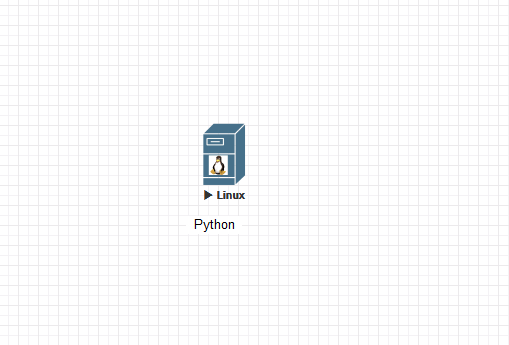
**Discovery 28: Writing and Troubleshooting Python Scripts**

**Topology**



**Introduction**

Your configuration tasks are as follows:

Write a Hello World Script

Write a Network Script

Analyze and troubleshoot a script

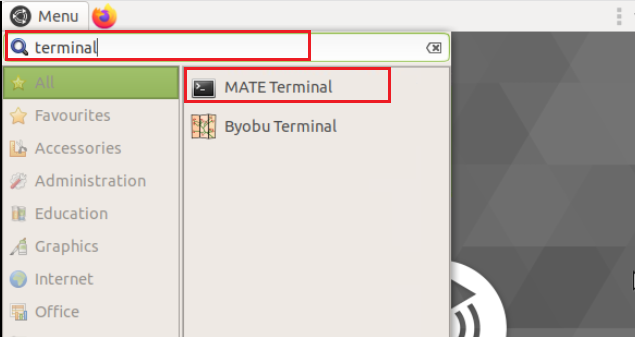
**Note**

Actual lab devices are not used in this activity. This activity is a simulation based on a series of tasks. Often, it is not always possible to provide real lab equipment because of the nature and type of technology. These lab simulations are based on real equipment and actual lab tasks. The labs are performed using a simulation of real equipment. There are no setup or initialization time requirements, and the simulation is available immediately.

**Task 1: Writing a Hello World Script**

**Step 1:** Search for terminal and select MATE Terminal from the Student Workstation. Then navigate to the **files/lab5** directory.

Double-click the MATE Terminal option:



In the Linux terminal window, enter the following information:



Create a new script called **welcome.py** and save it in the lab5 directory. When executed, this script should print one line, **“Welcome to the world of Cisco Network Programmability!”**

**Step 2:** Using the **touch** command create a file called **welcome.py** and then use the **gedit** command to open the file in the editor:

Enter the following commands:



**Step 3:** The editor will appear. Within the editor, add the following lines of text to your script and save it so it can be executed.

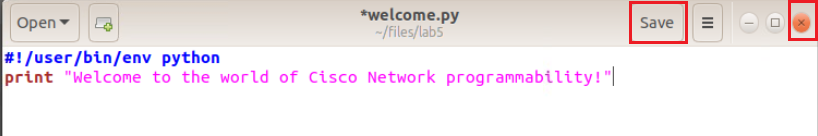
From the editor window, enter the following information:

#!/usr/bin/env python

print "Welcome to the world of Cisco Network Programmability!"

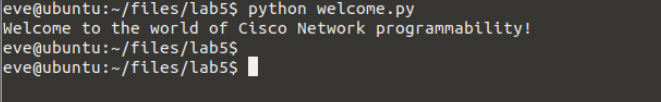
**Step 4:** Click the **Save** option at the upper right. Then, click the **"x"** option at the upper right of the window to close the editor. The terminal window will be displayed.

Click the options below:



**Step 5:** From the terminal window, execute the script by issuing the **python welcome.py** command.

Enter the following commands:

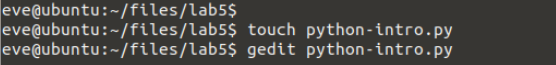


If the script was correctly written, you should see the line **“Welcome to the world of Cisco Network Programmability!.”**

**Task 2: Writing a Network Script**

**Step 1:** In the **lab5** directory, using the **touch** and **gedit** commands, create a new script called **python-intro.py** and then open the editor to create the new file. The file will have several sections and will be created in the next few steps.

Enter the following information:



**Step 2:** After you enter the **gedit** command, the editor window will open. Create a dictionary called **facts\_1** with the following key value pairs:

|  |  |
| --- | --- |
| Keys | Values |
| os | 7.2 |
| FQDN | networks.com |
| location | sjc |
| vlans\_list | [1, 5,10] |
| neighbors | ['sw2', 'sw3'] |

Enter the following information:

facts\_1 = {

'os': '7.2',

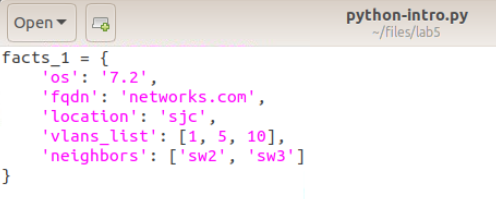
'fqdn': ‘networks.com',

'location': 'sjc',

'vlans\_list': [1, 5, 10],

'neighbors': ['sw2', 'sw3']

}



**Step 3:** Create a dictionary called **facts\_2** with the following key value pairs:

|  |  |
| --- | --- |
| Keys | Values |
| os | 7.2 |
| FQDN | networks.com |
| location | syd |
| vlans\_list | [1, 10, 20] |
| neighbors | ['sw1', 'sw3'] |

Enter the following information:

facts\_2 = {

'os': '7.2',

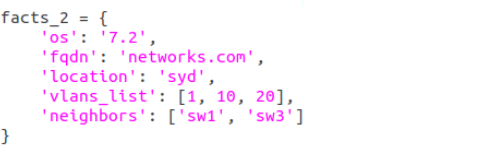
'fqdn': ‘networks.com',

'location': 'syd',

'vlans\_list': [1, 10, 20],

'neighbors': ['sw1', 'sw3']

}



**Step 4:** Create a dictionary called **facts\_3** with the following key value pairs:

|  |  |
| --- | --- |
| Keys | Values |
| os | 7.2 |
| FQDN | networks.com |
| location | nyc |
| vlans\_list | [1, 4, 20] |
| neighbors | ['sw1', 'sw2'] |

Enter the following information:

facts\_3 = {

'os': '7.2',

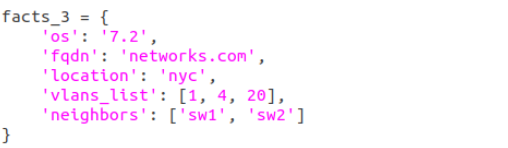
'fqdn': ‘networks.com',

'location': 'nyc',

'vlans\_list': [1, 4, 20],

'neighbors': ['sw1', 'sw2']

}



**Step 5:** Create a dictionary called **facts** with the following key value pairs:

|  |  |
| --- | --- |
| Keys | Values |
| sw1 | facts\_1 |
| sw2 | facts\_2 |
| sw3 | facts\_3 |

Enter the following information:

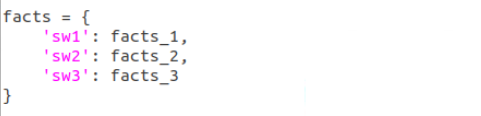
facts = {

'sw1': facts\_1,

'sw2': facts\_2,

'sw3': facts\_3

}



**Step 6:** Enter the commands to instruct the script to print all facts for **sw1**.

Enter the following information:

print 'sw1 facts:'

print facts['sw1']



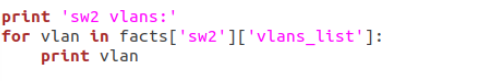
**Step 7:** Enter the commands to instruct the script to print the list of VLANs for **sw2** and ensure that a **for loop** is used.

Enter the following information:

print 'sw2 vlans:'

for vlan in facts['sw2']['vlans\_list']:

print vlan

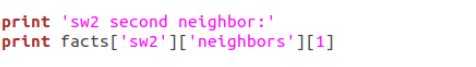


**Step 8:** Enter the commands to instruct the script to print **sw2’s** second neighbor.

Enter the following information:

print 'sw2 second neighbor:'

print facts['sw2']['neighbors'][1]



**Step 9:** Enter the commands to instruct the script to print **sw3’s** first neighbor.

Enter the following information:

print 'sw3 first neighbor:'

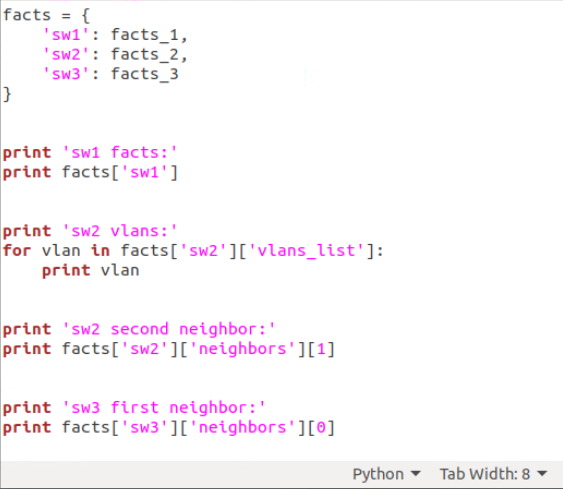
print facts['sw3']['neighbors'][0]



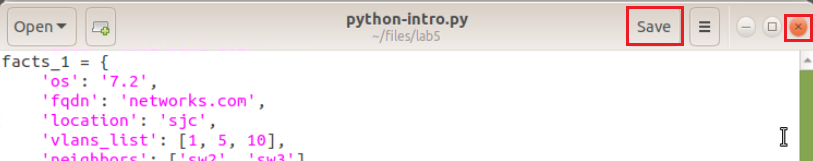
**Step 10:** Verify that the script has been written correctly. The first four sections of the script have created lists of facts. You then instructed the script to print information that is contained in those facts when the script is run.

The final version of the script should look like the following:



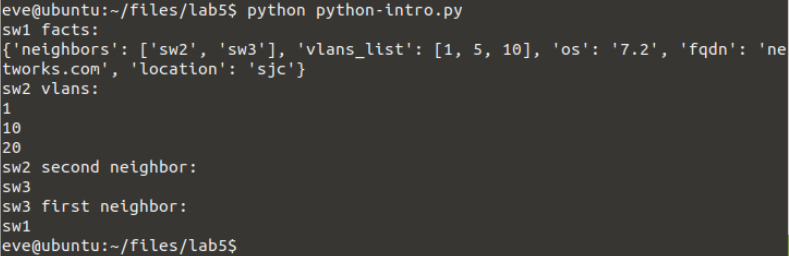


**Step 11:** Click the **Save** option at the upper right. Then, click the **"x"** option at the upper left of the window to close the editor. The terminal window will be displayed.

****

**Step 12:** Using the **python** command, execute the script from the Linux terminal.

Using the appropriate language constructs should give you the following result in your terminal:

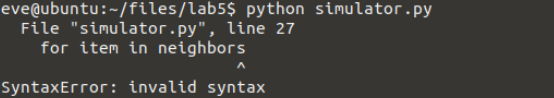


The first part of the results shows several of the facts from sw1, the second part show sw2’s VLANs, the third part shows sw2’s second neighbor, and the last part shows sw3’s first neighbor. This example showed how you can extrapolate information from a list using a script.

**Task 3: Analyzing and Troubleshooting a Script**

**Step 1:** From the Linux terminal on the student PC, run the **simulator.py** program (in the lab5 directory) and observe the results.

Enter the following information:



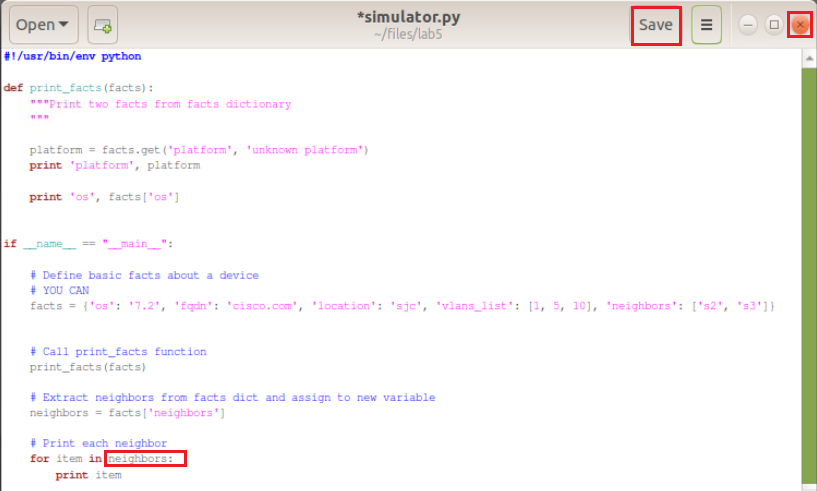
The script fails to run due to an error in line number 27 “for item in neighbors,” and the “^” indicates that the error is at the end of the line.

**Step 2:** Using the **gedit** command, fix the script so it runs smoothly. Open the script in the editor and fix line 27. Line 27 is missing a **“:”**at the end of the line.

Enter the following information:



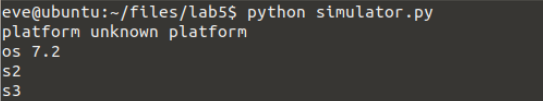
The response should look like the following:

****

**Step 3:** Click the **Save** option at the upper right. Then, click the **"x"** option at the upper left of the window to close the editor. The terminal window will be displayed.

**Step 4:** Using the **python** command, execute the script.

Enter the following information and review the output.



Once the script successfully runs, you have completed this discovery.