Objective: "This training will enable participants to use, program & customise python scripts. This course does not need any basic knowledge of Python and all the fundamentals will be covered during the training, gradually taking the learning to a deeper level. Apart from taking a deep dive into core python concepts, this training will also enrich the participants with decent knowledge of using python for regular scripting, Automation, regular expressions, threads, object oriented concepts and interfacing various kinds of meta data files. The latter half of training is dedicated to building APIs, working with frameworks, writing test cases, performing unit tests and API tests.

Prerequisite

basic Python knowledge

Lab Setup

Hardware

PC/Laptop

>= 8gb Ram

>= 20gb free hard disk space

Internet Access (mandatory)

Software Windows or Linux or Mac machines with any python IDE

if any python IDE not installed, then install the following:

python:

<https://www.python.org/downloads/>

Some packages may be needed to install during the course, which will need an active internet connection and installation permissions.

**Table of Contents:**

The training cannot be strictly followed as per the session-wise break up, and trainer will have to introduce some advance concepts early on, as most topics are inter-related. Trainer will pace up & slow down based on speed of participants.

**Day 1**

Variables & Data

Syntax

Variables

Data types

Naming Rules

Multiple assignment

del & none

builtins

identifiers

basic operations

Operators

bool type

Arithmetic

Relational

Assignment

Special Functions

type()

str() int() float()

input() complex()

eval()

String Operations

String Operations

access & deletion

Slicing & Indexing

negative index, out-of-bounds

escape sequences

important string functions

Data Structures: Lists & Tuples

declaration

access

negative index

slicing

immutability

delete item, list

reassign

extend, append, add

tuple packing

multidimensions

built-in functions

sort vs sorted

**Day 2**

Data Structures: Sets

creation

unordered

duplicate elements

access

mutability and immutability

remove vs discard

multidimensions

other built-in functions

Data Structures: Dictionaries

coupling data, key-value

creation

unordered

duplicate values

access

mutability and immutability

nested data structures

multidimensions

other built-in functions

Memory & namespaces

internal layout

dynamism

python layer (interpreter)

id()

namespace:

built-in, module, function

mutability

Decision making

For Loop

range & enumerate

While Loop

Break Statement

Next Statements

Repeat Statement

if, if…else Statements

nested

Operators: continued

Logical

Membership

Identity

Bitwise

**Day 3**

Functions in Python

user defined functions

Calling Python Functions

Functions with Arguments

return

default arguments

named arguments

global, local

multiple definitions

Using Modules

The Import Statement

Search Path

important libraries & modules

datetime, os, sys, time

creating custom modules

importing parts vs whole

File Handling

Files and File Paths

os.makedirs() & os.path

split, size, listdir etc

open close

read write

**Day 4**

Object Oriented Programming

Introduction to Python Classes

concept of OOP

defining classes

methods

object creation

self

\_\_init\_\_

encapsulation

\_\_private\_\_

OOP: inheritance

multilevel

multiple

super

documentation

docstrings

\_\_doc\_\_

multiple documentation

**Day 5**

File Formats

XML

What is XML

Python XML Parsing Modules

xml.etree.ElementTree Module

root, tags & tree

parse

extract information

Finding Elements

Modifying XML files

Adding to XML

Deleting from XML

JSON

what is JSON

python json module

load & dump

loads & dumps

Regular Expression

RE Objects

Pattern matching

Parsing data

expressions

simple substitutions

searching & finding

Multi-Threading

concurrent execution

threads & processes

Thread-Local Data

Thread Objects

Lock Objects

miscellaneous

map and filter

lambda functions

generator

**Day 6**

Web interfaces

Requests

Post & Get

initiating communication with python

response

extracting information

Frameworks

Flask

Setting up

virtual env

Templates

Web Forms

Data Modelling

Database

User Logins

Error Handling

RESTful API Building

what is an API

class-based REST interface

extension-based REST interface

complete RESTful API

Debugging, Error Handling, and Testing

**Day 7**

Testing in Python

Different frameworks

dependencies

pytest, unittest, nose

UnitTest

Introduction

setup() teardown()

testsuite class

Assertion

assert class

various asserts

Test Discovery

Skip Test

Exceptions Test

Time Test

**Day 8**

API testing

building an API

corner cases

Asserts

automating tests

Hardware interface with Python (briefly)

Bluetooth

USB

PyUSB

communication handshake

sending & receiving data

mini project

cumulation of most topics