

Programming for Network Engineers (PRNE)

Programming for Network Engineers (PRNE)

In this Programming for Network Engineers course provides a structured, hands-on introduction to Python programming tailored specifically for network engineers and IT professionals transitioning into automation and programmability. It focuses on building strong foundational coding skills that can later be applied to network automation, scripting, and API-driven workflows. No prior programming experience is assumed, and concepts are introduced progressively with practical relevance to networking tasks.

Students begin by learning the role of programmability in modern networking and how Python fits into automation workflows. The curriculum covers core Python syntax and constructs, including data types, string manipulation, conditionals, loops, functions, classes, and namespaces. Emphasis is placed on writing readable, reusable, and maintainable code while understanding how Python manages data and execution flow.

The course also explores data storage formats such as CSV, XML, and JSON, module and package management, user input handling, and exception management. Debugging techniques and tools are introduced to help students identify and resolve errors efficiently. Extensive hands-on labs reinforce each concept through practical exercises, enabling learners to write, test, debug, and maintain Python scripts that form the foundation for real-world network automation.

How you'll benefit

This class 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

Why Attend with Current Technologies CLC

- Our Instructors are in the top 10% rated by Cisco
- Our Lab has a dedicated 1 Gig Fiber Connection for our Labs
- Our Labs run up to Date Code for all our courses

Who Should Attend

The primary audience for this course is as follows:

- Network administrators
- Network engineers with little or no programming or Python experience
- Network managers
- Systems engineers

Course Duration

4 days

Course Price

\$3,595.00 or 36 CLCs

Methods of Delivery

- Instructor Led
- Virtual ILT
- On-Site

Prerequisites

Before taking this course, you should have:

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

OUTLINE

Module 1: Introducing Programmability and Python for Network Engineers

Module 2: Scripting with Python

Module 3: Examining Python Data Types

Module 4: Manipulating Strings

Module 5: Describing Conditionals, Loops, and Operators

Module 6: Exploring Classes, Methods, Functions, Namespaces, and Scopes

Module 7: Exploring Data Storage Options

Module 8: Exploring Python Modules and Packages

Module 9: Gathering and Validating User Input

Module 10: Analyzing Exceptions and Error Management

Module 11: Examining Debugging Methods

LAB OUTLINE

- Lab 1: Execute Your First Python Program
- Lab 2: Use the Python Interactive Shell
- Lab 3: Explore Foundation Python Data Types
- Lab 4: Explore Complex Python Data Types
- Lab 5: Use Standard String Operations
- Lab 6: Use Basic Pattern Matching
- Lab 7: Reformat MAC Addresses
- Lab 8: Use the if-else Construct
- Lab 9: Use for Loops
- Lab 10: Use while Loops
- Lab 11: Create and Use Functions
- Lab 12: Create and Use Classes
- Lab 13: Use the Python main() Construct

- Lab 14: Traverse the File Structure
- Lab 15: Read Data in Comma-Separated Values (CSV) Format
- Lab 16: Read, Store, and Retrieve Data in XML Format
- Lab 17: Read, Store, and Retrieve Data in JavaScript Object Notation (JSON) Format
- Lab 18: Read, Store, and Retrieve Data in a Raw or Unstructured Format
- Lab 19: Import Modules from the Python Standard Library
- Lab 20: Import External Libraries
- Lab 21: Create a Python Module
- Lab 22: Prompt the User for Input
- Lab 23: Use Command-Line Arguments
- Lab 24: Manage Exceptions with the try-except Structure
- Lab 25: Manage Exceptions with the try-except-finally Structure
- Lab 26: Use Assertions
- Lab 27: Use Simple Debugging Methods
- Lab 28: Use the Python Debugger
- Lab 29: Code a Practical Debugging Script