Python Automation Bootcamp Customized

(40 hours)

Course Description

The "Automation with Python" course is a comprehensive program that focuses on leveraging the power of Python to automate repetitive tasks and enhance productivity. Through hands-on projects and practical examples, participants will learn various automation techniques, including input validation, file manipulation, web scraping, working with different data formats, and GUI automation. This course equips learners with the necessary skills to automate workflows, save time, and optimize efficiency using Python.

Audience

This course is suitable for:

- Python developers who want to enhance their automation skills and streamline their workflows.
- IT professionals and system administrators seeking to automate routine tasks and improve productivity.
- Data analysts and researchers looking to automate data processing and analysis workflows.
- Professionals from various domains who want to leverage Python automation for increased efficiency.
- Individuals with basic Python knowledge who are interested in learning automation techniques.

Pre-requisite Knowledge/Skills

Participants should have:

- Basic understanding of Python programming, including syntax, data types, and control flow.
- Familiarity with Python libraries and modules.
- Proficiency in writing and executing Python scripts.

Course Objectives

The objective of this course is to provide participants with the knowledge and skills required to automate tasks using Python. By the end of the course, learners will be able to design and implement automation scripts, effectively utilize Python libraries for automation, and optimize workflows to save time and increase productivity.

Course Outline

The course comprises 40-hours of theory and labs.

Module 1: Python Basics

- Entering Expressions into the Interactive Shell
- The Integer, Floating-Point, and String Data Types
- String Concatenation and Replication
- Storing Values in Variables
- Your First Program
- Dissecting Your Program
- The `print()` Function
- The `input()` Function
- The `len()` Function
- The `str()`, `int()`, and `float()` Functions

Module 2: Flow Control

- Boolean Values
- Comparison Operators
- Elements of Flow Control
- Flow Control Statements
- `break` and `continue` Statements
- `for` Loops and `range()` Function
- Importing Modules
- Demonstration Programs

Module 3: Functions

- `def` Statements with Parameters
- `def`, `call`, `pass`, Arguments, Parameters
- Return Values and Return Statements
- The None Value
- Keyword Arguments and the `print()` Function
- The Call Stack
- Local and Global Scope
- The global Statement
- Exception Handling

- Program Demonstration

Module 4: Lists

- The List Data Type
- Getting Individual Values in a List with Indexes
- Changing Values in a List
- List Concatenation and List Replication
- Working with Lists
- Augmented Assignment Operators
- Methods in List
- Sequence Data Types
- Mutable and Immutable Data Types
- The Tuple Data Type
- Passing References
- Program Demonstration

Module 5: Dictionaries and Structuring Data

- The Dictionary Data Type
- `keys()`, `values()`, `items()` Methods
- `get()` Method
- Pretty Printing (pprint) Module
- Using Data Structures to Model Real-World Cases
- Nested Dictionaries and Lists

Module 6: Manipulating Strings

- Working with Strings
- Escape Characters
- Indexing and Slicing Strings
- Useful String Methods
- Using `ord()` and `chr()` Functions
- Program Demonstration

Module 7: Pattern Matching and Regular Expressions

- Finding Patterns of Text Without Regular Expressions
- Finding Patterns of Text with Regular Expressions
- More Pattern Matching

- Character Class
- Wild Card Character
- Using `sub()` Method
- Managing Complex Regexes
- Program Demonstration

Module 8: Input Validation

- The PyInputPlus Module
- The `min`, `max`, `greaterThan`, and `lessThan` Keyword Arguments
- `limit`, `timeout`, and `default` Keyword Arguments
- The `allowRegexes` and `blockRegexes` Keyword Arguments
- Passing a Custom Validation Function to `inputCustom()`
- Program Demonstration

Module 9: Reading and Writing Files

- Files and File Paths
- Absolute and Relative Paths
- Modifying Lists of Files Using Glob Patterns
- File Reading/Writing Process
- Saving Variables with Shelve Module
- Using `pprint.pformat()` Function
- Program Demonstration

Module 10: Debugging

- Raising Exceptions
- Assertions
- Logging
- Mu's Debugger
- Program Demonstration

Module 11: Web Scraping

- `MapIT.py` Module
- Working with HTML
- Using Developer Tools to Find HTML Elements
- Parsing HTML Using BS4 Module
- Program Demonstrations

Module 12: Working with Excel Spreadsheets

- Installing OpenPyXL Module
- Opening and Reading Excel Documents
- Working with Columns and Rows
- Program Demonstration

Module 13: Working with CSV Files and JSON Data

- CSV Module
- Reader Objects
- Reading Data from Reader Objects in a For Loop
- Writer Objects
- DictReader and DictWriter CSV Objects
- Program Demonstration
- JSON and APIs
- JSON Module
- Reading and Writing JSON Using Functions
- Program Description

Module 14: Timekeeping, Task Organization, and Program Starting

- Time Module
- Programs Using Functions of the Time Module
- DateTime Module
- Time Delta Data Type
- Converting DateTime Objects into Strings
- Converting Strings into DateTime Objects
- Review of Python Time Functions
- Multithreading
- Launching Other Programs from Python
- Program Demonstration

Module 15: Sending Email and Text Messages

- Sending and Receiving Email with the Gmail API
- SMTP
- Connecting to an SMTP Server
- Sending SMTP Messages
- IMAP

- Retrieving and Deleting Emails with IMAP
- Connecting to IMAP Server
- Performing Search
- Working with Body and Source of an Email
- Sending Text Messages with SMS Email Gateways
- Sending Text Messages with Twilio
- Program Demonstrations

Module 16: Efficient GUI Automation - Streamlining Tasks with Keyboard and Mouse Control

- Installing PyAutoGUI Module
- Controlling Mouse Movements
- Controlling Mouse Interaction
- Planning Your Mouse Movements
- Working with the Screen
- Image Recognition
- Getting Window Information
- Controlling the Keyboard
- Setting Up Your GUI Automation Scripts
- Review of PyAutoGUI Functions
- Program Demonstrations
- Displaying Message Boxes
- Program Demonstration

Module 17: Python and REST APIs

- REST Architecture
- REST APIs and Web Services
- Consuming APIs and Building APIs
- API Integration - Flask

Module 18: OS-Services

- The OS Module
- Environment Variables
- External Commands
- Directory Trees
- Running Unix Commands in Your Python Program