



Python



But I'm Too Busy to Automate...



The Problem:

*You want to do something useful
(immediately), but you can't do much until
you have the requisite knowledge/skill.*

No one
wants to
learn their
scales

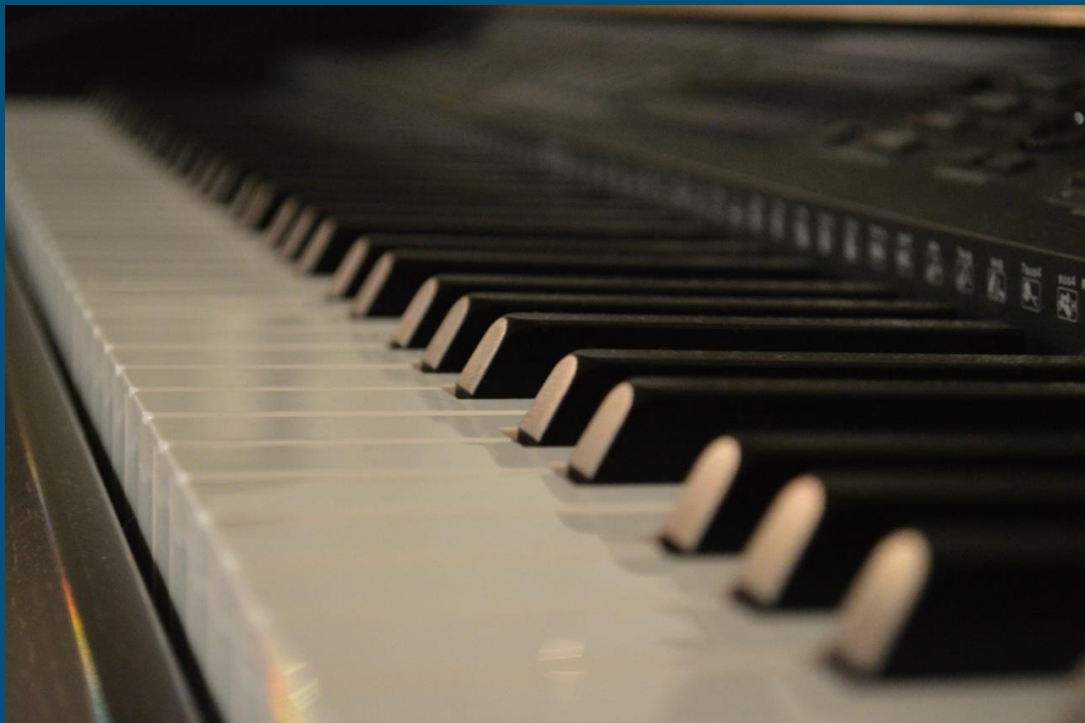


Photo: Joshua Powers (Flickr)

How do we get around this?



Andrew Skudder (Flickr)

Why Python?

But that is the wrong question. A better question is what characteristics do network engineers need in a programming language?

- Widely supported (meaning lots of library support)
- Easily available on systems
- Language accommodates beginners through advanced
- Maintainable
- Allows for easy code reuse
- High-level

Who is this talk for?



Photo: Björn Bechstein (Flickr)

What are some things we want to do?

- Gather information.
- Configure devices.
- Compare configurations to known standards.
- Upgrade devices.
- More holistic monitoring.
- Automatically integrate to other systems.
- Reduce variation.
- Smaller, more frequent changes.
- Less manual work.

Let's roll

1. Learning and doing
2. Get Python
3. <https://repl.it/languages/python>
4. https://github.com/ktbyers/pynet/tree/master/interop_2016/

Example gathering data from router

```
#!/usr/bin/env python
```

```
from netmiko import ConnectHandler
from getpass import getpass
```

```
ip_address = raw_input("Enter IP address: ")
```

```
device = {
    'device_type': 'cisco_ios',
    'ip': ip_address,
    'username': 'pyclass',
    'password': getpass(),
    'port': 22,
}
```

```
net_connect = ConnectHandler(**device)
output = net_connect.send_command_expect('show version')
print output
```

Writing to a file/reading from a file:

```
with open(file_name, "w") as f:  
    f.write(output)
```

```
with open(file_name) as f:  
    output = f.read()
```

High-level

1. Python Characteristics
2. General Items
3. Data Types
4. Conditionals
5. Loops
6. Functions
7. Libraries
8. Back to the beginning
9. Where to go from here



Photo: SpaceX (Flickr)

Python Characteristics

Indentation matters.

Use spaces not tabs.

Python programmers are particular.

Py2 or Py3.

General Items

The Python interpreter shell

Assignment and variable names

Python naming conventions

Printing to standard out/reading from standard in

Quotes, double quotes, triple quotes

comments

dir() and help()

Data Types

- Numbers
- Strings
- Lists
- Booleans/None
- Dictionaries



Photo: J E Theriot (Flickr)

Numbers

```
>>> math.e
```

```
2.718281828459045
```

```
>>> math.pi
```

```
3.141592653589793
```

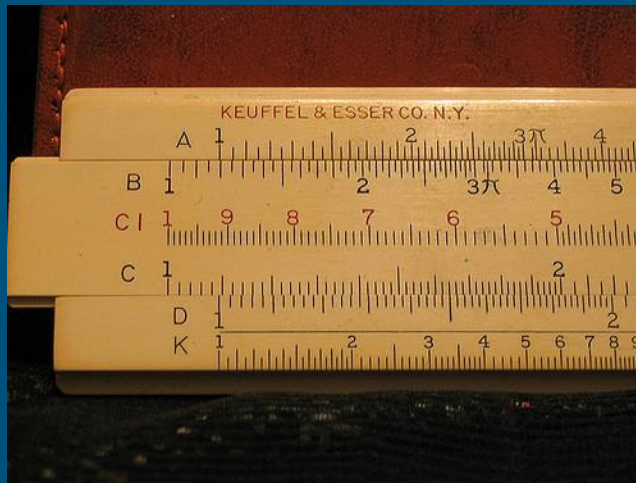


Photo: Cabeel (Flickr)



Photo: Koochbor (Flickr)

Strings

- String methods
- Chaining
- `split()`
- `strip()`
- `substr` in string
- unicode
- raw strings
- `re`

Lists

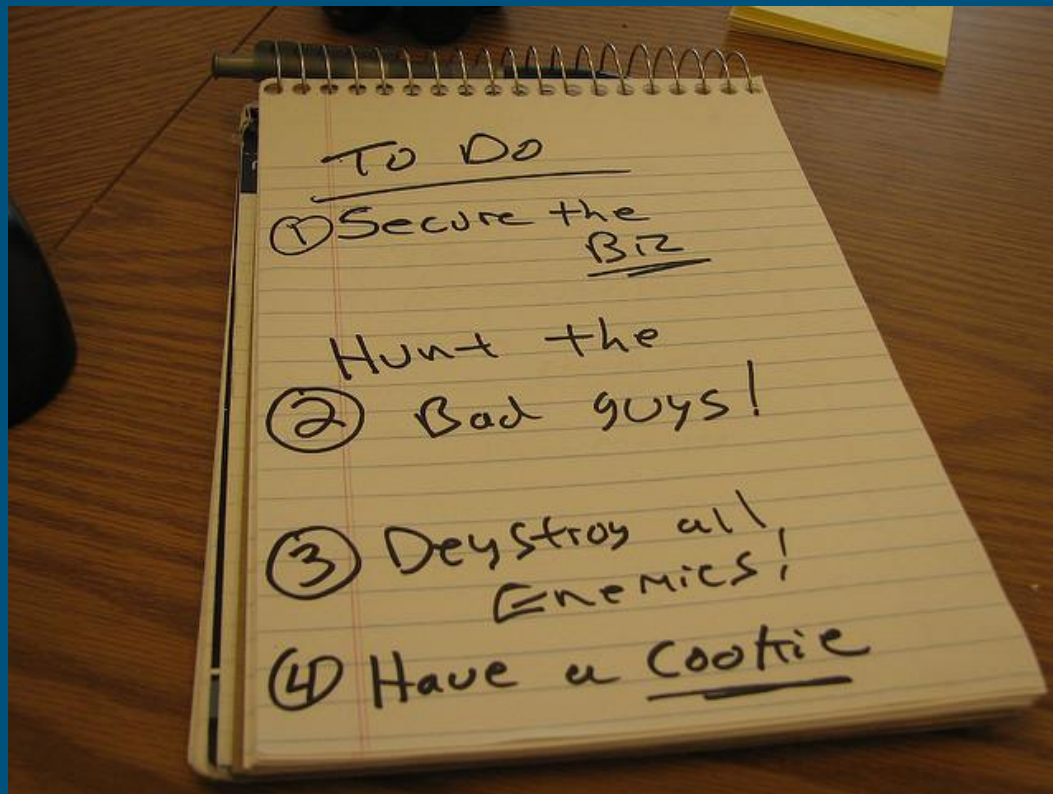
Zero-based indices

.append()

List slices

Tuple

Copying a list



Booleans and None

Boolean operators (and, or, not)

is

Truthy

Comparison operators (==, !=, <, >, >=, <=)

None

Can we do anything useful yet?



Photo: Peter Trimming (Flickr)

Conditionals

```
if a == 15:  
    print "Hello"  
elif a >= 7:  
    print "Something"  
else:  
    print "Nada"
```

Loops

- for
- while
- break
- continue
- range(len())
- enumerate



Photo: Mário Monte Filho (Flickr)

Yes, we can actually do something.



Photo: Nigel Howe (Flickr)

Two last building blocks - dictionaries

- Creating
- Updating
- `get()`
- Iterating over keys
- Iterating over keys and values



Photo: Holger Zscheyge (Flickr)

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

Libraries

Installing libraries

```
import x
```

```
from x import y
```



Photo: Viva Vivanista (Flickr)

Libraries relevant to Network Engineers

ipaddress
pysnmp
paramiko
netmiko
trigger
pexpect
NAPALM
requests
ncclient/pyez
pyeapi
ciscoconfparse
telnetlib
pyserial



Photo: Luis Brizzante (Flickr)

More useful items:

- Quick telnet example
- Quick SNMP example
- Quick email example



Photo: Richard Jones (Flickr)

Back to the beginning

```
#!/usr/bin/env python
```

```
from netmiko import ConnectHandler  
from getpass import getpass
```

```
ip_address = raw_input("Enter IP address: ")
```

```
device = {  
    'device_type': 'cisco_ios',  
    'ip': ip_address,  
    'username': 'pyclass',  
    'password': getpass(),  
    'port': 22,  
}
```

```
net_connect = ConnectHandler(**device)  
output = net_connect.send_command_expect('show version')  
print output
```



Photo: Gordon Tarpley (Flickr)

Where do you go from here?

Write Code / Automate Things

Books:

Learn Python the Hard Way

Treading on Python Volume 1:

Foundations

Online Courses:

Free Python for Network

Engineers

Network to Code Slack Channel



Photo: Cristian Bortes (Flickr)