

Document Generated: 07/05/2024

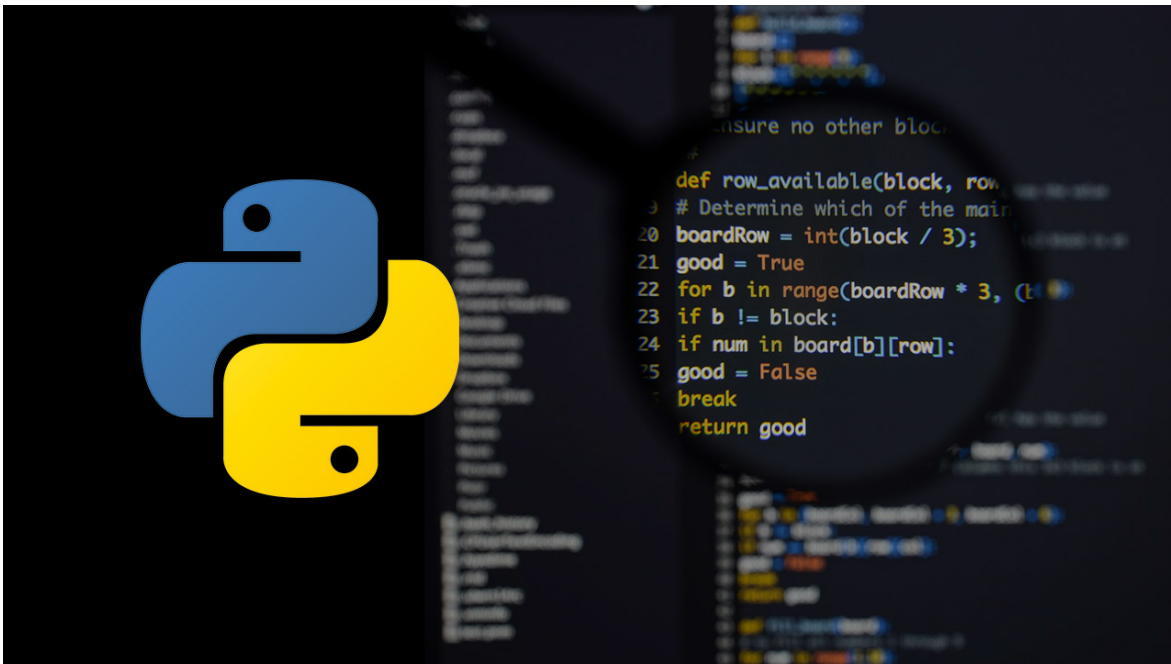
Learning Style: Virtual Classroom

Provider:

Difficulty: Intermediate

Course Duration: 5 Days

Python Programming for Network Administrators (TTPS4822)



About this course:

Targeted for network administrators looking to automate administrative tasks across a set of distributed clients **Python for Network Administrators** is an **introductory and beyond-level** practical, hands-on Python training course that leads the student from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and using the extensive functionality of Python modules with a focus on network-focused modules such as SSH, Git, and RESTful services. This comprehensive, practical course provides an in-depth exploration of working with the programming language, not an

academic overview of syntax and grammar. Students will immediately be able to use Python to complete these types of tasks in the real world.

Course Topics Covered: Throughout the course, students will learn about and explore:

- Running Python Scripts
- Getting Started
- Flow Control
- Sequence Data
- Defining Functions
- Working with Files
- Dictionaries and Sets
- Errors and Exception Handling
- Using Modules
- Regular Expressions
- Highlights of the Standard Library
- Introduction to Python Classes
- Real-life programming
- Special data types
- Network services, including SSH, REST
- Dates and times

The average salary of a Python programmer is **\$111,557** per year.

Course Objective:

- Create working Python scripts following best practices
- Use python data types appropriately
- Read and write files with both text and binary data
- Search and replace text with regular expressions
- Get familiar with the standard library and its work-saving modules
- Use lesser-known but powerful Python data types
- Create "real-world", professional Python applications
- Work with dates, times, and calendars
- Know when to use collections such as lists, dictionaries, and sets
- Understand Pythonic features such as comprehensions and iterators
- Write robust code using exception handling
- Automate network administrative tasks across distributed clients using SSH, REST, and Git

Audience:

- This course is appropriate for advanced users, system administrators and web site administrators who want to use Python to support their server installations, as well as anyone else who wants to automate or simplify common tasks with the use of Python scripts.

Prerequisite:

- Students should have basic development experience in any programming language, along with a working, user-level knowledge of Unix/Linux, Mac, or Windows.

Course Outline:

Module 1: An Overview of Python

- What is python?
- 1 -- An overview of Python
- What is python?
- Python Timeline
- Advantages/Disadvantages of Python
- Getting help with pydoc

Module 2: The Python Environment

- Starting Python
- Using the interpreter
- Running a Python script
- Python scripts on Unix/Windows
- Editors and IDEs

Module 3: Getting Started

- Using variables
- Builtin functions
- Strings
- Numbers
- Converting among types
- Writing to the screen
- Command line parameters

Module 4: Flow Control

- About flow control
- White space
- Conditional expressions
- Relational and Boolean operators
- While loops
- Alternate loop exits

Module 5: Sequences

- About sequences
- Lists and list methods
- Tuples

- Indexing and slicing
- Iterating through a sequence
- Sequence functions, keywords, and operators
- List comprehensions
- Generator Expressions
- Nested sequences

Module 6: Working with files

- File overview
- Opening a text file
- Reading a text file
- Writing to a text file
- Reading and writing raw (binary) data
- Converting binary data with struct

Module 7: Dictionaries and Sets

- About dictionaries
- Creating dictionaries
- Iterating through a dictionary
- About sets
- Creating sets
- Working with sets

Module 8: Functions

- Defining functions
- Parameters
- Global and local scope
- Nested functions
- Returning values

Module 9: Sorting

- The sorted() function
- Alternate keys
- Lambda functions
- Sorting collections
- Using operator.itemgetter()
- Reverse sorting

Module 10: Errors and Exception Handling

- Syntax errors
- Exceptions
- Using try/catch/else/finally
- Handling multiple exceptions
- Ignoring exceptions

Module 11: Modules and Packages

- The import statement
- Module search path
- Creating Modules
- Using packages
- Function and Module aliases

Module 12: Classes

- About o-o programming
- Defining classes
- Constructors
- Methods
- Instance data
- Properties
- Class methods and data

Module 13: Regular Expressions

- RE syntax overview
- RE Objects
- Searching and matching
- Compilation flags
- Groups and special groups
- Replacing text
- Splitting strings

Module 14: The standard library

- The sys module
- Launching external programs
- Math functions
- Random numbers
- The string module
- Reading CSV data

Module 15: Dates and times

- Working with dates and times
- Translating timestamps
- Parsing dates from text
- Formatting dates
- Calendar data

Module 16: Working with the file system

- Paths, directories, and filenames
- Checking for existence
- Permissions and other file attributes

- Walking directory trees
- Creating filters with fileinput
- Using shutil for file operations

Module 17: Network services

- Grabbing web content
- Sending email

Module 18: RESTful Services

- REST Overview
- Essential RESTful services with Flask
- Consuming RESTful services
- Overview of Reactive Programming with RxPY

Module 19: Programmatic SSH

- SSH, SCP and SFTP Overview
- Essential Paramiko
- Maintaining multiple clients

Module 20: DevOps and GitPython

- DevOps Overview
- Git Overview
- Programmatic Git with GitPython

Credly Badge:



Display your Completion Badge And Get The Recognition You Deserve.

Add a completion and readiness badge to your LinkedIn profile, Facebook page, or Twitter account to validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

[Find Out More](#) or [See List Of Badges](#)