

## Programming for Network Engineers

**Duration: 4 Days**    **Course Code: PRNE**    **Version: 2.0**    **Delivery Method: Virtual Learning**

### Overview:

The Programming for Network Engineers (PRNE) course is designed to equip you with fundamental skills in Python programming. Through a combination of lectures and lab experience in simulated network environments, you will learn to use Python basics to create useful and practical scripts with Netmiko to retrieve data and configure network devices. Upon completion of this course, you should have a basic understanding of Python, including the knowledge to create, apply, and troubleshoot simple network automation scripts.

#### **This course will help you:**

- Explain the need for network engineers to learn how to program
- Explain how programming relates to the journey into network automation and programmability
- Create useful and practical scripts to retrieve data and configure network devices
- Create, apply, and troubleshoot simple network automation scripts
- Gain hands-on experience with Python programming

### Virtual Learning

This interactive training can be taken from any location, your office or home and is delivered by a trainer. This training does not have any delegates in the class with the instructor, since all delegates are virtually connected. Virtual delegates do not travel to this course, Global Knowledge will send you all the information needed before the start of the course and you can test the logins.

### Target Audience:

Networking individuals looking to understand the fundamentals of Python programming which is essential for today's software defined networks.

### Objectives:

- |                                                                                    |                                                                        |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| ■ <b>After completing this course, you should be able to:</b>                      | ■ Describe the options for Python data manipulation and storage        |
| ■ Create a Python script                                                           | ■ Describe Python modules and packages, their uses, and their benefits |
| ■ Describe data types commonly used in Python coding                               | ■ Explain how to manipulate user input in Python                       |
| ■ Describe Python strings and their use cases                                      | ■ Describe error and exception management in Python                    |
| ■ Describe Python loops, conditionals, operators, and their purposes and use cases | ■ Describe Python code debugging methods                               |
| ■ Describe Python classes, methods, functions, namespaces, and scopes              |                                                                        |

### Prerequisites:

#### **Attendees should meet the following prerequisites:**

- Familiarity with Cisco IOS®-XE software or other Cisco network device configuration and operation skills
- Basic network management knowledge
- CCNA - Implementing and Administering Cisco Solutions

### Testing and Certification

#### **Recommended as preparation for the following exams:**

- There is no exam currently aligned to this course

### Follow-on-Courses:

- ENAUI - Implementing Automation for Cisco Enterprise Solutions

## Content:

### Introducing Programmability and Python for Network Engineers

- Programmability
- Continued Importance of the CLI
- Need for Python
- First Python Program

### Scripting with Python

- Running a Python Script
- Python Interactive Shell
- Editors and IDEs

### Examining Python Data Types

- String Data Type
- Integer Data Type
- Float Data Type
- Boolean Data Type
- type() Function
- Python Variables
- List Analysis
- Tuple Data Analysis
- Data Set Types
- Python Dictionary
- Nested Data Types
- Navigating Complex Data Structures

### Manipulating Strings

- String Manipulation
- String Splitting
- Modifying Strings
- String Concatenation
- Whitespace Stripping
- Formatting and Templating
- Escape Characters
- Regular Expressions
- String Methods

### Describing Conditionals, Loops, and Operators

- Whitespace and PEP 8
- Conditionals
- Operators
- Example of Using Conditionals
- Loops
- Loops with Lists, Dicts and Ranges

### Exploring Classes, Methods, Functions, Namespaces, and Scopes

- Practical Reuse of Codes
- Functions
- Code Commenting
- Namespaces and Scopes
- Classes and Methods
- Using Inheritance to Extend Functionality
- Main Construct

### Exploring Data Storage Options

- Data Formats
- Reading Data from a Simple Text and CSV File
- Writing Data to a Simple Text and CSV File
- Reading Data from a JSON Text File
- Writing Data to a JSON Text File
- Data Access in a RAW or Unstructured Format

### Exploring Python Modules and Packages

- Python Standard Library Modules
- Reusable code with Modules and Packages

### Gathering and Validating User Input

- Obtaining User Input
- Command-Line Arguments

### Analyzing Exceptions and Error Management

- Errors and Exceptions
- Managing Exceptions
- Assertions

### Examining Debugging Methods

- Code Debugging Process
- Python Debugging Functions
- Debuggers
- Python Debugger
- Pdb Commands

### Labs

- Execute Your First Python Program
- Use the Python Interactive Shell
- Explore Foundation Python Data Types
- Explore Complex Python Data Types
- Use Standard String Operations
- Use Basic Pattern Matching
- Reformat MAC Addresses
- Use the if-else Construct
- Use for Loops
- Use while Loops
- Create and Use Functions
- Create and Use Classes
- Use the Python main() Construct
- Traverse the File Structure
- Read Data in Comma-Separated Values (CSV) Format
- Read, Store, and Retrieve Data in XML Format
- Read, Store, and Retrieve Date in JavaScript Object Notation (JSON) Format
- Read, Store, and Retrieve Data in a Raw or Unstructured Format
- Import Modules from the Python Standard Library
- Import External Libraries
- Create a Python Module
- Prompt the User for Input
- Use Command-Line Arguments
- Manage Exceptions with the try-except Structure
- Manage Exceptions with the try-except-finally Structure
- Use Assertions
- Use Simple Debugging Methods
- Use the Python Debugger
- Code a Practical Debugging Script

### Further Information:

For More information, or to book your course, please call us on 00 971 4 446 4987

[training@globalknowledge.ae](mailto:training@globalknowledge.ae)

[www.globalknowledge.com/en-ae/](http://www.globalknowledge.com/en-ae/)

Global Knowledge, Dubai Knowledge Village, Block 2A, First Floor, Office F68, Dubai, UAE