Python programming for Network Automation(5 days)

By Dr. Vishwanath Rao

PSession 1: Introduction to Python Basics and Networking Fundamentals
Objective: Introduce Python basics and provide an overview of networking libraries essential for network automation.

Topics:

1. Overview of Python:

Introduction to Python as a versatile and powerful programming language.
Syntax fundamentals: indentation, comments, and basic structure of Python code.
Variables, data types (integers, floats, strings, booleans), and basic operations (arithmetic, comparison, logical).

2. Loops and Conditionals:

Explanation of loops (for and while) for iterating over sequences and performing repetitive tasks.
Introduction to conditional statements (if, elif, else) for decision-making in Python programs.

3. Functions (Routines):

Understanding the concept of functions as reusable blocks of code.
Declaring and defining functions, passing arguments, and returning values.
Importance of modular programming and code organization.

4. Error Handling (Try-Except):

Introduction to error handling using try-except blocks for handling exceptions gracefully.
Handling common types of exceptions and providing meaningful error messages.
Best practices for error handling in Python programs.

5. Setting up Python Environment:

Installation of Python on various platforms (Windows, macOS, Linux).
Configuring Integrated Development Environment (IDE) for Python development.
Setting up Python interpreter and project structure.

6. Basic Python Programming Exercises:

Hands-on exercises to reinforce Python basics, loops, routines, and error handling:
Writing Python scripts to perform arithmetic operations, manipulate strings, and use boolean expressions.
Implementing loops and conditional statements for solving problems.
Creating and calling functions to encapsulate reusable code.
Using try-except blocks to handle potential errors gracefully.

7. Introduction to Networking Libraries: Paramiko and Netmiko:

Overview of Paramiko library: its features, capabilities, and role in SSH

communication.
Introduction to Netmiko library: purpose, supported platforms, and advantages for network automation.
Installation of Paramiko and Netmiko libraries using pip.
Brief demonstration of how Paramiko and Netmiko facilitate SSH-based communication with network devices.

Hands-on Activity:
1. Python Environment Setup:
Walkthrough of installing Python and IDE.
Setting up a Python project with a basic file structure.
2. Basic Python Exercises:
Interactive coding session to practice Python fundamentals, including loops, functions, and error handling.
Writing and running Python scripts to perform various operations and solve simple problems.
3. Paramiko and Netmiko Setup:
Installing Paramiko and Netmiko libraries using pip.
Verifying installation and checking library versions.
4. Simple Paramiko/Netmiko Script:
Writing a simple Python script using Paramiko or Netmiko to establish an SSH connection to a network device.

Session 2: Working with Paramiko for SSH Communication
Objective: Understand the Paramiko library for SSH communication with network devices.
Topics:
1. Introduction to Paramiko Library:
Overview of Paramiko library: a Python implementation of SSHv2 protocol for secure communication.
Features and capabilities of Paramiko for SSH connection management and remote command execution.
2. Establishing SSH Connections to Network Devices:
Explanation of SSH (Secure Shell) protocol for secure communication over networks.
Steps to establish SSH connections using Paramiko:
Importing Paramiko library.
Creating SSH client instance.
Providing authentication credentials (username, password, or SSH keys).
Connecting to network devices using IP addresses or hostnames.
3. Executing Show Commands and Capturing Output:
Understanding the concept of executing commands remotely on network devices.
Using Paramiko to execute show commands and retrieve output from network devices.

Parsing and processing command output for further analysis or automation tasks.
Hands-on Activity:
1. Writing Python Scripts to Establish SSH Connections using Paramiko:
Setting up a Python script environment with necessary imports.
Writing Python code to create an SSH client instance using Paramiko.
Providing authentication credentials (username, password, or SSH keys) within the script.
Establishing SSH connections to network devices specified by IP addresses or hostnames.
2. Executing Show Commands on Network Devices and Capturing Output:
Developing Python scripts to execute show commands (e.g., show version, show interface) on network devices.
Capturing command output using Paramiko's exec_command() method and processing the output.
Displaying or storing the command output for further analysis or automation tasks.

Session 3: Automating Network Configuration with Netmiko / Paramiko
Objective: Learn how to automate network device configuration using the Netmiko library.
Topics:
1. Introduction to Netmiko/Paramiko Library for Network Automation:
Overview of Netmiko: an open-source Python library built on top of Paramiko for network automation tasks.
Features and advantages of using Netmiko for automating network device configurations.
Comparison between Netmiko and Paramiko for network automation tasks.
2. Pushing Configurations to Network Devices:
Understanding the concept of configuration management in network automation.
Using Netmiko to push configurations (e.g., setting up VLANs, configuring interfaces) to network devices.
Techniques for applying configuration changes safely and efficiently across multiple devices.
3. Error Handling and Exception Management:
- Importance of error handling in network automation scripts for robustness and reliability.

Common errors and exceptions encountered during network automation tasks.
Implementing try-except blocks and error handling mechanisms to handle exceptions gracefully.
Hands-on Activity:
1. Writing Python Scripts to Push Configuration Changes using Netmiko:
Setting up a Python script environment with necessary imports for Netmiko.
Writing Python code to establish SSH connections to network devices using

Netmiko.
Implementing functions to push configuration changes (e.g., adding VLANs, configuring interfaces) to network devices.
Executing the script to push configuration changes and verifying the results.
2.	Implementing Error Handling Mechanisms for Robust Automation Scripts:
Enhancing Python scripts with try-except blocks to handle potential errors during SSH connections and configuration changes.
Implementing error logging to capture and report errors encountered during script execution.
Testing the error handling mechanisms by deliberately inducing errors and observing script behaviour.

Session 4: Data Handling and Manipulation, Introduction to Regular Expressions (Regex) for Parsing Command Output
Objective: Understand how to read data from various sources and use regular expressions for parsing command output in network device programming.
Topics:
1.	Reading Data from Various Sources:
Overview of data sources commonly used in network device programming: CSV, Excel, text files, and notepad.
Techniques for reading data from different file formats using Python libraries (e.g., csv, pandas).
Connecting to and accessing data from SQL Server databases using Python libraries (e.g., pyodbc).
2.	Introduction to Regular Expressions (Regex):
Explanation of regular expressions as a powerful tool for pattern matching and text processing.
Syntax and usage of regular expressions in Python for searching, matching, and extracting text patterns.
Common metacharacters and special sequences used in regular expressions.
Hands-on Activity:
1.	Reading Data from CSV, Excel, Text Files, and Notepad:
Writing Python scripts to read data from CSV, Excel, text files, and notepad using appropriate Python libraries (csv, pandas, io).
Demonstrating techniques to handle different file formats and extract relevant data for network device programming tasks.
2.	Connecting to a SQL Server Database and Executing Queries:
Setting up a connection to a SQL Server database using Python libraries (pyodbc).
Executing SQL queries to retrieve data from tables or views in the database.
Processing and manipulating SQL query results for further analysis or integration with network device programming scripts.
3.	Introduction to Regular Expressions (Regex):
Providing an overview of regular expression syntax and components.
Demonstrating basic regular expression patterns and their usage in Python.

Discussing real-world scenarios where regular expressions are useful, such as parsing command output from network devices.

4. Parsing Command Output using Regex:

Writing Python scripts to extract specific information from command output using regular expressions.

Identifying patterns and using regex to parse command output into structured data for further processing or analysis.