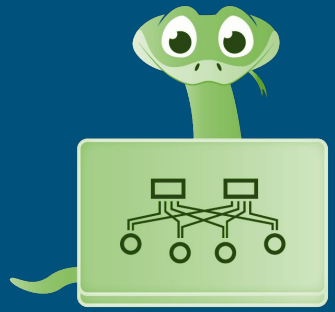

PYTHON

FOR NETWORK ENGINEERS

Onsite Training Session
June 2019

$ whoami

Kirk Byers
Network Engineer:
CCIE #6243 (emeritus)

Programmer:
Netmiko
NAPALM
Nornir

Teach Python, Ansible, Nornir in a Network Automation context

$ whoami

Carl Montanari
Network Engineer:
Dual-CCIE (#37652 R/S and SP)

R/S and Data Center focus, then down the automation rabbit hole!

Help folks to learn and implement network automation

Day2

1. Functions
2. Regular Expressions
3. Python Classes and Objects
4. Libraries/PIP/Virtual Environment
5. Modules
6. Packages
7. Namespaces (optional)
8. Linting

Flickr: au_tiger01

Functions:

- Defining a function
- Positional arguments
- Named arguments
- Mixing positional and named arguments
- Default values
- Passing in *args, **kwargs
- Functions and promoting the reuse of code

Exercises:
./day1/func_ex1.txt
./day1/func_ex2.txt
./day1/func_ex3.txt
./day1/func_ex4.txt

Python Regular Expresions

import re

Other re methods

re.split()
re.sub()
re.findall()

Exercises:
./day1/regex_ex1.txt
./day1/regex_ex2.txt

re.search(pattern, string)

- always use raw strings
- re.M/re.MULTILINE
- re.DOTALL
- re.I
- Parenthesis to retain patterns
- greedy/not greedy (.*?)

match.group(0)
match.groups()
match.groupdict()

Named patterns

(?P<software_ver>Ver.*)

Regular Expression Resources

Regular Expression Tutorial
https://regexone.com/lesson/introduction_abcs
This is a good resource if you are new to regular expressions.

Online Regular Expression Tester
https://regex101.com/
Select 'Python' on the left-hand side.

Python Regular Expression HowTo
https://docs.python.org/2/howto/regex.html
This is a good overview of regular expression special characters.
Start at the very top of the page and read through the 'Repeating Things' section.

Classes and Objects

```
class Server:
    def __init__(self, hostname, username, password):
        self.hostname = hostname
        self.username = username
        self.password = password
    def test_method(self):
        print(f"Device is: {self.hostname}")
        print(f"Username is: {self.username}")

svr1 = Server('test.domain.com', 'admin', 'passw')
svr1.test_method()
```

Exercises:
./day2/classes_ex1.txt
./day2/classes_ex2.txt

Libraries

import x

from x import y

sys.path

PYTHONPATH

Installing packages (pip)

Virtual Environments

Photo: Viva Vivanista (Flickr)

Virtualenv

virtualenv-3.6 -p /usr/bin/python36 test_venv

source /path/to/virtualenv/bin/activate

deactivate

pip list

pip install netmiko==2.3.0
pip install pycodestyle

pip freeze

Exercises:
./day2/virtualenv_ex1.txt

Python Code Structure:

- Imports at top of the file
- CONSTANTS
- Functions / classes
- if __name__ == “__main__”:
- Main code or main() function call

Exercises:
./day2/reuse_ex1.txt

Modules and Packages

Python Module

A Python file that you can import into another Python program

Example, storing device definition in an external file.

Python Package

An importable Python directory

__init__.py

Exercises:
./day2/reuse_ex2.txt
./day2/reuse_ex3.txt

Python Linters

Auto formatting with Python Black

pylint or pycodestyle

Consistency and conventions make your life easier.

Finds obvious errors. Finds problems you might not be aware of (reuse of builtins).

pylint my_file.py
pycodestyle my_file.py
pylama my_file.py

Review Exercise

Process the 'show_ip_int_brief.txt' file and create a data structure from it.

1. Create a dictionary of dictionaries.
2. The keys for the outermost dictionary should be the interface names.
3. The value corresponding to this interface name is another dictionary with the fields
   'ip_address', 'line_status', and 'line_protocol'.
4. Use pretty-print to print out your data structure.

Your output should be similar to the following:

```
{'FastEthernet0': {'ip_address': 'unassigned',
                   'line_protocol': 'down',
                   'line_status': 'down'},
 ... }
```

Exercises:
./day2/review_ex1.txt

Review Exercise

Process the 'show_arp.txt' file and create a data structure from it.

1. Create a dictionary where the keys are the ip addresses and the corresponding values are the mac-addresses.

2. Create a second dictionary where the keys are the mac-addresses and the corresponding values are the ip addresses.

3. Use pretty print to print these two data structures to the screen.

Exercises:
./day2/review_ex2.txt

The end...

Questions?

ktbyers@twb-tech.com

carl@twb-tech.com