

Network Programming & Automation

Jason Edelman

jason@networktocode.com

```
>>> network.toCode()
```

Who is Network to Code?

- Founded in 2014
- Consultancy specifically focused on network automation technologies
- Provide Training and Professional Services helping roll out network automation tools and technology
- Long term "Training and Enablement" programs that often include 15-25 days of training within a year (for private engagements)
- Flagship course is 4-5 day on-site instructor-led Python and Ansible Network automation course

Course Objectives

- Learn and obtain foundational programming and automation skills with a focus on network engineering
- Provide an on-ramp for your network automation journey
- Understand data types relevant for all programming languages and tools
- Be able to read and write Python scripts to perform network automation tasks
- Be able to read and write Ansible playbooks to perform network automation tasks
- Understand a variety of network APIs and types and how to consume them in Python and Ansible

Target Audience

This course was built for Network Engineers that are looking to gain an introduction into Python and Ansible exploring its relevance with respect to Network Automation & Programmability.

The course does not assume any background in programming or software development methodologies, but the student should have some experience in the networking field as examples pertain to managing network devices such as switches and routers.

High Level Course Outline

- Module 1 – Getting Started with Python
- Module 2 – Automating Network Devices with SSH
- Module 3 – Automating Network Devices with APIs
- Module 4 – Network Automation with Ansible

Topics such as YAML, JSON, XML, and Jinja2 template are reviewed in several sections.

Course Summary

- Python Fundamentals
 - Data Types
 - Conditionals, Loops, Functions
 - Writing Scripts
 - Command Line Arguments
 - Working with Files
 - Python Libraries
 - Open Source Network Libraries
 - HTTP-Based and RESTful APIs
 - Vendor APIs
- Ansible
 - Playbooks, Plays, Tasks, Modules, Inventory
 - Writing Playbooks
 - Configuration Templating with Jinja2
 - Managing Network Configurations
 - Automated Data Collection & Reporting
 - Multi-vendor configuration with Ansible Roles
 - Dynamic Inventory

Course Layout

- **Day 1 - Getting Started with Python**

- Understanding Python Data Structures
- Consuming Nested Objects
- Using Python Modules
- Reading & Writing to Files
- Writing Basic Python Scripts

- **Day 2 - Automating Devices with SSH**

- SSH'ing to network devices with **netmiko**
- Adding **Conditional Logic** to Scripts
- Optimizing Code by adding **Loops**
- Writing Modular Code with Python **Functions**
- Passing in User input with **Command Line Arguments**

- **Day 3 (AM)- Network APIs**

- HTTP-Based (RESTful) APIs
- Arista eAPI
- Nexus NX-API
- IOS-XE RESTCONF
- Postman
- Python requests

- **Day 3 Bonus Material (Time Permitting)**

- Text Parsing (RegEx)
- TextFSM
- NAPALM (in Python) - also covered in Ansible

Course Layout

- **Day 4 (PM) - Ansible Overview**

- Creating an Inventory File
- Writing your First Playbook
- Group Variables
- Ansible Playbook Options: check mode, verbose mode
- Modules: debug, "config" modules
- Passing in Interactive User Input
- YAML Overview

- **Day 5 - Ansible (cont'd)**

- Configuration Templating with Jinja2
- Backing Up and Restoring Configurations Using Open Source Modules

- **Day 5 - Ansible (cont'd)**

- Configuration Deployment with NAPALM
- Executing Show commands
- Text Parsing (RegEx) with Ansible
- Performing Real-time Compliance Checks
- Diving Deeper into the Config Module
- Making API calls from Ansible
- Dynamic Device Discovery with SNMP via Ansible
- Understanding Roles
- Creating Automated Reports
- Dynamic Inventory

Topology

Everyone has a virtual topology that will be hosted in the public cloud for the week. It includes:

Base:

- 3 x CSR 1000V (16.8.1 w/ RESTCONF/NETCONF/YANG)
- 2 x NX-OSv (Virtual 9K running 7.0.3)
- 4 x cEOS (4.21.0F)
- 3 x vMX (18.2R1.9)
- 1 x Ubuntu Jump host (18.04)

Introductions

- Name
- Job/Role
- Experience
 - Programming/Automation/NetEng Experience

Housekeeping

- Length & Time
- Breaks
- Feedback

Lab & Lecture Material

- Access to Lecture / Lab Material
 - Email your GitHub USER ID (not email address) to labs@networktocode.com or directly to your instructor.
 - If you don't have one, create one at <https://github.com>

QUESTIONS?