Course Title: Python for Network Engineer Duration: 5 days

Program objectives

Some of the key skills you will gain upon completion of this program include:

- Basic Language Syntax
- Object Oriented Features in Python
- Exception Handling
- Regular Expression
- Testing
- Textfsm
- Virtual environment
- HTML parsing
- XML
- JSON
- DevOps
- Continuous Integration tool, Jenkins / GitLab CI
- Software Configuration Management)

Prerequisite

This course is designed for the Intermediate to Advance Level. The participants need not have any prior exposure to Python programming language. Prior familiarity with some other programming language (such as Java or C++) would be useful, but it is not mandatory.

Audience

This training is suitable for all testers and programmers who want to get deep hands-on experience of Robot Framework and Selenium Automation Tester.
The training is generic and useful for people working in any domain. The application under test in the hands-on exercises is web based, and participants thus get immediately useful skills for web test automation as a by-product.

Prerequisite

This course is for Intermediate to Expert Level Developers. The participants should have prior exposure to Basic Python programming language.

Set up Requirements

Computer with the following software
Operating System: Unix OS Ubuntu 16.04 / CentOS 7.0/ Window XP/Vista/7/8/10
python 3.7 on Anaconda Notebook Tools

Sublime3 Text Editor (Optional)
PyCharm Community Edition
Python Library selenium library
Web Browser (Chrome / Firefox)

Note: Training Session include 90 % Hands on Session and 10% Interactive Discussion

Day Wise Break Up

Day	Module	Topics
Day 1	Module 1	Introduction to Python Dynamic Typing, Object Types Complex Object Type Operators Unbounded Integers Useful functions type() id() dir() help() chr() unichr()
	Module 2	Simple Program Using Basic Python Programming with Python 3.x Pycharm Community IDE Installation and Configuration
	Module 3	Basic Language Construct ▪ Data types and Variables ▪ String type ▪ Format method ▪ Operators and Expressions ▪ Indentation
	Module 4	Data Structures Mutable and Immutable Data Structures 1) List, Subscripting, Nested List 2) Tuple, Use cases 3) String Manipulation 4) Dictionary with Case Study 5) Use Cases and Assignment

	Module 5	Control Structure 1) Indentation 2) if elif else 3) while 4) for (nested) 5) Use Cases and Assignments **Assignments and Group Activities**
Day 2	Module 6	Functions User Define Functions 1) global variable 2) default arguments 3) variable arguments *arg 4) Multiple Variable Default Argument 5) **kwarg 6) Use Case Design Multiplier Sequence Operation using a) lambda b) filter c) map d) reduce e) sum/max/min f) set g) enumerate h) sorted i) reversed j) range/ xrange Operation Using 1) List /Tuple Comprehension 2) Dictionary Comprehension 3) Dictionary Use Case

	Module 7	Modules User Define Modules Import Categories 1) using import 2) using from Built In Modules 1) math 2) os 3) sys 4) random 5) pickle / Unpickle (Object Serialization)
Day 3	Module 8	Object Oriented Programming ▪ Classes and Objects ▪ The “self” keyword ▪ Methods and Attributes ▪ Constructor and Destructor ▪ Instance and static member ▪ Class Inheritance ▪ Built In Attributes ▪ __private ▪ public ▪ _protected ▪ Multiple Inheritance ▪ Super keyword **Assignments and Group Activities**
	Module 9	Files Objects and Methods open() read(), readlines() write(), writelines() tell() using with statements Use Case using File Handling
	Module 10	Exception Handling Built in Exceptions exceptions module User Define Exceptions

Day 4	Module 11	**How to use REGEX and TEXTFSM** Regular expressions Pattern Writing Compiling Match/Search Group/Groups findall re.sub re.split Hands on Session. TEXTFSM Installation Using textfsm **Assignments and Group Activities**
	Module 12	**Virtual environment** Coding Standards (PEP 8). Self-documenting code (Pydoc). Best practices of programming; Writing efficient code **Unit Testing** Testing Fundamental Types of Testing Unittest Framework Run Test Write Unittest.TestCase for Python Code
	Module 13	HTTP/HTTPS calls – What it is used for? What is certification/authentication How to use HTTP library in python; examples of HTTP/HTTPS endpoint calls. Encoding/decoding formats JSON, XML – Overview Uses of JSON / XML using python Serialisation YAML / JINJA– What/how/why do we use. **Assignments and Group Activities**

Day 5	Module 14	What is Devops? What is Devops as a culture? API: Introduction to different API (OPENAPI, RESTAPI) Tools - POSTMAN, CURL (demonstration of how to use these tools to test and debug API endpoints) Overview: CRUD actions for Client-server comms; Exploring DevOps Why Does DevOps Matter? Core DevOps Principles The Three Ways The First Way The Theory of Constraints The Second Way The Third Way Chaos Engineering Learning Organizations Key DevOps Practices Continuous Delivery Site Reliability & Resilience Engineering **Assignments and Group Activities**

	Module 15	Encoding/decoding formats: JSON, XML – Overview and why is this required; How you verify? How to use JSON / XML Serialisation: YAML / JINJA– What/how/why do we use. How to use REGEX and TEXTFSM
	Module 16	CI – Overview and demonstrate on how to use CI eg . JENKINS, GITLAB Cis; Explain and how to use LINTER CD/CT SCM (Software Configuration Management) Overview and Branching Strategies. Contrast between GIT/GIT Lab/GIT Hub; Demonstrations with Git Lab OPS – Monitoring or tool chains, notifications of built. **Assignments and Group Activities**
	Module 17	**Group Projects and Presentations** **Summary for Five Days**