# AIRHEADS meetup

aruba
a Hewlett Packard
Enterprise company

## Getting Started with Python & REST APIs for Network Engineers
*By Joe Neville*

20 September 2018

#ArubaAirheads

**Agenda**

- Why Python for Networkers?
- How to start – Real Life Story
- Advice on Getting Started

- I cannot teach you Python in one hour – so this is about teaching you why you should start and how you can start to learn.
- Aimed at networkers with little / no coding experience.

# What is Python?

- **High-level programming language**
- **Ease of use over speed**
- **Open-source**
- **Runs everywhere (Wins, Linux, macOS)**
- **Popular**
- **COMMUNITY!**

# Popular: TIOBE Index for Sept 2018

| Sep 2018 | Sep 2017 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 1 | | Java | 17.436% | +4.75% |
| 2 | 2 | | C | 15.447% | +8.06% |
| 3 | 5 | ^ | Python | 7.653% | +4.67% |
| 4 | 3 | v | C++ | 7.394% | +1.83% |

Source: https://www.tiobe.com/tiobe-index/

# Top 3 – First time 🥉

## TIOBE Index for September 2018

**September Headline: Python enters the TIOBE index top 3 for the first time**

Python has entered the TIOBE index top 3 for the first time in its history. This really took a long time. At the beginning of the 1990s it entered the chart. Then it took another 10 years before it reached the TIOBE index top 10 for the first time. After that it slowly but surely approached the top 5 and eventually the top 3. Python is becoming increasingly ubiquitous. It is already the first choice at universities (for all kinds of subjects for which programming is demanded) and is now also conquering the industrial world. Python's selling points are easy to learn, easy to install and easy to deploy. Other interesting moves this month are: Rust jumps from #36 to #31, Groovy from #44 to #34 and Julia from #50 to #39.

# Community 🤝

- **Lots of tutorials (books, videos, blogs)**
    - **Lots of free resource**
- **Online searches (stack overflow)**
    - **Ready-made answers**
- **Wealth of third-party libraries**
    - **E.G. requests**

# Why Python For Networkers?

- IT = change & convergence
- Think virtualization, public cloud, containers, serverless
- Software and server mgmt. already experienced great change, automation
- And EVERYTHING has an API!

# Why Python For Networkers?

- Networking is behind in this respect but it is changing….slowly.
- We want to speed up that change.
- CLI is identified as a primary inhibiter to change.

# Why Python For Networkers?

- **CLI is for humans, humans are slow and make mistakes, which cause outages.**
- **Renewed interest in network automation, just like other areas of IT that have automated (servers).**
- **APIs on Network kit as an alternative interface.**

# Why Python For Networkers?

## The New Legacy - Static

- Static / manual configuration

- Primarily using CLI

- Majority of campus customers use these practises.

## The New Hotness - Automation

**Requires code!**

- Combine multiple processes into workflows.

- Tasks can pass through multiple phases without manual intervention.

- Programmatic. "If this, then (do) that"

# Mobile First APIs

No surprise to see APIs on our software products

AIRWAVE
NETWORK
MANAGEMENT

CLEARPASS
ACCESS
MANAGEMENT

But now APIs rolled out to hardware

ArubaOS-CX

ArubaOS-Switch

# Aruba 🧡 APIs

## ArubaOS-CX

Auto-generated from database 100% day one

**8400**

**8320**

aruba
a Hewlett Packard
Enterprise company

# Aruba 🧡 APIs

# ArubaOS-Switch

**Started in 16.02**
**Now on v4 as rollout continues**

# Basic operations with Python 🐍

1. Login – send HTTP POST to
   `https://<ipaddr>/rest/v1/login`



username/password

`192.168.2.1/rest/v1/login`
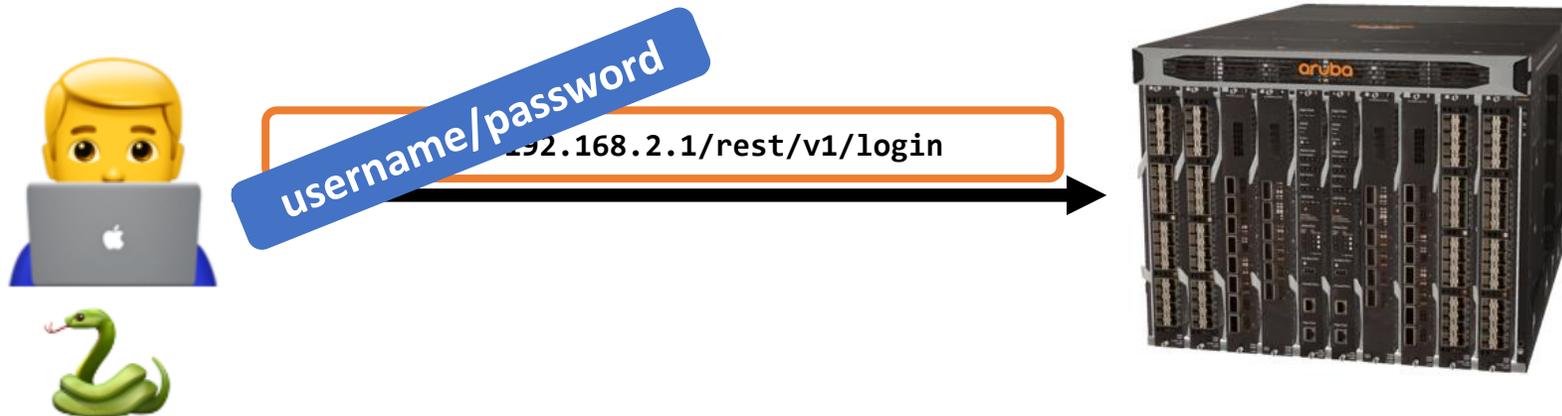
# Basic operations with Python 🐍

## 2. Response from CX - Cookie

response
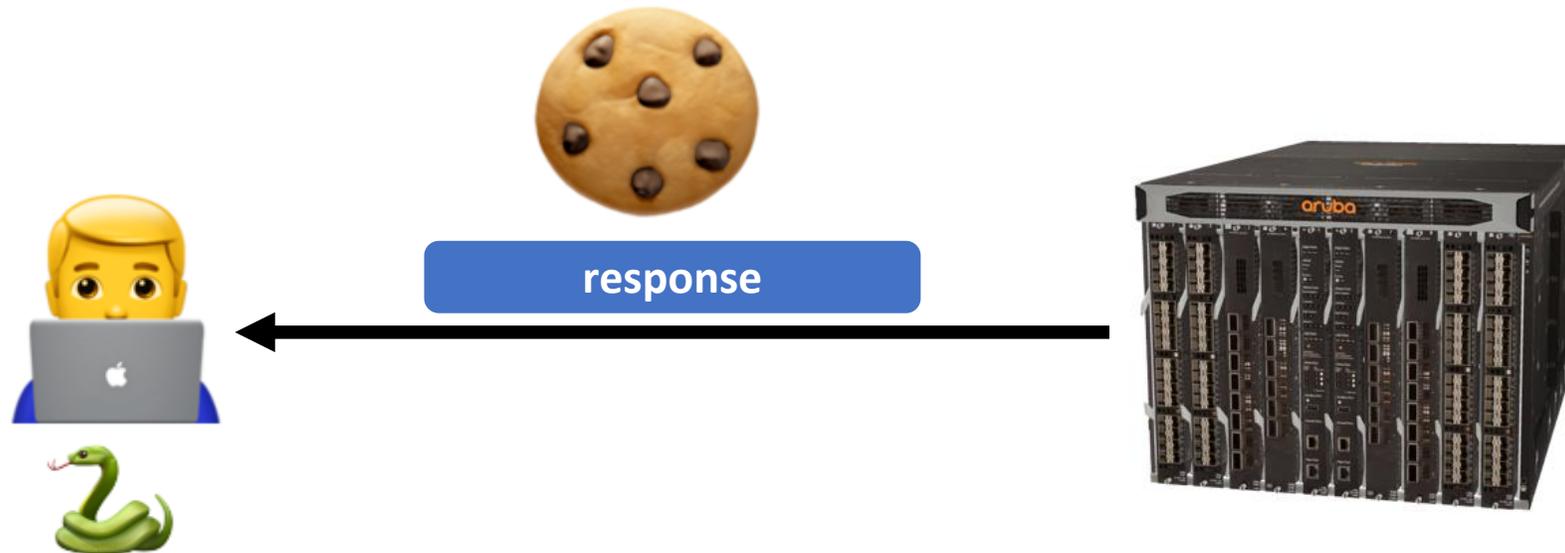
# Basic operations with Python 🐍

3. Add cookie to data & GET, POST, PUT, DELETE

`https://192.168.2.1/rest/v1/system/bridge/vlans`

# Basic operations with Python 🐍

## 4. JSON returned

[{'aclv4_in_statistics': {},
  'aclv4_in_status': {},
  'aclv4_out_statistics': {},
  'aclv4_out_status': {},
  'aclv6_in_statistics': {},
  'aclv6_in_status': {},
  'aclv6_out_statistics': {},
  'aclv6_out_status': {},
  'admin': 'up',
  'flood_enabled_subsystems': ['/rest/v1/system/subsystems/syster
  'id': 1,
  'internal_usage': {},
  'mgmd_counters': {'igmp_fast_leave_counter': 0,
                    'igmp_forced_fast_leave_counter': 0,
                    'igmp_membership_timeout_counter': 0,
                    'igmp_rx_bad_checksum_counter': 0,
                    'igmp_rx_malformed_pkt_counter': 0,
                    'igmp_rx_unknown_msg_type_counter': 0,
                    'igmp_rx_v1_all_host_query_counter': 0,
                    'igmp_rx_v1_member_report_counter': 0,
                    'igmp_rx_v2_all_host_query_counter': 0,
                    'igmp_rx_v2_group_specific_query_counter': 0,
                    'igmp_rx_v2_member_leave_counter': 0,
                    'igmp_rx_v2_member_report_counter': 0,
                    'igmp_rx_v3_all_host_query_counter': 0,
                    'igmp_rx_v3_group_specific_query_counter': 0,
                    'igmp_rx_v3_gssq_counter': 0,
                    'igmp_rx_v3_member_report_counter': 0,
                    'igmp_rx_wrong_version_query_counter': 0,
                    'igmp_tx_flood_on_vlan': 0,
                    'igmp_tx_forward_to_routers': 0,
                    'igmp_tx_v2_group_specific_query_counter': 0,
                    'igmp_tx_v3_group_specific_query_counter': 0},
  'mgmd_drop_unknown_status': {},
  'mgmd_dynamic_group_count': {'igmp_exclude_mode': 0, 'igmp
  'mgmd_enable': {},
  'mgmd_enable_status': {},
  'mgmd_igmp_block_ports': []

**That's a lot of info…but it is structured!**

a Hewlett Packard Enterprise company

# Basic operations with Python 🐍

5. Extract required info from Python dictionary (key-value pair)

6. Consume (print, if/else logic)

7. Logout

```
{
'id': 5,
'name': 'VOICE_VLAN_5
}
```

**key**      **value**
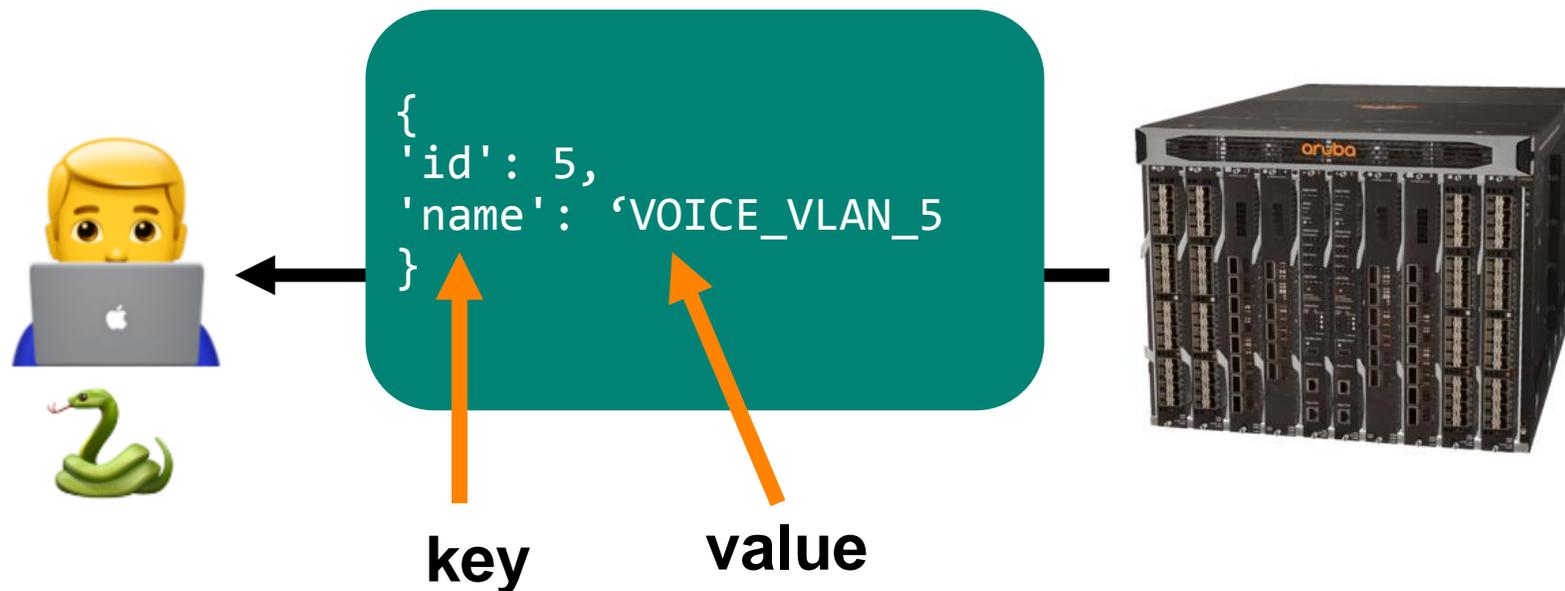
# "But I'm a networker…"

👐
*Push back!*

- "…I'm not a programmer"
- "…my company has dedicated developers to write code, why should I?"
- "CLI and SNMP are critical to my job, not Python"

## Network Automation is inevitable

# But why automate networks?

- Short answer: Save money. 

- Not quicker, but more reliable.
- Tasks completed quicker & cheaper.
- Automate menial tasks and focus on higher-level services & innovations.

# Gartner's View…

**By 2020, only 30% of network operations teams will use the command line interface (CLI) as their primary interface, down from 85% at YE16.**

# Their prediction seems off



Source: https://blogs.gartner.com/andrew-lerner/2018/01/04/checking-in-on-the-death-of-the-cli/
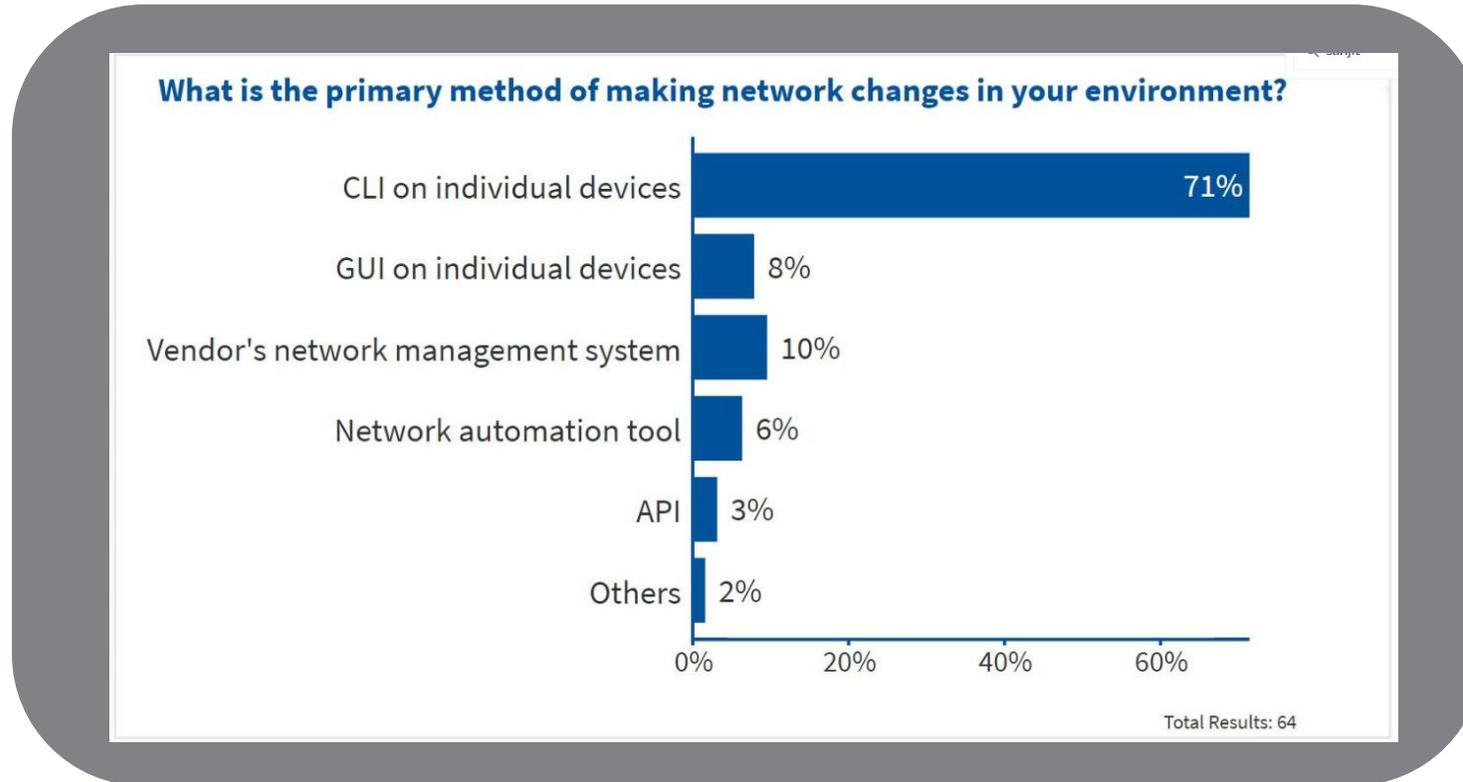
# But wait…

However, the real kicker is that when we asked folks where they plan to make the most strategic networking investments in 2018, **"network automation"** won by a landslide with 49%

Source: https://blogs.gartner.com/andrew-lerner/2018/01/04/checking-in-on-the-death-of-the-cli/

**BOOM! Python is your gateway in!**

# How to start learning Python?

## Believe this:

### Teach Yourself Programming in Ten Years

**Peter Norvig**

#### Why is everyone in such a rush?

Walk into any bookstore, and you'll see how to *Teach Yourself Java in 24 Hours* alongside endless variations offering to teach C, SQL, Ruby, Algorithms, and so on in a few days or hours. The Amazon advanced search for [title: teach, yourself, hours, since: 2000 and found 512 such books. Of the top ten, nine are programming books (the other is about bookkeeping). Similar results come from replacing "teach yourself" with "learn" or "hours" with "days."

The conclusion is that either people are in a big rush to learn about programming, or that programming is somehow fabulously easier to learn than anything else. Felleisen *et al.* give a nod to this trend in their book *How to Design Programs*, when they say "Bad programming is easy. *Idiots* can learn it in *21 days*, even if they are *dummies*." The Abtruse Goose comic also had their take.

Let's analyze what a title like *Teach Yourself C++ in 24 Hours* could mean:

- **Teach Yourself:** In 24 hours you won't have time to write several significant programs, and learn from your successes and failures with them. You won't have time to work with an experienced programmer and understand what it is like to live in a C++ environment. In short, you won't have time to learn much. So the book can only be talking about a superficial familiarity, not a deep understanding. As Alexander Pope said, a little learning is a dangerous thing.

- **C++:** In 24 hours you might be able to learn some of the syntax of C++ (if you already know another language), but you couldn't learn much about how to use the language. In short, if you were, say, a Basic programmer, you could learn to write programs in the style of Basic using C++ syntax, but you couldn't learn what C++ is actually good (and bad) for. So what's the point? Alan Perlis once said: "A language that doesn't affect the way you think about programming, is not worth knowing". One possible point is that you have to learn a tiny bit of C++ (or more likely, something like JavaScript or Processing) because you need to interface with an existing tool to accomplish a specific task. But then you're not learning how to program; you're learning to accomplish that task.

- **in 24 Hours:** Unfortunately, this is not enough, as the next section shows.

**Translations**

Thanks to the following authors, translations of this page are available in:

Arabic
(Mohamed A. Yahya)

العربية

Bulgarian
(Boyko Bantchev)

Chinese
(Xiaogang Guo)

Croatian
(Tvrtko Bedekovic)

Esperanto
(Federico Gobbo)

# How to start learning Python?

**Ten Years?**

- Don't worry, you don't need to be a "real" programmer.
- Primary focus remains Networking.
- Coding is an additional tool.

# How to start learning Python?

## Ten Years?

- But…this should reassure you.
- You cannot learn Python in 10 minutes.
- Coding makes you feel stupid.
- Don't give up!

# Story Time

## My story:

- No coding background
- No formal IT or computer science education
- No interest in computing outside work
- First networking job - 2001
- CCIER&S 2007 – all Cisco, not code!

# Story Time

## My story:

- Around 2012 & SDN: Networking conversations had little traditional networking.
- All about programmability.
- First heard of APIs.
- I was working on MPLS/BGP network designs.

This is how I felt:

# Story Time

## 2014-2016:

- Knew I wanted to learn coding. But struggled to progress.
- Many false starts, learning the same things over and over.
- No relation to my Networking day job.

# Story Time

## 2017

- Finally 'got it'.
- 30 mins of Python each work day, most work days.
- Reading 'intermediate' Python books.
- Writing code for ArubaOS-Sw & CX APIs.

# Looking back

## What changed?

1. Committed, regular study.
2. 2017 – playing with REST APIs on Aruba switches

**Networking** 🤝 **Python**

# Everybody learns differently

## But don't think about learning Python too much. Just do it. Bloggers are STILL debating the point:

If you're thinking "I should be learning Python", I ask to what goal?

http://ipengineer.net/2017/10/network-automation-engineer-persona-part-two/

Ivan Pepelnjak
@ioshints
**Following**

[Blog] Data Point: Why Automation Won't Replace Humans ift.tt/2MHGjtg

9:06 AM - 18 Sep 2018

4 Likes

Show 332: Don't Believe The Programming Hype

DREW CONRY-MURRAY                              *MARCH 24, 2017*

http://packetpushers.net/podcast/podcasts/show-332-dont-believe-programming-hype/

EtherealMind
@etherealmind
**Follow**

I'm not bashing Python, but only a few people are ever going to make the effort to master it. Tools like @Stack_Storm make it possible for ordinary network professionals to do automation that will get them a pay rise. Whats unique is that a VENDOR is doing this. #NFD17

12:31 AM - 25 Jan 2018

# Everybody learns differently 🐍

**Any encouragement for people to just learn Python means certain responses:**

**Joe Neville**
@joeneville_

Message to any colleagues in networking without coding skills: Don't waste any more thought on whether to learn python. Just start. 🐍

2:33 PM - 18 Oct 2017

**Python is the delivery system.**

**Not the active ingredient.**

# Python is *not* the end game

# Python, CLI, API, SNMP are just tools

# What will Python teach you?

## AKA what do you need to learn as a networker?

- Built-in Data types:
  1. string
  2. integers
  3. lists
  4. dictionaries

```
>>> a = "this is a string"
>>> print(type(a))
<class 'str'>
>>>
>>> b = 99
>>> print(type(b))
<class 'int'>
>>>
>>> my_list = ['clark', 'bruce', 'diana']
>>> print(type(my_list))
<class 'list'>
>>>
>>> my_dict = {'bruce': 'bats'}
>>> print(type(my_dict))
<class 'dict'>
```

# What will Python teach you?

**AKA what do you need to learn as a networker?**

- Statements – "doing things with data":
  if/else, for loops, print
- Scraping data:
  JSON, XML
- Structures for arranging & reusing of code:
  Functions, modules, classes

# What will Python teach you?

## All this is in one book:
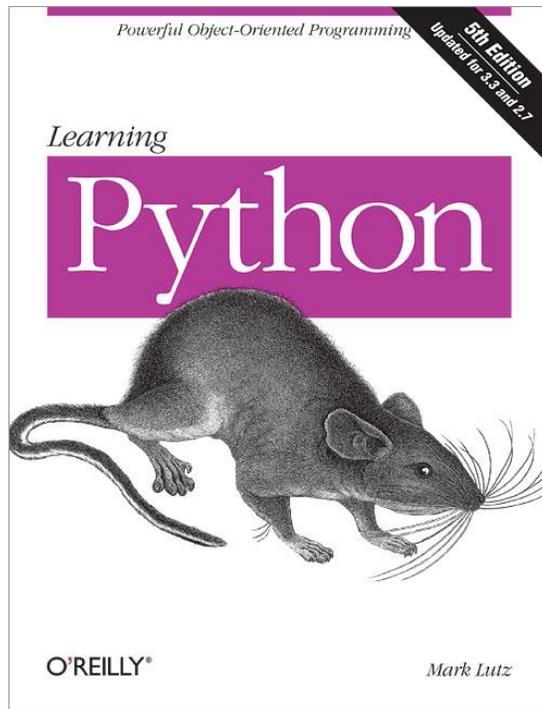
But that book is 1648 pages long.

# Learning Python: Challenges

- **The books and tutorials teach Python, not Python for networkers.**
- **Hard to struggle through examples that seem irrelevant.**
- **String splitting, format command, Unicode…**
- **Hard to retain newly learnt knowledge.**

# Learning Python: My Advice

- **Think of learning Python like exercise (jogging).**
- **At first jogging is hard work, not fun.**
- **Each run is a goal in itself. To go for a run. The End.**
- **Little and often gets results. 3 x 5km/week, much better than 1x15km/week.**


- **Same with Python**

# Learning Python: My Advice

- Short but often periods of study (30mins).
- Don't get bogged down with end goals or career aspirations.
- Just plough through the basics and *keep at it*.

- Making progress through the exercises IS the goal.

# Learning Python: My Advice

- Once you're cool with string vs. integer, list vs. dict vs. tuple, you've got an advantage:
- REST APIs on Aruba switches (Switch & CX)
- You can apply your knowledge
- That's a HUGE PLUS.

**Networking** 🤝 **Python**

# Learning Python: My Advice

- **Combine new Python skills with Networking knowledge.**
- **Start exploring network automation, programmatic approach to networks.**
- **But start small and build.**
- **Cultural hacks.**
- **No POST, only GET.**

**Networking** 🤝 **Python**

# Side point: Learn Git

- If you're committed to learning Python you should invest time in learning git.
- Very difficult at first, becomes a powerful tool.
- Versioning and collaboration.

# Let's Get Started…
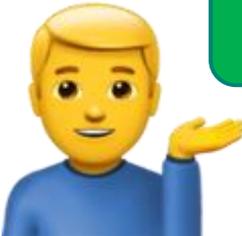
🙋‍♂️ **Python 2?**

# Let's Get Started…

# Let's Get Started…

- Build a Dev Environment
  - Pick an OS (Wins, Mac, Linux distro)
  - Get Python3 installed
  - Pick an IDE (Atom, Visual Studio Code, Pycharm)
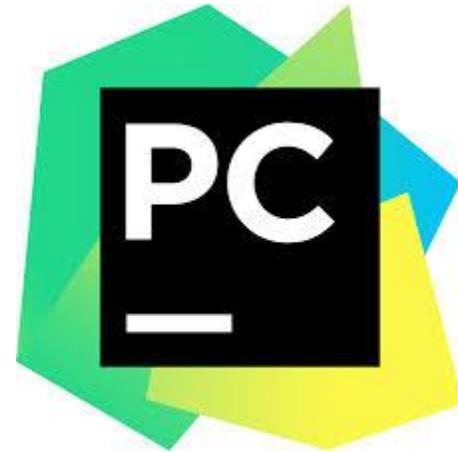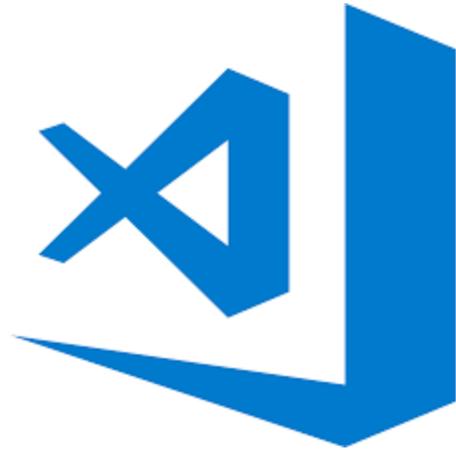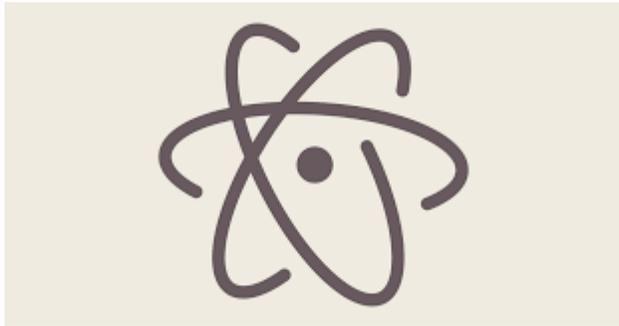
# Let's Get Started…

- **Hit the books (or videos):**
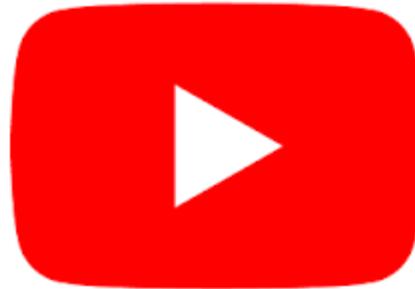
# Let's Get Started…

- Get some kit – ArubaOS-Sw or ArubaOS-CX
- Get the API schema, get on CX Swagger.
- Play around with authentication – login / logout

- Try simple GET calls: VLAN, MAC-table

- Pick a simple config change. Do it with CLI then try to perform it with API.

# Let's Get Started…

**Want some examples?**



**You got it!**

# Let's Get Started…

**Want some sample code for Aruba?**
*Ansible recently posted.*

You got it!
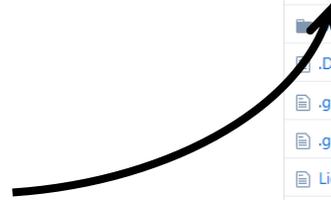


https://github.com/aruba/arubaos-switch-api-python

# Let's Get Started…

- **Want a reading list?**



http://www.nullzero.co.uk/python-reading-list/

# Automation Q & A:
## When questions are statements!

Q: "Isn't automation just for big customers?"
Q: "Shouldn't a network management platform
do all these things?"
Q: "Is this all just the new SDN?"

# Demo Time

**encryptCheck.py**

**Service Desk Ticketing**

**REST API CALLS**

**HTTP**

**HTTP**

**write**

**Audit**

8320-1

8320-2

3810M-1

3810M-2