP 8 overview).
- Executing code in interactive and script modes.

2. 2. Core Data Types and Variables

Introduction to Variables

- Declaration and assignment of variables.
- Naming conventions and best practices.

Exploring Basic Data Types

- Numbers (integers, floats, and complex numbers).
- Strings: Manipulation and formatting.
- Booleans and logical operations.

Type Conversion and Input/Output

- Converting between data types.
- Using input() and print() effectively.

3. 3. Operators and Expressions

Arithmetic Operators

- Addition, subtraction, multiplication, division, modulus, and more.

Comparison and Logical Operators

- Greater than, less than, equal to, and logical connectives.

- Compound Statements and Operator Precedence

4. 4. Control Flow: Making Decisions

Conditional Statements

- if, elif, and else structures.
- Nested conditionals and decision trees.

Looping Constructs

- for loops: Iterating over ranges, lists, and dictionaries.
- while loops: Conditions and infinite loop prevention.

- Break, Continue, and Pass Statements

5. 5. Functions: Modular and Reusable Code

Defining and Calling Functions

- Syntax for defining functions.
- Using parameters and returning values.

Scope and Lifetime of Variables

- Local vs. global variables.

Lambda Functions

- Introduction to anonymous functions for simplicity.

6. 6. Working with Collections

Lists

- Creating, indexing, slicing, and modifying lists.
- List comprehensions for efficient operations.

Tuples and Sets

- Characteristics and use cases.
- Basic tuple operations.

Dictionaries

- Storing and accessing key-value pairs.
- Iterating through dictionaries.

7. 7. Error Handling and Debugging

Understanding and Handling Errors

- Common Python errors and exceptions.
- Using try, except, else, and finally.

Debugging Techniques

- Using print statements and Python debuggers.

8. 8. Practical Exercises and Applications

Writing a Small Python Program

- Hands-on implementation of learned concepts (e.g., a simple calculator or data summarizer).

Preparing for Advanced Python Courses

- Recommendations for further study.
- Insights into Python's broader ecosystem (libraries like NumPy, pandas, and matplotlib).

This fast-paced, two-day course is the perfect stepping stone to unlock Python's true potential in domain-specific applications. Get ready to build confidence and take on your Python journey with industry-grade skills!

© 2026 Faysal Aziz · Kuala Lumpur