Городецкий Константин Павлович

От:	Константин Городецкий <gorodetskiykp@gmail.com>
Отправлено:	16 февраля 2018 г. 7:33
Кому:	Городецкий Константин Павлович
Тема:	Fwd: [PyNet Learning Python 2018] - Lesson3 / Conditionals and Loops
Категории:	ОИТ

---------- Forwarded message ----------
From: **ktbyers@twb-tech.com** <ktbyers@twb-tech.com>
Date: 2018-02-16 2:01 GMT+10:00
Subject: [PyNet Learning Python 2018] - Lesson3 / Conditionals and Loops
To: gorodetskiykp@gmail.com

Konstantin

Note: There is a table of contents for each video at the bottom of this email including timestamps to where various content is located. This should be helpful in navigating the videos.

In this email of Learning Python we are going to cover the following:

1. **Conditionals**
Video https://vimeo.com/245104620
Length is 8 minutes

2. **Boolean Logic (Booleans, Ternary Operator, None)**
Video https://vimeo.com/245112558
Length is 8 minutes

3. **Python For Loops**
Video https://vimeo.com/245466297
Length is 5 minutes

4. **For Loops (Enumerate)**
Video https://vimeo.com/245477015
Length is 6 minutes

5. **For Loops (Break and Continue)**
Video https://vimeo.com/245478016
Length is 9 minutes

6. **While Loops**
Video https://vimeo.com/245545155
Length is 5 minutes

7. **Loops Miscellaneous**
   Video https://vimeo.com/245552604
   Length is 6 minutes

Additional Content:

How To Write Conditional Statements in Python 3

How To Construct For Loops in Python 3

How To Construct While Loops in Python 3

Exercises

Reference code for these exercises is posted on GitHub at:
https://github.com/ktbyers/pynet/tree/master/learning_python/lesson3

1. Read the "show_vlan.txt" file into your program. Loop through the lines in this file and extract all of the VLAN_ID, VLAN_NAME combinations. From these VLAN_ID and VLAN_NAME construct a new list where each element in the list is a tuple consisting of (VLAN_ID, VLAN_NAME). Print this data structure to the screen. Your output should look as follows:

```
[('1', 'default'),
 ('400', 'blue400'),
 ('401', 'blue401'),
 ('402', 'blue402'),
 ('403', 'blue403')]
```

2. Read the contents of the "show_arp.txt" file. Using a for loop, iterate over the lines of this file. Process the lines of the file and separate out the ip_addr and mac_addr for each entry into a separate variable.

Add a conditional statement that searches for '10.220.88.1'. If 10.220.88.1 is found, print out the string "Default gateway IP/Mac" and the corresponding IP address and MAC Address.

Using a conditional statement, also search for '10.220.88.30'. If this IP address is found, then print out "Arista3 IP/Mac is" and the corresponding ip_addr and mac_addr.

Keep track of whether you have found both the Default Gateway and the Arista3 switch. Once you have found both of these devices, 'break' out of the for loop.

3. Read the 'show_lldp_neighbors_detail.txt' file. Loop over the lines of this file. Keep reading the lines until you have encountered the remote "System Name" and remote "Port id". Save these two items into variables and print them to the screen. You should extract only the system name and port id from the lines (i.e. your variables should only have 'twb-sf-hpsw1' and '15'). Break out of your loop once you have retrieved these two items.

4. You have the following data structure:

```
arp_table = [('10.220.88.1', '0062.ec29.70fe'),
 ('10.220.88.20', 'c89c.1dea.0eb6'),
 ('10.220.88.21', '1c6a.7aaf.576c'),
 ('10.220.88.28', '5254.aba8.9aea'),
 ('10.220.88.29', '5254.abbe.5b7b'),
 ('10.220.88.30', '5254.ab71.e119'),
 ('10.220.88.32', '5254.abc7.26aa'),
 ('10.220.88.33', '5254.ab3a.8d26'),
 ('10.220.88.35', '5254.abfb.af12'),
 ('10.220.88.37', '0001.00ff.0001'),
 ('10.220.88.38', '0002.00ff.0001'),
 ('10.220.88.39', '6464.9be8.08c8'),
 ('10.220.88.40', '001c.c4bf.826a'),
 ('10.220.88.41', '001b.7873.5634')]
```

Loop over this data structure and extract all of the MAC addresses. Process all of the MAC addresses to get them into a standard format. Print all of the new standardized MAC address to the screen. The standardized format should be as follows:

00:62:EC:29:70:FE

The hex digits should be capitalized. Additionally, there should be a colon between each octet in the MAC address.

5. **[Optional/bonus]**

***** Note, to actually test this in your environment, change the test IP addresses to something in your environment that you can ping successfully. *****

Construct a list of 254 IP addresses. The base IP address should be equal to '10.10.100.0' or '10.10.100.'.

You should use the 'range' builtin to accomplish this.

Your list should have all of the IP addresses from 10.10.100.1 to 10.10.100.254.

Use Python's 'enumerate' to print out all of the IP addresses and their corresponding list index. The output should look similar to the following:

```
0 ---> 10.10.100.1
1 ---> 10.10.100.2
2 ---> 10.10.100.3
3 ---> 10.10.100.4
4 ---> 10.10.100.5
...
```

Use a list slice to create a new list that goes from 10.10.100.3 to 10.10.100.6.

Using a loop and os.system("ping -c 3 10.10.100.3") try pinging all of the IP addresses in this short list. For Windows the command will probably be os.system("ping -n 3 10.10.100.3").

Put a variable at the top to define whether you are using Windows or Linux/MacOs. This should be similar to the following:

```
WINDOWS = False

base_cmd_linux = 'ping -c 2'
base_cmd_windows = 'ping -n 2'
# Ternary operator
base_cmd = base_cmd_windows if WINDOWS else base_cmd_linux
```

CLASS OUTLINE

1. Conditionals (VIDEO1)
   A. Structure of python conditionals   [0:10]
   B. What is an expression in Python   [1:54]
   C. Example of if statement [2:47]
   D. Example of if statement using a variable   [3:35]
   E. Example of if/else statement   [5:08]
   F. Example of if/elif/else statement   [5:43]

2. Boolean Logic (Booleans, Ternary Operator, None) (VIDEO2)
   A. Boolean logic
      1. True and True (logical and)   [0:41]
      2. True or False (logical or)   [1:06]
      3. not True   [1:17]
   B. Constructing more complex expressions   [1:27]
   C. Casting other data types to be booleans   [2:24]
   D. How Python evaluates other data types   [2:39]
   E. Typecasting with bool   [4:10]
   F. Python ternary operator   [5:00]
   G. None   [7:22]
      1. Default return value for Python functions   [7:46]

3. Python For Loops (VIDEO3)
   A. General meaning of loops   [0:03]
   B. Meaning of for loops   [0:29]
      1. Example for loop   [1:25]

4. For Loops (Enumerate) (VIDEO4)
   A. Value of the loop-variable at the end of the loop   [0:31]
   B. Tracking both the index and the element   [0:48]
   C. Using enumerate   [1:30]
   D. Assigning a tuple to multiple variables   [3:30]
   E. Do not modify a list while looping over the list   [5:38]

5. For Loops (Break and Continue) (VIDEO5)
   A. Other ways to exit the for loop   [0:24]
      1. break   [0:30]
      2. continue   [1:47]
   B. pass as a no-op   [3:24]
   C. Nesting for loops   [4:12]
   D. The interable does not need to be a list   [6:22]
      1. It can be a tuple
      2. It can be a dictionary
      3. It can be a string
   E. Range   [7:09]
   F. Using 'in' for list membership   [8:57]

6. While Loops (VIDEO6)
   A. Format of a while loop   [0:09]
   B. Exit from a while loop [0:39]

```
   C. Example of a while loop   [1:09]
   D. Infinite loop   [2:45]

7. Loops Miscellaneous (VIDEO7)
   A. Meaning of else   [1:04]
   B. use variables instead of indices for readability   [3:00]
   C. while True:   [5:00]
      1. break is the only way out   [5:36]
```

Kirk Byers