

Practical Network Programming in Python (PNPP)

COURSE OVERVIEW:

This 5-day course balances lecture and lab exercises to ensure students can apply key Python programming knowledge and skills required to address network automation challenges in a multi-vendor infrastructure.

WHO WILL BENEFIT FROM THIS COURSE?

This course is designed for network architects, network engineers and application developers that have experience designing, building and maintaining enterprise and/or service provider networks or building network automation applications.

PREREQUISITES:

Network engineers participating in this class should have a basic understanding of Python or other scripting/programming languages and working experience with enterprise network infrastructure equipment. Application developers participating in this class should have a basic understanding of networking equipment and their functions on the network.

Recommended pre-course reading or training for network engineers include:

- [Automate the Boring Stuff with Python](#)
- [Python Programming Course](#)

Pre-course work for Application Developers include ICND1 level knowledge or equivalent:

- [ICND1](#)

COURSE OBJECTIVES:

After completion of this course, students will be able to...

- Explain what Python is and why it is the language of choice for network automation
- Describe how a Python script executes
- Explain data types, application flow control, and libraries
- Use an Integrated Development Environment (IDE) to develop, debug and run scripts
- Use different techniques to capture and use network device output from a variety of equipment
- Apply configuration via multiple libraries
- Validate that executed scripts achieve the desired network functionality
- Demonstrate a programmatic way of planning and implementing a new network service
- Build custom libraries for frequently used functions
- Apply learned skills to data center services including routing, switching, security and more
- Implement techniques for managing code and applying version control
- Apply Continuous Improvement Continuous Development (CICD) for network engineers
- Prepares engineers for hybrid cloud deployments

COURSE OUTLINE:

MODULE 1 – INTRO TO PYTHON

Lesson 1: Introduction to Python

In this lesson students will review programming basics, learn the fundamentals of Python and examine its code structure and program flows. They will also learn how libraries are used to communicate with networking equipment.

Lesson Objectives:

After completing this lesson, the student will be able to:

- Demonstrate basic programming knowledge
- Use the PyCharm IDE to create, edit, execute and debug and manage Python applications
- Describe Python code flow and execution
- List various network programming libraries and describe how to use them to communicate with various network devices through a Python application
- Examine network device configuration from a Python perspective

Day 1

Introduction to Python and Programming Languages

Interpreted vs Compiled Languages

How a script executes

IDE – Intro to PyCharm Community Edition

Creating a project

Editing code, code completion

Running a program

breakpoints and debugging

version control – Gitlab

Data-types in Python

Numerics, Strings, Lists, Tuples, Dictionaries

Flow control

for loops

if/then/elif

while loops

Network Programming Library Overview

sshpass, pexpect and calling command line tools from python

netmiko

Yang Explained

Netconf, ncclient

Restconf

Lesson 2: Exploring Network Device Data Formats:

This lesson focuses on communicating with network devices and the various formats of data produced through interactions with those pieces of networking equipment. Through this exploration students will become comfortable with working with network programming libraries.

Lesson Objectives:

After completing this lesson, the student will be able to:

- Issue commands and Inspect/manipulate varied network device output using Python libraries
- Describe the type of output provided by different network programming libraries
- Use the IDE to examine capabilities and limitations of various network programming libraries
- Use the IDE debugger to step through Python code to discover how to communicate with a device programmatically

Day 2

Modeling Network Configuration CLI in python

Templates in Jinja2/Yaml

Configuration Management

Auditing and State Validation

JSON vs XML vs raw CLI

'show command' parsing

python string operations

TextFSM

Genie parser

json-pretty filter

Device Capabilities:

Cisco: IOS, XE, XR, NXOS (all different approaches)

YDK, Yang Explorer

pyATS

Juniper JUNOS:

unified library: pyez

Arista EOS:

unified library: pyeapi

Device Real-time Interaction and debugging

netmiko programming

Nexus NX-API

IOS-XE Rest API

MODULE 2 – NETWORK OPERATIONS AUTOMATION

Lesson 1: Network Device Orchestration

In this lesson students will leverage their strong understanding of the network from an architectural view and how to use a services-model approach to plan, automate and validate network functionality. Students will learn how to think like a programmer when addressing network configuration challenges; this insight will also help improve communications in environments where their developers and network engineers exist in separate teams.

Lesson Objectives:

After completing this lesson, the student will be able to:

- Approach network configuration from a programmer's perspective
- Use a service-chaining approach to building applications
- Validate and audit operational state and configurations ensuring performance as expected

Day 3

Intro to device orchestration
Environment and command line variables and Python
Code re-use/abstraction
Working with a Datacenter Design
Service chain definition – enabling a new application launch in the datacenter
vlan management automation
Layer 2 validation
interface management automation
Layer 3 validation
BGP/OSPF/Routing configuration automation
Routing and Protocol validation
Security Automation
ACL Validation
CDP/LLDP Validation

Lesson 2: Configuration, Validation and Auditing:

In this lesson students will create/modify live device configurations. Topics covered include non-destructive configuration modifications, configuration management, sequential vs. parallel execution and other best practices. Students will apply what they learned in the previous model to validate and audit their automated network configuration changes.

Lesson Objectives:

After completing this lesson, the student will be able to:

- Build and modify device configurations
- Validate and audit applied configurations and changes
- Plan for practical deployments and assess dependencies and scaling requirements
- Turn large configuration files into a database structure for querying

Day 4

Multi Device and Dependency Management – Sequencing
Full configuration management
Jinja2 / YAML config creation
artifact storage
configlets (Jinja)
device state templates (Genie, TextFSM)
CMDB – Device Inventory
Data based python scripting
SQLite

Lesson 3: Network Automation Ecosystems:

In this lesson students will explore the principles of Continuous Integration/Continuous Development (CI/CD) for the purposes of version control and pipeline execution. Additionally, students will learn to use the RobotFramework to drive their custom automation solutions and to call 3rd party Python libraries, import their own custom libraries, and leverage its powerful validation, auditing and reporting functionality.

Lesson Objectives:

After completing this lesson, the student will be able to:

- Customize their automation toolset to address their specific device mix and functional needs
- Use RobotFramework to call Python libraries and audit network functionality
- Discuss how they can apply the solutions developed in this course to automation frameworks beyond RobotFramework such as Ansible.

Day 5

Gitlab – CI/CD Introduction
Version Control
Pipeline execution

RobotFramework

Why Robot - a look at the automation ecosystems available
Ansible, Puppet, Chef, Salt
Robot scripts and keywords
Calling 3rd party and custom python libraries from Robot scripts
Combining:
Configuration Management
Health Checks
Auditing and Reporting

SUNSET LEARNING INSTITUTE (SLI) DIFFERENTIATORS:

Sunset Learning Institute (SLI) has been an innovative leader in developing and delivering authorized technical training since 1996. Our goal is to help our customers optimize their cloud technology investments by providing convenient, high quality technical training that our customers can rely on. We empower students to master their desired technologies for their unique environments.

What sets SLI apart is not only our immense selection of trainings options, but our convenient and consistent delivery system. No matter how complex your environment is or where you are located, SLI is sure to have a training solution that you can count on!

Premiere World Class Instruction Team

- All SLI instructors have a four-year technical degree, instructor level certifications and field consulting work experience.
- Sunset Learning has won numerous Instructor Excellence and Instructor Quality Distinction awards since 2012

Enhanced Learning Experience

- The goal of our instructors during class is ensure students understand the material, guide them through our labs and encourage questions and interactive discussions.

Convenient and Reliable Training Experience

- You have the option to attend classes at any of our established training facilities or from the convenience of your home or office with the use of our HD-ILT network (High Definition Instructor Led Training)
- All Sunset Learning Institute classes are guaranteed to run – you can count on us to deliver the training you need when you need it!

Outstanding Customer Service

- Dedicated account manager to suggest the optimal learning path for you and your team
- Enthusiastic Student Services team available to answer any questions and ensure a quality training experience