# Course Abstract

In this 4-day course, participants establish a solid foundation in core Python programming language fundamentals using Python 3. Exercises cover common tasks in:

• data acquisition (via raw input from users, from files, from URLs, and from sockets)

• data parsing (including text, HTML, column-oriented, CSV and delimited files, JSON, and XML)

• data analysis (using list comprehensions, sets, dictionaries, and named tuples)

• data formatting (using templating to create HTML and other output formats)

Participants can expect to learn Python norms, idioms, and programming practices developing a clear mental model of the language (including magic methods, the object model, memory management, mutability, performance, and scaling, etc.). Participants will develop an expressive style that realizes the full advantages of the language becoming proficient at utilizing patterns of problem solving.

Topics include program organization, principles of object-oriented programming, testing, and debugging.

# Audience

This course is designed for sales engineers, security specialists, developers, quality assurance engineers, and anyone needing to use a powerful and clean scripting language.

# Prerequisites

The course assumes that participants have some prior experience with another programming language such as C++ or Java but assumes no prior knowledge of Python. Students should be familiar with:

* Basic concepts of programming such as variables, data types, statements, control-flow, functions, arrays, data structures, and common programming problems
* Concepts required for searching and sorting data structures
* Working with files, file systems, folders, editors, command shells, operating system environment settings, internet connections, and other essential operating system features

# Duration

4 Days

# Learning Outcomes

Upon completion of this course, participants will be able to:

* Develop Python solutions to interesting problems
* Establish a clear mental model of the language to prepare for more complex scripting solutions and applications

# Course Topics

**Day 1**

**Python Overview**

Business Level Overview

Language Philosophy

Python Implementations and Versions

Establishing the Python Environment

*Exercise: Environment Setup and Running Scripts*

Whitespace, Indentation and program formatting

Variables and Naming Conventions

Python Data Types

Coding Style Guides

Strings and Encoding Formats

Unicode

String Methods

Conversions

Introducing Sequences

Operators

Control Structures

Truthy/Falsey Values

*Exercise: Working with Strings*

*Exercise: Building a GREP Utility*

Lists

Copying and Sorting Lists

Advanced Iterating Techniques

Lambdas

*Exercise: Sorting and Data Structures*

List Comprehensions

*Exercise: List Comprehensions*

Creating Named Tuples

Sets

Dictionaries

*Exercise: Dictionaries*

**Day 2**

**Files and Flow Control**

Exception Handling

Working with Files

Using the 'with' Control

*Exercise: Files and Exceptions*

Print Formatting

Pip and Third-Party Tools

**Functions**

Default Arguments

Positional vs. Keyword Arguments

Multiple Positional/Keyword Arguments

Creating Modules

Rules for Creating Modules

*Exercise: Intermediate Functions and Modules*

Variable Scope

Argument Passing

**Day 3**

**Object-Oriented Python**

Creating Objects

Constructors

Using self

Magic Methods

Properties

*Exercise: Creating Classes*

Inheritance Basics

Class and Static Methods

**Modules of the Python Standard Library**

sys, os,

os.path, os.stat

os.walk

shutil for Copying Files and Directories

time, datetime Modules

csv Module

Docstrings and Comments

Exercise: Walking Directories and Copying Files

pathlib

Exercise: Working with pathlib.Path

Parsing XML with ElementTree

Creating XML with ElementTree

Exercise: Using ElementTree

Reading YAML Files with PyYAML

Creating YAML Files with PyYAML

**Day 4**

**Network Programming**

Creating TCP clients and servers using Sockets

Mid-level Networking Modules: ftplib, http.client

urllib Modules

Introducing the Requests Module

Requests Module Example

Handling Request Errors

Python Match-Case Control

BeautifulSoup

Scraping Content from HTML with Requests and BeautifulSoup

Using the JSON Module

Processing JSON Data

*Exercise: Parsing JSON data and screen scraping*

**Search and Regular Expressions**

Using the re Module

MatchObjects

Matching Flags

String Replacement and Modification

*Exercise: Working with Regexes*

# Course Code

IN1467