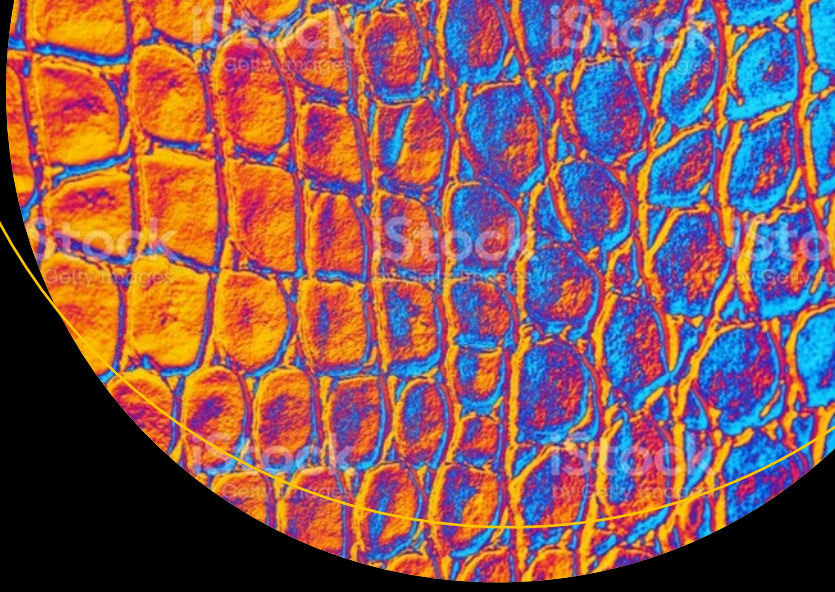


Introduction to Python Network Automation

The First Journey

Brendan Choi

Apress®

Introduction to Python Network Automation

The First Journey

Brendan Choi

Apress®

Introduction to Python Network Automation: The First Journey

Brendan Choi
Sydney, NSW, Australia

ISBN-13 (pbk): 978-1-4842-6805-6 ISBN-13 (electronic): 978-1-4842-6806-3
https://doi.org/10.1007/978-1-4842-6806-3

Copyright © 2021 by Brendan Choi

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image, we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors, nor the Publisher can accept any legal responsibility for any errors or omissions that may be made. The Publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Celestin Suresh John
Development Editor: James Markham
Coordinating Editor: Aditee Mirashi

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, email orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC, and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6805-6. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

For my wife Sue and children, Hugh, Leah, and Caitlin.

Table of Contents

About the Author xix

About the Technical Reviewer xxi

Acknowledgments xxiii

Introduction xxv

Chapter 1: Introduction to Python Network Automation 1

Laying the Foundation 1

Exploring Your Skills and Prerequisites 2

General Competencies of Three Main IT Domain Group 3

Comparative Analysis of IT Engineers' Responsibilities 6

Python and Network Automation Studies 10

Why Do People Want to Learn Python? 10

What Do I Need to Study Network Automation Using Python? 12

Hardware: Minimum Specifications for Laptop or PC 13

Software Requirements 14

Building a Network Automation Development Environment Using GNS3 16

Downloading Supplementary Guides and Source Code 20

Summary 21

Storytime 1: Mega Speed Shoe Factory 21

Chapter 2: Learn Python Basics on Windows 23

"Hello, World!" and print() Function in Interactive Mode 24

Preparing for the Python Exercises 27

Understanding Data Types 28

Indentation and Code Blocks 30

Commenting 31

Python Naming Convention 32

Doing the Python Exercises 33

Variables and Strings 34

Exercise 2-1: Create Variables and Assign Various Data Types 34

Exercise 2-2: Create Variables and Use the print() Function to Print Output 35

Exercise 2-3: Use Abbreviated Variable Assignment 36

Exercise 2-4: Enter a String, Escape Character Backslash (\), and type() Function 36

Exercise 2-5: Determine If a Variable Is a Container, a Tag, or a Pointer 38

Exercise 2-6: Use String Indexing 39

Exercise 2-7: Understand Variable Assignment Errors 39

Exercise 2-8: Avoid "SyntaxError: EOL While Scanning String Literal" 40

Exercise 2-9: Avoid "NameError: name 'variable_name' is not defined" 40

Exercise 2-10: Avoid "SyntaxError: invalid syntax" 41

Exercise 2-11: Avoid "TypeError: 'str' object is not callable" 41

Exercise 2-12: Add Multiline Comments with Triple Quotation Marks 43

Exercise 2-13: Use \ as an Escape Character to Remove Special Character Meanings 44

Exercise 2-14: Enter (Inject) Values/Strings in a String Using %s 44

Printing, Concatenating, and Converting Strings 45

Exercise 2-15: Use the print() and len() Functions to Create a Simple Function 45

Exercise 2-16: Use the lower() and upper() String Methods 46

Exercise 2-17: Perform String Concatenation and Use the str() Method 46

Exercise 2-18: Learn to Change a String Using Curly Brackets and .format() 47

Exercise 2-19: Adjust the Text Position with Curly Brackets and .format() 48

Exercise 2-20: Adjust the Number of Decimal Places 49

Exercise 2-21: Ask for and Receive User Input with input() 49
Exercise 2-22: Change a Word or Characters in a String 50
Recap: Variables and Strings 51
Numbers and Arithmetic Operators 52
Exercise 2-23: Use Arithmetic Operators 52
Exercise 2-24: Understand Integers vs. Strings 53
Recap: Arithmetic Operators 53
Booleans and Relational Operators 54
Exercise 2-25: Use Booleans 54
Exercise 2-26: Use Relational Operators 55
Exercise 2-27: Use Boolean Expressions to Test True or False 55
Exercise 2-28: Use Logical (Membership) Operators 56
Exercise 2-29: Change the Order of Operation with ( ) 56
Control Statements: if, elif, and else 56
Exercise 2-30: Use if and else 57
Exercise 2-31: Use if, elif, and else 57
Exercise 2-32: Write Code with if, elif, and else 58
Recap: Boolean and Conditionals 59
Functions 59
Exercise 2-33: Defining a Function 60
Exercise 2-34: Assign Default Values to a Function 60
Exercise 2-35: Define Hello and Goodbye Functions 61
Exercise 2-36: Use the Odd or Even Function 62
Exercise 2-37: Nest a Function Within a Function 62
Recap: Functions 63
Lists 63
Exercise 2-38: Create a List and Index Items 63
Exercise 2-39: Use append, extend, and insert in a List 64
Slicing 65
Exercise 2-40: Slice a List 65

Exceptions and Error Handling 66
Exercise 2-41: Avoid a ValueError Error 66
Exercise 2-42: Handle Errors with try and except in a List 66
Exercise 2-43: Find an Index of an Item in a List with the Customized Exception 68
Practicing Lists 69
Exercise 2-44: Practice a List 69
Using for Loops and while Loops 71
Exercise 2-45: Use the for Loop's upper( ) and capitalize( ) Methods 71
Exercise 2-46: Use the while Loop and len( ) Function 71
Sorting and Ranges 72
Exercise 2-47: Use sort( ) vs. sorted( ) in a List 72
Exercise 2-48: Link Two Lists 72
Exercise 2-49: Find the List Length Using the len( ) Function 73
Exercise 2-50: Use range( ) and the for Loop 73
Exercise 2-51: Use the String List for loop( ) and range( ) with Arguments 74
Recap: Lists and Loops 75
Tuples 76
Exercise 2-52: See Some Basic Tuple Examples 76
Exercise 2-53: Convert a Tuple to a List 76
Exercise 2-54: Determine Whether a Tuple Immutable 77
Exercise 2-55: Convert a Tuple to a List and a List to a Tuple 78
Exercise 2-56: Use a for Loop in a Tuple 78
Exercise 2-57: Assign Multiple Variables to a Tuple 79
Exercise 2-58: Create a Simple Tuple Function 79
Exercise 2-59: Use Tuples as List Elements 80
Recap: Tuples 80
Dictionaries 81
Exercise 2-60: Understand Dictionary Basics 81
Exercise 2-61: Avoid a Dictionary TypeError and Convert Two Lists to a Single Dictionary 82
Exercise 2-62: Use Keys to Print Values from a Dictionary 82
Exercise 2-63: Change a Dictionary Value 83

Exercise 2-64: Add a New Set of Keys and Values to a Dictionary 83
Exercise 2-65: Find the Number of Dictionary Elements 84
Exercise 2-66: Delete Dictionary Keys and Values 84
Exercise 2-67: Write a Python Script with a Dictionary 84
Exercise 2-68: Use a Dictionary for a Loop and Formatting 85
Recap: Dictionaries 86
Handling Files 87
Exercise 2-69: Read and Display Hosts File from Your PC 87
Exercise 2-70: Open and Close Hosts Files 88
Exercise 2-71: Create Code to Close a File in Two Ways 89
Exercise 2-72: Create a Text File, and Read, Write, and Print 90
Exercise 2-73: Use rstrip( ) or lstrip( ) to Remove Whitespace 93
Exercise 2-74: Python File Mode Exercise: Use r Mode 94
Exercise 2-75: Python File Mode Exercise: Use r+ Mode 95
Exercise 2-76: Python File Mode Exercise: Use a Mode 96
Exercise 2-77: Python File Mode Exercise: Use a+ Mode 99
Exercise 2-78: Python File Mode Exercise: Use w Mode 99
Exercise 2-79: Python File Mode Exercise: Use w+ Mode 100
Exercise 2-80: Python File Mode Exercise: Use x Mode 101
Exercise 2-81: Python File Mode Exercise: Use x Mode 102
Exercise 2-82: Open a Byte File in Python 103
Exercise 2-83: Handle Errors with try and except 104
Recap: Python file Handling Concept 105
Using Python Modules 105
Time Module 106
Exercise 2-84: Import the time Module 106
Sleep Method 106
Exercise 2-85: Use the time.sleep( ) Function 106
Exercise 2-86: Browse a Path Using the sys Module 107
Exercise 2-87: Add a New Filepath Using the sys Module 108
Exercise 2-88: Check Built-ins and sys.builtin_module 109

Exercise 2-89: Use a Simple import sys Module in a try and except Exercise 110
Exercise 2-90: Understand Lambdas by Making a Calculator 111
Recap: Module Concepts 112
Summary 112
Chapter 3: More Python Exercises 113
Getting Ready for the Exercises 113
Exercise 3-1: Concatenate a List and a Tuple into a Single List 114
Exercise 3-2: Use Python as a Calculator 114
Exercise 3-3: Do Some Basic String format( ) Exercises 114
Exercise 3-4: Ask for a Username 115
Exercise 3-5: Get a Username: Version 1 116
Exercise 3-6: Get a Username: Version 2 117
Exercise 3-7: Get a Username: Version 3 118
Exercise 3-8: Add a Temporary Filepath, Import ex3_7.py as a Module, and Run the Script 120
Exercise 3-9: Use Commas to Add Spaces Between Strings 120
Exercise 3-10: Practice if; if and else; and if, elif, and else 121
Exercise 3-11: Practice for ~ in range with end=' ' 122
Exercise 3-12: Practice for ~ in range 123
Exercise 3-13: Practice for line in ~ 123
Exercise 3-14: Use the split( ) Method 124
Exercise 3-15: Practice with lstrip( ), rstrip( ), strip( ), upper( ), lower( ), title( ), and capitalize( ) 124
Exercise 3-16: Create a file and read it four different ways 125
Exercise 3-17: Read and Output Files for a More Detailed Understanding 126
Exercise 3-18: Use the getpass( ) Module and User Input 127
Exercise 3-19: Understand the Difference Between Encoding and Decoding 128
Exercise 3-20: Handle CSV Files in Python with the csv Module 128
Exercise 3-21: Output a CSV File 132
Exercise 3-22: Find the Price of a Cisco ISR 4331 Router from a CSV File 132
Exercise 3-23: Calculate the Total Cost of the Router Purchases: No Module 133
Exercise 3-24: Calculate the Total Cost of the Router Purchases: Using the csv Module 134
Exercise 3-25: Convert dd-mmm-yy and Calculate the Difference in Days and Then in Years 134

Exploring the Python IDE Environment 136
Summary 137
Storytime 2: Machine vs. Human, The First Confrontation 138

Chapter 4: Introduction to VMware Workstation 139

VMware Workstation at a Glance 139
Type-1 vs. Type-2 Hypervisors 141
VMware Workstation Pro for a Proof-of-Concept Lab 142
Before Using VMware Workstation 142
What's Next on VMware Workstation 15 Pro? 145
VMware Workstation 15 Pro User Console 145
Basic Operations of VMware Workstation 15 Pro 146
VMware Workstation Pro: Basic Operations 147
VMware Workstation Menu 149
Virtual Network Adapters 152
Virtual Network Editor Overview 152
Virtual Network Interface Description 155
Revealing Each Virtual Network Interface Types 159
Summary 168

Chapter 5: Creating an Ubuntu Server Virtual Machine 169

Downloading and Installing a Ubuntu Server 20 Image 170
Downloading the Ubuntu Server 20.04 LTS Image 171
Installing Ubuntu Server 20.04 LTS 172
Logging In to a New Ubuntu Server 20 via SSH 196
Customize Ubuntu Server 202
Ubuntu VM Customization 1: Enable Root User SSH Login on Ubuntu Server 20.04 203
Ubuntu VM Customization 2: Install a Desktop GUI and Other Packages 205
Ubuntu VM Customization 3: Enable Root User GUI Access 210
Taking a Snapshot of a Virtual Machine 214
Cloning a Virtual Machine 217
Summary 222

Chapter 6: Creating a CentOS 8 Server Virtual Machine 223

Downloading and Installing a CentOS 8 Server Image 223

Downloading the CentOS 8 Server Image 224

Installing CentOS 8 Server 226

Logging In to a New CentOS 8 Server via SSH 260

Managing a Network Adapter on Linux 263

Creating a GNS3 VM by Importing an .ova File 269

Downloading and Installing the GNS3 VM from an .ova File 270

Summary 274

Storytime 3: The Origin of Hypervisors 274

Chapter 7: Linux Fundamentals 275

Why Learn Linux? 276

The Beginning of Linux 278

Understanding the Linux Environment 279

Understanding Linux Directories and File Formats 280

vi vs. nano 281

Introduction to vi 282

Introduction to nano 293

Linux Basic Administration 306

Changing the Hostname 306

Linux Basic File and Directory Commands 309

Linux File and Directory Exercises 311

Summary 327

Chapter 8: Linux Basic Administration 329

Information on Linux: Kernel and Distribution Version 329

Information on Linux: Use the netstat Command to Validate TCP/UDP Ports 332

Installing TFTP, FTP, SFTP, and NTP Servers 339

FTP Server Installation 340

Installing the SFTP Server 346

Installing the TFTP Server 350
Installing the NTP Server 355
Linux TCP/IP Troubleshooting Exercise 359
Summary 364
Storytime 4: Becoming a Savvy Linux Administrator, Perhaps Every IT Engineer's Dream? 364
Chapter 9: Regular Expressions for Network Automation 367
Why Regex? 368
To re or Not to re 369
Studying Regular Expressions Using Python 371
Method 1: Using Notepad++ 372
Method 2: Using the Linux Shell 374
Regular Expression Breakdown: [A-Z]{3}\d{4}[/]\w+ 375
Method 3: Using the Internet to Study Regular Expressions 375
Regex Operation: The Basics 376
Character Class ([]) 378
Dot (.): Single Character Match 379
Asterisk (*): Repetition 380
Plus (+): Repetition 380
{m, n}: Repetition 380
Python's re Module 382
Python re String Methods 383
Match Object Method 387
Compile Options 388
re.DOTALL (re.S) 389
re.IGNORECASE (re.I) 389
re.MULTILINE (re.M) 390
re.VERBOSE (re.X) 391
\: Confusing Backslash Character 392
Regular Expressions: A Little Revision Plus More 393
More Metacharacters 393

Lookahead and Lookbehind Assertions 403
Lookahead, Lookbehind, and Noncapturing group 403
Practice More Lookarounds 405
Lookaround Application Examples 407
sub Method: Substituting Strings 409
Substitute Strings Using sub 410
Using sub and \g to Swap Positions 410
Insert a Function in sub Method 411
Summary 413
Chapter 10: GNS3 Basics 415
GNS3 at a Glance 415
Installing GNS3 for the First Time 417
Downloading the GNS3 Installation File 417
GNS3 Installation and Setup 418
GNS3 Installation 418
GNS3 Setup Procedures 418
Getting Familiar with GNS3 424
GNS3 Menus and GUI 424
Gracefully Shutting Down GNS3 and GNS3 VM 426
Starting GNS3 as an Administrator 428
Using GNS3 for the First Time: Cisco IOS and Windows Lab 429
Cisco IOS Software License and Downloading an Older Cisco IOS 430
Decompressing Cisco IOS for GNS3 Use 431
Installing Cisco IOS on the GNS3 VM 442
Creating an IOS Lab Topology on GNS3 and Connecting to the Internet 454
Installing the Microsoft Loopback Adapter 462
Accessing GNS3 Network Devices Using MS Loopback 469
Configuring the GNS3 IOS Router Using a Python Script from the Windows Host PC 478

Cisco IOS and GNS3 Appliance Lab 482
Importing and Installing the GNS3 Linux Appliance Server 482
Manually Assigning IP Address to GNS3 Linux Appliance Server 486
Using the GNS3 Appliance Linux's Python to Manage R1 487
Summary 489

Chapter 11: Cisco IOS Labs 491

Cisco IOS and the Linux VM Lab 491
Creating a New GNS3 Project for the Linux VM Lab 492
Uploading and Downloading Files to the GNS3 IOS Router from Linux VMs (File Transfer Testing Lab) 503
Copying (Cloning) a GNS3 Project 511
Summary 514

Chapter 12: Building a Python Automation Lab Environment 515

Cisco CML-PERSONAL Software License Information and Software Downloads 516
Downloading the Cisco CML-PERSONAL IOSvL2 (Switch) Image 516
Downloading the Cisco CML-PERSONAL IOSv (Router) Image and startup_config Files 517
Installing the Cisco CML-PERSONAL L2 Switch and CML-PERSONAL L3 on GNS3 518
Installing the Cisco CML-PERSONAL L2 Switch on GNS3 518
Quick Communication Test on CML L2 Switch Integration on GNS3 528
Installing the Cisco CML-PERSONAL L3 Router on GNS3 529
Quick Communication Test on CML L3 Router Integration on GNS3 539
Building a CML-PERSONAL Lab Topology 541
Summary 548

Chapter 13: Python Network Automation Lab: Basic Telnet 549

Python Network Automation: Telnet Labs 549
Telnet Lab 1: Interactive Telnet Session to Cisco Devices on a Python Interpreter 551
Telnet Lab 2: Configure a Single Switch with a Python Telnet Template 557
Telnet Lab 3: Configure Random VLANs Using a for Loop 562

Telnet Lab 4: Configure Random VLANs Using a while Loop 566
Telnet Lab 5: Configure 100 VLANs Using the for ~ in range Loop Method 569
Telnet Lab 6: Add a Privilege 3 User on Multiple Devices Using IP Addresses from an External File 573
Telnet Lab 7: Taking Backups of running-config (or startup-config) to Local Server Storage 579
Summary 582

Chapter 14: Python Network Automation Labs: SSH paramiko and netmiko 583

Python Network Automation Labs Using the paramiko and netmiko Libraries 583
Python SSH Labs: paramiko 584
paramiko Lab 1: Configure the Clock and Time Zone of All Devices Interactively in the Python Interpreter 584
paramiko Lab 2: Configuring an NTP Server on Cisco Devices Without User Interaction (NTP Lab) 592
paramiko Lab 3: Create an Interactive paramiko SSH Script to Save Running Configurations to the TFTP Server 598
Python SSH Labs: netmiko 605
netmiko Lab 1: netmiko Uses a Dictionary for Device Information, Not a JSON Object 606
netmiko Lab 2: Develop a Simple Port Scanner Using a Socket Module and Then Develop a nemiko Disable Telnet Script 614
netmiko Lab 3: config compare 621
Summary 628

Chapter 15: Python Network Automation Labs: cron and SNMPv3 629

Cron and SNMPv3 Labs 629
Cloning a GNS3 Project for the Next Lab 630
Quick-Start to the Linux Scheduler, cron 637
Ubuntu 20.04 LTS Task Scheduler: crontab 637
CentOS8.1 Task Scheduler: crond 641
Learn cron Job Definitions with Examples 646

Using Python to Run an SNMPv3 Query 648
Quick-Start Guide to SNMP 649
Learn to Use SNMPwalk on a Linux Server 653
Borrowing Some Example Python SNMP Code 666
Summary 673

Chapter 16: Python Network Automation Labs: Ansible, pyATS, Docker, and the Twilio API 675

Python Network Automation Development Labs 675
Quick-Start Guide to Ansible: virtualenv Lab 1 676
Installing virtualenv and Getting Started with Ansible 678
Quick-Start Guide to pyATS (Genie): VirtualEnv Lab 2 685
Sendmail Lab Using an Imported Docker Image 700
Docker Components and Account Registration 701
Docker Installation 702
Test-Driving Docker 703
Docker Sendmail Python Lab 707
Lab: Sendmail Email Notification script Development 712
CPU Utilization Monitoring Lab: Send an SMS Message Using Twilio 717
TWILIO Account Creation, Install Twilio Python Module and SMS Message Setup 718
CPU Utilization Monitoring Lab with an SMS Message 723
Summary 732

Chapter 17: Upgrading Multiple Cisco IOS XE Routers 733

Practical Python Network Automation Lab: IOS XE Upgrade Lab 733
Applying OOP Concepts to Your Network 734
Flow Control and Controlling User Input: UID, PWD, and Information Collector 738
Lab Preparation 743
CSR 1000v IOS XE Software and Download 744
Cisco CSR 1000v Installation on VMware Workstation 745

Discussion of How an IOS (IOS-XE/XR) Is Upgraded on Cisco Devices 756
Tasks Involved in a Cisco IOS Upgrade 756
Summary 759

Chapter 18: Python Network Automation Lab: Cisco IOS Upgrade Mini Tools Development 761

Cisco IOS Upgrade Application Development 761
Part A: Pre-check Tools Development Connectivity Validation Tool 762
Collect User's Login Credentials and User Input 774
Collect a New IOS Filename and MD5 Value from a CSV File 778
Check the MD5 Value of the New IOS on the Server 788
Check the Flash Size on Cisco Routers 792
Make Backups of running-config, the Interface Status, and the Routing Table 796
Part B: IOS Uploading and More Pre-check Tools Development 799
IOS Uploading Tool 799
Check the New IOS MD5 Value on the Cisco Device's Flash 804
Options to Stop or Reload the Routers 810
Check the Reloading Device and Perform a Post-Reload Configuration Verification ... 819
Summary 829

Chapter 19: Python Network Automation Labs: Combining and Completing the Cisco IOS Upgrade Application 831

Creating a Single Python File 832
Summary 852
Final Words 853

Index 855

About the Author

Brendan Choi currently works as a senior technical consultant for Australia's leading IT integrator and Internet service provider, Telstra. He is a Certified Cisco Network Engineer and a Certified VMware Data Center Engineer, working on enterprise networking, data center, unified communications, security, and virtualization technologies for more than 17 years and now using Python automation to manage enterprise devices. He began his IT career with Cisco Systems and then moved on to networking and systems integrator roles with several reputable enterprise IT integrators. He previously worked for Dimension Data for seven years as a senior network engineer focusing on trending technologies to streamline processes, increase work efficiency, and develop technical training to improve team productivity through various open source software. He started exploring Python and network automation about seven years ago but got serious about its potential in the last five years. He prepared *Python 3 Network Automation Lab Guide for Network & System Engineers* for his colleagues and trained more than 100 network and systems engineers in his last company. Based on blog content and the lab guide, he authored and published *Python Network Automation: By Building an Integrated Virtual Lab* in Korean. He is interested in all aspects of IT technologies but most interested in enterprise networking, security, and virtualization technologies. He writes to share his love for general IT. His learning experience is shared through his blog (`https://italchemy.wordpress.com/`) and YouTube channel (Python Automation).

About the Technical Reviewer

Giuseppe Citerna, CCIE No. 10503, was the first CCIE Academy Instructor in Europe, the Middle East, and Africa. Passionate about technologies and learning systems, he is obsessed with the constant research of systems to allow everyone to leverage the Internet as a tool for both human and professional growth. He is currently working as a learning director at IeXa Academy (www.iexa.it), a startup founded with his longtime friend and partner Aldo Menichelli. It is active in the education technology field, and it is radically changing the vocational training system. He is also working as a learning and development consultant for Dimira, a Cisco Gold Partner system integrator. He started his career at Telecom Italia's TAC in 1999. Then he moved to NextiraOne, where he worked on enterprise networking, data center virtualization, unified communications, and knowledge management projects. Alongside his technical career, in 1999 he started working in education at the ELIS Center in Rome at the Italian Cisco Academy Training Center. Throughout his 20-year career, he has trained thousands of network engineers at several major companies. His core areas of interest are data center virtualization, network automation and programmability, and anything related to technologies applied to learning.

Acknowledgments

This book's content is based on various resources, including my study notes, work instructions, cheat sheets, content shared on personal blogs, and training guides used for internal network automation training (2018). The original version was published in Korean on July 25, 2019, but more than half of the content had to be rewritten to make this book suitable for its new audience. So, technically, this is my second published book, and I have spent many hours locked up in my study authoring this book.

I am very thankful to both my English publisher (Apress) and my Korean publisher (Acorn) for allowing me to write and publish the Python network automation learning techniques I have compiled over the years from the enterprise networking field. I am also grateful to my family, who stood by me while completing this book. My wife Sue is the biggest supporter of my work, and publication of my work has been only possible with her support. I had dreamed of writing a book and dedicating it to our three lovely children, Hugh, Leah, and Caitlin, so they can grow into strong and healthy adolescents who will live up to their dreams and also learn to share any valuable knowledge or unique experiences with others around them. Hugh, you will be a kind young man playing piano; Leah, you will continue to set the standard for everyone and strive for excellence; and Caitlin, one day, you will learn coding to become an outstanding data scientist/white hacker to help others.

I also would like to give special thanks to Giuseppe Citerna for his excellent technical review; this book has been made twice as good for our readers because of his experience. I also thank all the Apress staff who helped me to polish this book. Finally, I sincerely thank all the readers who have purchased this book to begin their Python network automation journey.

Brendan Choi (2021)

Introduction

A decade ago, network engineer roles consisted of installing, configuring, maintaining, and troubleshooting LANs, WANs, and any other IP connectivity technologies. Practical working knowledge of networking concepts and strong networking protocol knowledge were must-have skills for network engineers. Some of these protocols included OSPF, EIGRP, BGP, MPLS, VRF, IPsec, QoS, DNS, VTP, VPN, Broadcast, Multicast, and GRE. Even then, the job requirements for a network engineer demanded real-world experience; broad application of the advanced principles, theories, concepts, and techniques of networking; and the ability to troubleshoot, report, and resolve a networking problem.

Roughly five years ago, network engineers were forced to take on more administrative responsibilities based on ITIL's IT Service Management and ISO 9001's designated quality management standards. Middle management found a way to push down their administration work onto their subordinates. During this period, new buzzwords were invented by some tech geniuses and sold to the enterprise network market as the silver bullet to solve all of the IT industry's problems; these hot keywords were *software-defined network* and *network automation*. The introduction of these words hinted that the enterprise networking market was about to rapidly change. In fact, the industry started to demand more hybrid engineers with network programming skills on top of their existing responsibilities. In short, the market wanted more network engineers who could think and code like real application developers using one or more programming languages and open source programs. In line with the current market trend, Python has become one of the most popular programming languages among network engineers for networking task automation because of its ease of use and shorter development cycle. Python is not the only programming language used in network automation; other programming or scripting languages currently used include Shell, Bash, Perl, Java, API, Ansible, JSON, SOAP, Ruby, and REST.

Interestingly, a few years ago, not many organizations or engineers thought network engineers needed to know how to write code and develop automation scripts or applications. At that time, the network engineer role was clearly defined, with different groups of engineers divided based on the technology domain groups. Back then, the term *network automation* only meant creating ad hoc and simple Tcl scripts based on the Cisco IOS built-in feature. Network engineers, especially those who studied and worked mainly on Cisco technologies, were hesitant to dive into any programming language to make their work more efficient. Although network automation skills could become a real weapon for many network engineers, it would be a bumpy road to network programmability.

Traditional network engineers only had to deal with OSI layer problems between layers 1 through 4; they were happy in their bubble, and there was no hurry for any innovation. They had an excellent working knowledge of networking concepts and protocols that kept them on the job for many years with decent salaries. But the concept of a software-defined network had been taking the enterprise networking industry by storm. Cisco was the world leader and a household name for enterprise networking in routing and switching, security, voice services, and, more recently, server equipment–related data center solutions. Cisco had been the trusted vendor and partner to larger enterprise customers as well as to smaller, SMB customers all around the world. Remarkably, it had been a "one-stop shop" enterprise IT solutions vendor for more than 40 years. This meant you bought complete IT solutions from a single IT vendor, and Cisco's famous SmartNet Support, the best post-sales technical assistant center (TAC) support at any time of the day around the world, was available to all paid customers. Think of Cisco Systems as the world's largest supermarket for

enterprise equipment and solutions. Unfortunately, ever since Cisco's lawsuit against a Chinese networking equipment manufacturing vendor, Huawei in 2003, Cisco has not been the same. Its technologies and intellectual properties were stolen, copied, and used by the Chinese vendor, undercutting Cisco's business model and eating up Cisco's market shares.

A few years ago, to have a successful career as a network engineer, you only had to focus on learning Cisco-specific technologies to keep your job. Many enterprise networks were predominantly designed and deployed with IP devices manufactured and supported by Cisco until ten years ago. It probably was the Golden Age of Cisco, where Cisco led the enterprise networking and infrastructure solutions market based on its robust hardware-based solutions. Even today, when many IT companies hire a new network engineer, the first qualification they look at is what level of Cisco certification the candidate holds. During the initial job application process, having a current Cisco certification or not could be a single deciding factor to make it through the next round of the job interview process. Of course, practical experience is more important than the actual certification. Still, specific Cisco certifications have been an important measure of networking and technical proficiency.

No doubt, Cisco Systems has been the dominant vendor in the enterprise networking equipment and solutions market since the early 1990s and still is in the current market. Still, with the introduction of disruptive virtualization and software-defined networking (SDN) concepts and the recent onset of open source tools, network automation technologies have been threatening to disrupt Cisco's existing business models. The concept of automation has been around for many years; its roots can be found in the modern industrialization of the 1900s, and about 20 years ago, the automation phenomena swiped out many industries relying on electronic terminals (older computer types with specific functions). So, why has automation become the hot topic in today's IT industry in general, which includes networking, security, data center, unified communications, storage, systems, and the cloud?

When you examine the main driver for the latest automation craze in enterprise networking, you will see there has been a rapid development in high-performance computing based on faster and more efficient hardware. The hardware components have shrunk in size, providing smaller but more powerful hardware platforms for software to run in an optimized environment. The existing hardware-centric infrastructure architecture that required tens to hundreds of servers can now run on a handful of powerful yet energy-efficient servers running on a virtualized environment. What used to run on hardware-based servers became virtual machines, happily running on bare-metal hypervisors. Software-centric solutions based on high-performance hardware, faster intranet speed, and faster LAN and SAN network connectivity are now dominating our IT industry. The realization of the software-based solution over the traditional hardware-based solutions empowers organizations to be more flexible and economical, using more robust and reliable IT services throughout the whole life of the infrastructure devices and applications. For example, one of the most popular virtualization technology examples is cloud computing. There is very little client-owned, on-premise hardware in such an environment, and many of the IT services are on-demand services. Today these new technologies are already replacing more traditional IT infrastructures.

Engineers with passive learning strategies under the current enterprise networking market trend will have only a grim future waiting ahead. As history repeats itself, the market will leave those engineers behind and move on quickly. In the last decade, virtualization solutions have become the new norm of server and storage platforms. Now, even the networking devices are gradually moving away from hardware-based platforms and moving into the virtual environment, where everything can be controlled by application programming interfaces (APIs). Two such solutions making their mark are VMware's NSX and Cisco's ACI. Virtualized networking devices are just a handful of files running on hypervisors, making the administration of control, data, and management planes extremely easy with software. Soon, more and more engineers' work will be replaced by automated applications that talk to each other through APIs and programming languages such as Python and Go. Also, the fine lines between specific technologists based on different technology domains are getting blurred. Simultaneously, what traditional IT engineers have been doing will be replaced slowly with lines of code or intelligent applications.

As seen in many industries, there is no doubt that software and automation technologies will be taking over our lives. The enterprise network automation is part of this unstoppable movement. Software removes the traditional barriers that stood between systems, software, and network engineers, and all engineers are now encouraged to learn and write code. As a network engineer, once you learn to code and apply your new skills in your own company's network, there is a good chance that you will remain well-footed in the industry. This is my argument for why all network engineers must learn to code in a programming language. You must keep learning and apply your newly acquired coding skills to your work to take advantage of software-defined networks.

Target Audience

Before writing any documentation, you have to think about the context and consider who your target audience is. The same goes when writing a book of this magnitude. As the book's title suggests, this book is about introducing Python network automation and is written for people with little or no Python network automation experience. In other words, this book is not just about Python, nor an introduction to networking basics; it is a guide to help networking students and engineers pick up Python to start on their network automation journey.

Topics Covered

This book covers a wide range of IT topics to expose you to a selection of technologies in three different IT domains: networking, systems, and programming. This book focuses on IT technologies that will provide a strong foundation to extend your knowledge to the next level. Readers are recommended to read each chapter sequentially, from Chapter 1 to Chapter 19. Even if you are a seasoned network professional, you are recommended to read each chapter in sequence to get the most out of this book. Your success in reaching and completing the last lab in this book depends on building your lab environment in the first half of the book.

On the first page of each chapter, this book provides recommended difficulty scales based on the intricacies of tasks involved in each chapter; this is to set the right mindset and expectation of the readers. You start with a casual reading in Chapter 1, and as you progress through the book, each chapter's difficulty increases compared to the previous chapter. Each chapter was designed to equip you with different IT skills and by the time you reach the end of this book, you will reach at least an intermediate IT and Python skill level where you can build a fully functional proof-of-concept networking lab and start writing Python code for Network Automation. The difficulty levels are only there as a guideline, and the actual difficulty of tasks in each chapter will be dependent on your own experience and prior knowledge of the covered contents.

Figure I-1 shows the difficulty levels of each chapter, and Figure I-2 displays the topics covered in each chapter.

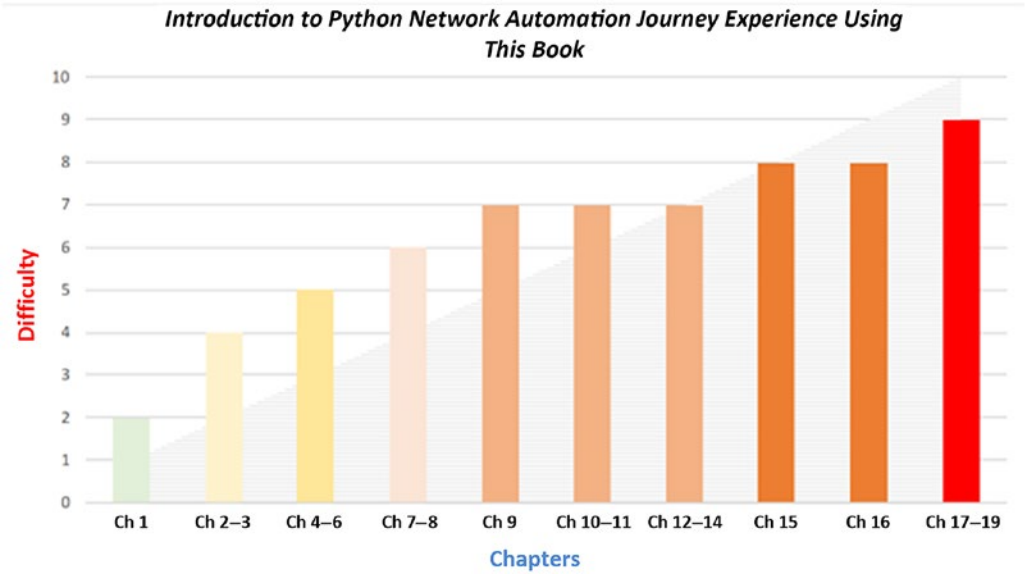


Figure I-1. *Chapter difficulty increments*

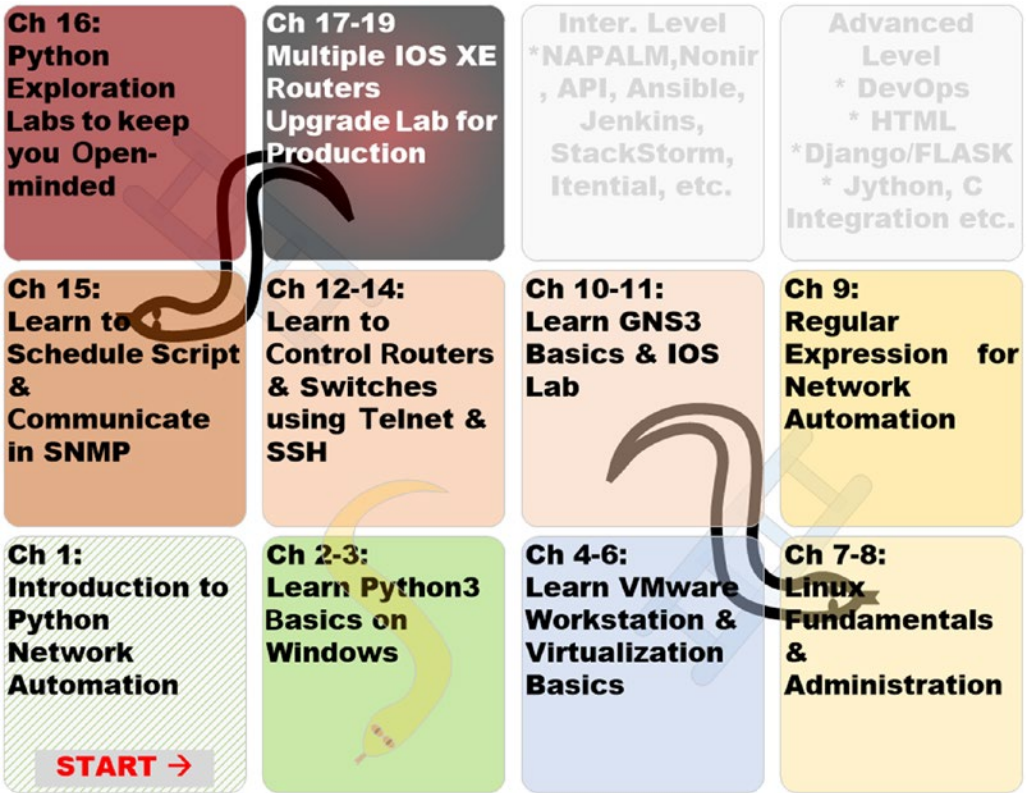


Figure I-2. *Chapter topics covered in this book*

Chapter 1: Introduction to Python Network Automation

The first chapter talks about why you want to study Python as the de facto network automation programming language for your work. It discusses the available skill sets of three IT domain groups: systems, DevOps, and network. You will study a network engineering group's weaknesses in network programmability and strategize how to embark on network automation studies using Python. Also, the chapter outlines the system and software requirements for a successful journey using this book.

Chapters 2–3: Learn and Practice Python Basics on Windows

You have to master Python basics to write working network automation code in Python. In Chapters 2 and 3, you are taken through a series of basic Python syntax and concepts on Windows 10. As the most commonly used end-user operating system is Windows, the entry to Python becomes easier while learning Python on Windows. These two chapters contain hands-on Python exercises connecting some Python concepts to networking concepts.

Chapters 4–6: Learn VMware Workstation Basics and Build Virtual Machines

VMware's desktop virtualization will be the gel holding together all our development integration. The chosen desktop virtualization software for this book is VMware Workstation 15 Pro, and you will be provided with a step-by-step user guide on how to use VMware Workstation 15 Pro on Windows 10. You will download Linux bootable and GNS3 VM ova images to start building virtual machines as part of initial lab preparation. You will get familiar with all the VMware Workstation basics and some virtualization basics while installing and creating virtual machines.

Chapters 7–8: Linux Fundamentals

Linux administration knowledge is a key to your Python network automation success as around 30 percent of enterprise systems are based on Linux operating systems (OSs). Using the two Linux virtual machines from Chapters 5 and 6, you will briefly learn the fundamentals of Linux. You will use Ubuntu 20 LTS (Debian derivative) and CentOS 8.1 (Red Hat derivative) servers to get exposure to two different Linux distros. In Chapter 7, you will learn to manage files and directories and use vi and nano text editors. In Chapter 8, you will learn the basic Linux system administration, Linux kernel validation, and networking commands to build TFTP, FTP, SFTP, and NTP servers.

Chapter 9: Regular Expression for Network Automation

If you want to write an intermediate to advanced Python script, you are going to need a tool called a *regular expression*, aka the `re` module in Python. In this chapter, you will experience the wonderful world of regular expressions. The chapter teaches you how to use various regular expressions and use Python's `re` module in certain networking-related scenarios. To move up to the intermediate level, you have to know how to handle or massage various data using regular expressions. This topic will make you so much stronger in your programming skills.

Chapters 10–11: GNS3 Basics and Cisco IOS Lab

GNS3 VM is a Linux (Ubuntu) virtual server hosting all our networking devices such as routers, switches, and firewalls. You will learn to install GNS3 on your Windows Host PC and then integrate the GNS3 VM to run on VMware Workstation as a VM. In Chapter 11, you will learn how to install an old Cisco IOS router image and then learn and master how to drive basic networking lab on GNS3 with ease. You will learn the GNS3 tips and tricks learned over the years working on the GNS3 lab building. At the end of these chapters, you will build various networking labs with flexibility and speed.

Chapters 12–14: GNS3 Cisco CML-PE integration and CML-PE labs

GNS3 is compatible with the later Cisco Modeling Lab (CML) layer 2 and layer 3 images; here, you will learn to integrate CML switch (L2) and router (L3) images with GNS3. For your network automation labs, you need both later model switches and routers; CML image integration with GNS3 replaces the hardware-based switches and routers and allows us to configure many routers and switches as virtual networking devices on GNS3. You will learn to build a test network topology and learn the basics to control routers and switches using Python's Telnet and SSH (paramiko) libraries. Your commands will be written into Python networking codes, and you will see the configuration changes on your routers and switches right in front of your eyes.

Chapter 15: Python Network Automation Exploration Labs: Cron and SNMPv3

Full automation does not involve human interaction; if you want to run your script at 2 a.m. while you are asleep, then cron is the right tool for this job. In this chapter, you will learn how to use Linux's task scheduler, cron. You will learn to use this tool on both Ubuntu and CentOS. You will trigger scripts to run at a specified time using the cron scheduler. You will also be introduced to Python and SNMPv3 to get you started with Python's SNMP network monitoring. You will get exposure to a set of tools that are very different, but you will be better equipped with two more tools under your belt.

Chapter 16: Python Network Automation Development Labs

This chapter will help you to think outside of the box for Python network automation use cases. You will learn how to use virtualenv in Python to avoid library incompatibilities and save time for new library testing. You will then get exposure to Docker images by downloading a Python network automation Docker image and running the Docker container using a tiny portion of the Linux host's resources. Using Docker, you will learn how to send an email notification to your email using Sendmail. You will also write a CPU utilization monitoring application and integrate a simple Twilio API script to send SMS to your mobile device when your testing router experiences high CPU utilization.

Chapter 17–19: Practical Python Network Automation Lab: IOS XE Upgrade Lab

You are now moving up to the next level! These final chapters are the highlight of this book. First, you will learn object-oriented programming (OOP) in Python in the context of networking devices. Then the chapter will briefly discuss the application flow control, followed by ten mini tools development for a Cisco CRS 1000v IOS XE router upgrade. By putting together smaller tools, you will build a fully functional and interactive IOS XE upgrade application to upgrade one or more Cisco Cloud Service Routers. Imagine upgrading 20 or 200 networking devices in one change window using Python's loop statements. You will begin to realize the real power of Python in enterprise network automation.

Configuring the Lab Environment

A significant part of software installations and lab building will be outlined in this book. Some chapters will require you to download essential software installation guides and complete a supplementary software installation before beginning the chapters. You can download the supplementary pre-installation guides from my GitHub page. There you will also find all the source code used in this book.

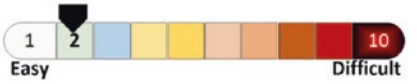
- *GitHub*: `https://github.com/pynetauto/apress_pynetauto`

CHAPTER 1

Introduction to Python Network Automation

This chapter serves as a primer to this book and discusses what it feels like to be an IT professional in today's IT industry. We will also discuss the different enterprise IT engineering domain groups and their responsibilities. The chapter then compares each domain group's weaknesses and strengths and draws up a working study plan that serves as the book's foundation. This chapter will also discuss why you might want to learn a programming language, in our case, Python, since this is your primary reason for purchasing this book and get started on your network automation journey. Finally, we will discuss the minimum PC/laptop requirements to build a fully working Python/Linux/network automation lab on a single PC. All lab machines, Linux servers, routers, and switches will be installed on a single PC/laptop using a recommended software set.

**Suggested difficulty level only.*
**This level may vary based on the reader's experience.*



Laying the Foundation

In recent years, network programmability concepts have been taking the enterprise networking industry by storm, and network automation has been at the eye of the storm for a few years. Enterprise businesses and organizations that are spending millions of dollars on traditional IT lifecycle management based on traditional IT frameworks have been searching for a new IT framework that will provide more stable and predictable network operations without interruption to their IP service and end-user applications. Quite a few network engineers have already started exploring network automation, and the remaining engineers are still trying to make a start. Still, many are encountering various challenges. Looking back on my own networking automation journey, I know it can be a slow, painful, and daunting experience.

Whether you have never touched a programming language in your career or are new to software-defined networks, starting on the network automation journey feels like you are trying to climb up a big mountain. Once you reach the first peak, you may realize that even bigger mountains await. You may even want to give up.

You probably have been living a comfortable life as a network engineer for the past few decades. Now your bosses expect you to upskill your nonexistent programming skills so you can replace manual tasks with lines of code so that you can add more value to your company. If you say no, your bosses have a Plan B: somebody else will write the code for your team to automate your work, and you will start losing credibility (and maybe even your job). You need to step outside of your comfort zone to start on our network programmability journey. Network programmability still is uncharted ground for most traditional network engineers.

© Brendan Choi 2021
B. Choi, *Introduction to Python Network Automation*, https://doi.org/10.1007/978-1-4842-6806-3_1

In this chapter, we will identify three main IT domain groups in a typical IT environment today. Then, we will define the common IT skill sets that each IT domain group possesses to determine each group's relational strengths and weaknesses. This book was written by a network engineer for network engineers and reviewed by a Cisco Network Academy instructor. The strengths and weakness discussions will be from an enterprise network industry perspective. The point is to learn how the Network group can use realistic learning strategies to get closer to the other two groups (DevOps and systems groups) and grow into cross-functional engineers who have strong networking skills as well as the skill sets to manage Linux operating systems and write code to develop business applications. This chapter also introduces a cross-functional "hybrid engineer" concept, which will soon be in enormous demand in the IT job market. The career knowledge growth for such hybrid engineers will take the shape of a *T*, as they have a firm foothold in their main IT domain and extend out to other domain skill sets as the top of the letter *T*.

In almost every chapter of this book, except this one, you will install, configure, practice, code, and learn from your PC or laptop, so your system's minimum requirements are the success of your Python network automation study with this book. Hence, we will review the minimum system requirements for your system and introduce you to the software used to gauge the size of crucial tasks outlined in this book. We will also briefly touch on an integrated development environment (IDE) for Python network automation development and provide all the download links for the software, source code, and files used in this book. At the end of this chapter, you will better understand a current network engineer's strengths and weaknesses. Hence, you will know the gaps (target study areas) you have to focus on to start writing Python code to develop your own network automation applications.

Exploring Your Skills and Prerequisites

The section discusses the three main IT domain groups' skill sets at work today. You will learn each group's strengths and weaknesses. From the network group's perspective, we will discuss and dissect the weaknesses in your skill set and develop workable study strategies to work on the weaknesses and turn them into your strengths. Studying Python syntax and concepts will help you reach 25 percent of your network automation goals. What about the other 75 percent? To write Python network automation code, you have to become strong in many other areas besides networking, and this chapter will help you become better in these areas. Of course, we will also cover Python basics from a networking perspective. Together, we will work on a study plan to address the common network engineer's weaknesses and guide you in the right direction for network automation using Python.

If you are currently working in the IT industry, especially on enterprise-level routing, switching, and security technologies, you should take pride in your job. You probably can relate to the image presented in Figure 1-1. Although the image might make you laugh, you have probably seen an IT engineer who walks and talks like he knows it all. We have all been in that situation where our clients expect us to know everything as the technical expert.

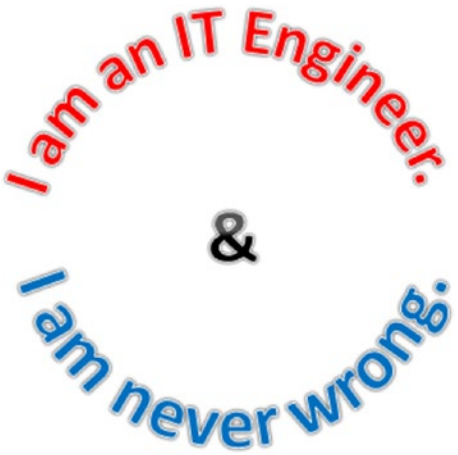


Figure 1-1. *IT engineers*

Sadly, the image in Figure 1-1 holds some truth to how some IT engineers think and behave at work. Since the first personal computer was invented in 1971, many IT jobs have come and gone as different technologies have emerged and disappeared. Most enterprise IT ecosystems continuously evolve as new technologies are introduced into the IT market. Unfortunately, some IT engineers are too stubborn to move with the times and often get caught out during technology transitional periods, which results in early termination from their IT careers. Currently, the age of artificial intelligence (AI) and IT automation has arrived in the enterprise network IT industry. If you refuse to skill up, your position might not be as secure as you think.

In fact, for many years, the IT industry has been trying to drive the operational costs down by offloading less-skilled jobs to developing countries where the IT operational cost is relatively cheaper than on the home turf. For many years, driving the operational costs down and reducing the overheads spent on human resources were trends for many organizations. We live in an era where human resources are an overhead cost in a business model, and many clients these days want to pay less but still demand quality IT delivery services. Contrary to many organizations' claims, many local IT jobs are considered overhead rather than valuable human resources for an organization in advanced countries. That is, many IT organizations claim they value their staff, and their priority is the well-being of these human resources in the organization's IT operation. Still, when the crunch time comes, almost every organization succumbs to the financial pressure and cuts IT operational costs to maximize the organization's profit. In the last 20 years, IT outsourcing and offshoring efforts have had great success to some extent. Still, each IT department is driven harder than before to drive the operational cost down, which has been the catalyst in speeding up the adoption of IT automations such as software-defined networks and infrastructure as code.

General Competencies of Three Main IT Domain Group

Bangalore, India, is commonly known as the Silicon Valley of India, and there is a common saying there: "There are two types of people in IT profession; one, IT professionals, and two, professionals who manage these people." In this book, to help your understanding, we are dividing IT domain groups into three different groups based on each group's competencies and characteristics. Then we'll compare their general technical competencies to each other to forecast what the near-future IT industry may look like for one particular group of engineers, that is, the network engineering (connectivity) group. Who will be at the forefront of the networking field in the next five years and beyond? Let's review the gaps and requirements to look ahead and develop a plan to study Python, network automation, and any other requirements to build your confidence to skill up in IT.