

## Network Automation:

Network automation is the process of automating the configuring, managing, testing, deploying, and operating of physical and virtual devices within a network. Network automation is a methodology in which software automatically configures, provisions, manages and tests network devices. It is used by enterprises and service providers to improve efficiency and reduce human error and operating expenses. Everyday network tasks and functions are performed automatically. Managing bandwidth and finding fast reroutes to implement the best computing paths. Automation is any process that is self-driven, that reduces and potentially eliminates, the need for human intervention. Automation was once confined to the manufacturing industry. Highly repetitive tasks, such as automobile assembly, were turned over to machines and the modern assembly line was born. Machines excel at repeating the same task without fatigue and without the errors that humans are prone to make in such jobs. Network Automation and Programmability are skills that network engineers of today and the future are going to need to know.

## Python:

Python is widely used to perform network automation. Interacting with network devices. Python is widely used to perform network automation. Python is an open source scripting language, thus used to automate anything. Python is a popular programming language. It was created by Guido van Rossum and released in 1991. Python has a simple syntax similar to the English language. The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular. Python is an interpreted language, which means you just type in plain text to an interpreter, and things happen. There is no compilation step, as in languages such as c. Python is a general-purpose programming language which is dynamically typed, interpreted, and known for its easy readability with great design principles. Python is one of the easier languages to get started with and interpret. Python is an object-oriented programming language. Almost everything in Python is an object, with its properties and methods. A Class is like an object constructor, or a "blueprint" for creating objects. Python is a widely-used, interpreted, object-oriented, and high-level programming language with dynamic semantics, used for general-purpose programming. There are two main kinds of Python, called Python 2 and Python 3. Python 2 is an older version of the original Python. Python 3 is the newer (to be precise, the current) version of the language. It's going through its own evolution path, creating its own standards and habits.

### String:

A string is simply one or more alphanumeric characters. A string can comprise many numbers or letters, depending on the Python version in use. String literals in python are surrounded by either single quotation marks, or double quotation marks. 'hello' is the same as "hello". You can display a string literal with the print() function:

### Comments:

Comments start with a #, and Python will render the rest of the line as a comment:

```
#This is a comment  
print("Hello, World!")
```

### Variables:

Variables are containers for storing data values. Unlike other programming languages, Python has no command for declaring a variable. A variable is created the moment you first assign a value to it.

```
x = 6  
y = "ali"  
print(x)  
print(y)
```

### Data Types:

In programming, data type is an important concept. Variables can store data of different types, and different types can do different things. Python has the following data types built-in by default, in these categories:

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview

```
x = 6  
print(type(x))
```

### Print Function:

The print() function is a built-in function. It prints/outputs a specified message to the screen/console window.

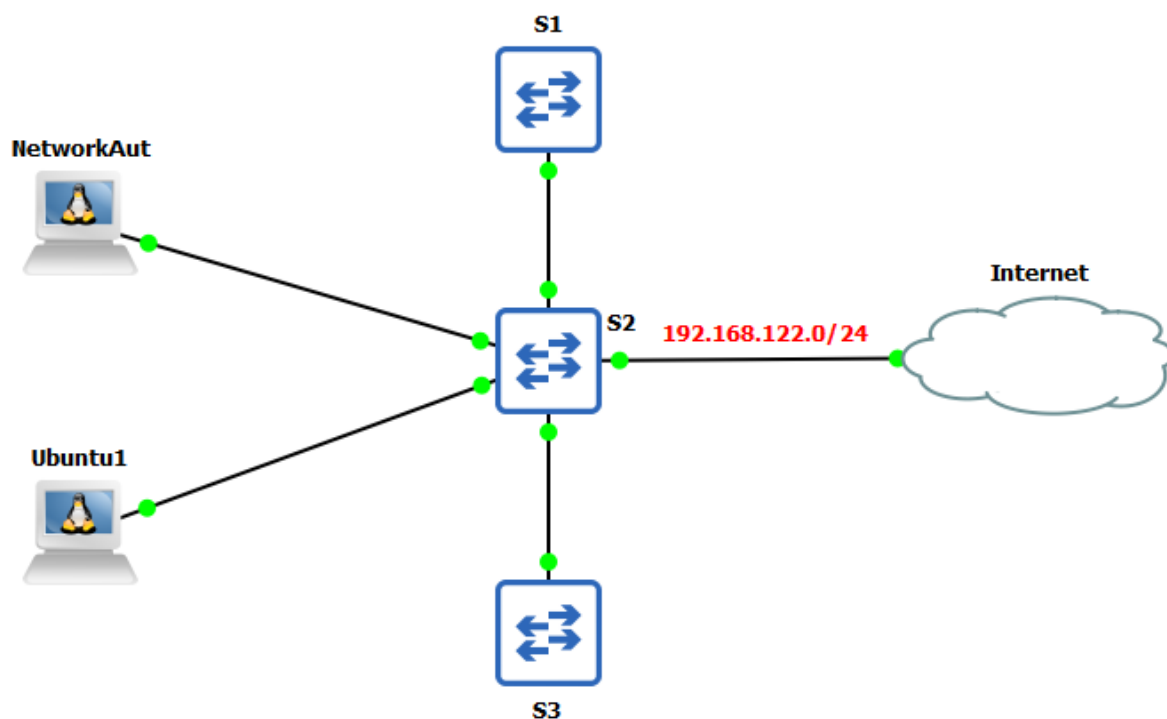
```
print("Hello Network Engineers")
```

### Numeric Type:

There are three numeric types in Python: Int, float and complex. Variables of numeric types are created when you assign a value to them. To verify the type of any object in Python, use the type() function: integer are numbers written without a fractional component. Floating are numbers that contain or are able to contain a fractional component.

```
x = 1 # int
y = 2.8 # float
print(type(x))
print(type(y))
```

### Lab Time:



NetworkAut and Ubuntu1 PC configuration	# DHCP config for eth0 auto eth0 iface eth0 inet dhcp
S1 Switch Configuration	S1(config)#interface vlan 1 S1(config-if)#ip address 192.168.122.101 255.255.255.0 S1(config-if)#no shutdown
S2 Switch Configuration	S2(config)#interface vlan 1 S2(config-if)#ip address 192.168.122.102 255.255.255.0 S2(config-if)#no shutdown
S3 Switch Configuration	S3(config)#interface vlan 1 S3(config-if)#ip address 192.168.122.103 255.255.255.0 S3(config-if)#no shutdown

### Switch 1 Telnet Configuration

```
S1(config)#enable password secret
S1(config)#username admin password cisco
S1(config)#line vty 0 4
S1(config-line)#login local
S1(config-line)#transport input all
S1(config-line)#exit
```

### Switch 2 Telnet Configuration

```
S2(config)#enable password secret
S2(config)#username admin password cisco
S2(config)#line vty 0 4
S2(config-line)#login local
S2(config-line)#transport input all
S2(config-line)#exit
```

### Switch 3 Telnet Configuration

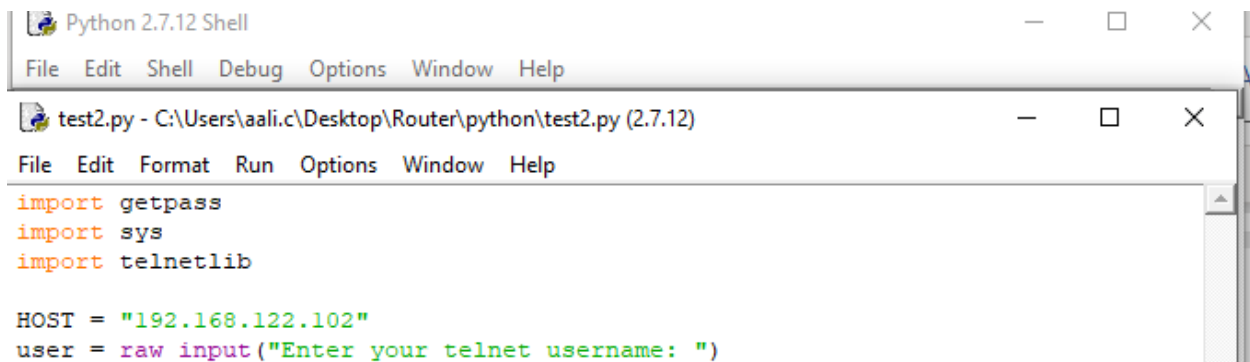
```
S3(config)#enable password secret
S3(config)#username admin password cisco
S3(config)#line vty 0 4
S3(config-line)#login local
S3(config-line)#transport input all
S3(config-line)#exit
```

### Ubuntu PC

```
root@Ubuntu1:~# apt-get update
root@Ubuntu1:~# apt-get install python
root@Ubuntu1:~# python --version
root@Ubuntu1:~# apt-get install vim
root@Ubuntu1:~# python c.py
```

### NetworkAuto PC

```
root@NetworkAut:~# apt-get update
root@NetworkAut:~# python --version
root@NetworkAut:~# apt-get install vim
root@NetworkAut:~# python a.py
```



The image shows two overlapping windows. The top window is titled 'Python 2.7.12 Shell' and has a menu bar with 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The bottom window is titled 'test2.py - C:\Users\aaali.c\Desktop\Router\python\test2.py (2.7.12)' and has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code in the bottom window is as follows:

```
import getpass
import sys
import telnetlib

HOST = "192.168.122.102"
user = raw_input("Enter your telnet username: ")
```