Python Programming for Network Administrators (TTPS4822)

Modality: Virtual Classroom

Duration: 5 Days

About this course:

Directed for network administrators hoping to automate authoritative works over a lot of distributed customers Python for the Administrators of Network is a beyond-level and introductory practical, hands-on Python instructional course that drives the understudy from the nuts and bolts of running and writing scripts of Python to further developed features, for example, regular expressions, file operations, working with binary data, and utilizing the broad usefulness of Python modules with an emphasis on network-focused modules, for example, the services of Git, SSH, and RESTful. This extensive, practical course gives an inside and out the exploration of working with the language of programming, not a scholarly diagram of linguistic grammar and syntax. Understudies will quickly have the option to utilize Python to finish these kinds of assignments in reality.

Covered Course Topics: Throughout the course, understudies will explore and find out about:

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

The normal pay of a Python software engineer is $111,557 annually.

Course Objective:

- Make working Python contents following accepted procedures
- Use python information types suitably
- Peruse and compose records with both content and binary information
- Look and supplant the content with customary expressions
- Get acquainted with the standard library and its work-sparing modules
- Utilize lesser-known however incredible Python information types
- Make "genuine world", proficient Python applications

- Work with times, dates, and schedules
- Realize when to utilize assortments, for example, lists, sets, and dictionaries.
- Comprehend Python features, for example, iterators and understandings
- Compose powerful code utilizing exemption handling
- Network administrative tasks automation across distributed clients using REST, SSH, and Git

Audience:

This course is suitable for system administrators, advanced users, and web site administrators who need to utilize Python to help their installations of servers, also any other person who needs to simplify or automate common assignments with the utilization of scripts of Python.

Prerequisite:

Applicants ought to have fundamental advancement involvement with any programming language, alongside working, client-level information on Unix/Linux, Windows or Mac.

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