# PYTHON FOR NETWORKING & SYSTEM ADMINISTRATORS (TTPS4824) Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

## Overview

**Introductory and beyond-level practical hands-on Python training 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.

**Who Should Attend?**

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.

## Prerequisites

Attendees should have a working knowledge in developing basic Java 8 applications. Students should have basic development experience in any programming language, along with a working, user-level knowledge of Unix/Linux, Mac, or Windows.

# PYTHON FOR NETWORKING & SYSTEM ADMINISTRATORS (TTPS4824) Training

Duration: **4 Day**

BİLGİNÇ
IT Academy
info@bilginc.com
bilginc.com

## What You Will Learn

Working within in an engaging, hands-on learning environment, guided by our expert Python practitioner, students will learn to:

- 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

## Outline

**Session 1:  An Overview of Python**

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

**Session 2:  The Python Environment**

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

**Session 3:  Getting Started**

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

# PYTHON FOR NETWORKING & SYSTEM ADMINISTRATORS (TTPS4824) Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

**Session 4: Flow Control**

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

**Session 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

**Session 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

**Session 7: Dictionaries and Sets**

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

**Session 8: Functions**

- Defining functions
- Parameters
- Global and local scope

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com


- Nested functions
- Returning values

**Session 9: Sorting**

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

**Session 10: Errors and Exception Handling**

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

**Session 11: Modules and Packages**

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

**Session 12: Classes**

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

**Session 13: Regular Expressions**

- RE syntax overview
- RE Objects
- Searching and matching
- Compilation flags

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com


- Groups and special groups
- Replacing text
- Splitting strings

**Session 14:  The standard library**

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

**Session 15:  Dates and times**

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

**Session 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

**Session 17: Network services**

- Grabbing web content
- Sending email

**Session 18: RESTful Services**

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

# PYTHON FOR NETWORKING & SYSTEM ADMINISTRATORS (TTPS4824) Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

**Session 19: Programmatic SSH**

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

**Session 20: DevOps and GitPython**

- DevOps Overview
- Git Overview
- Programmatic Git with GitPython