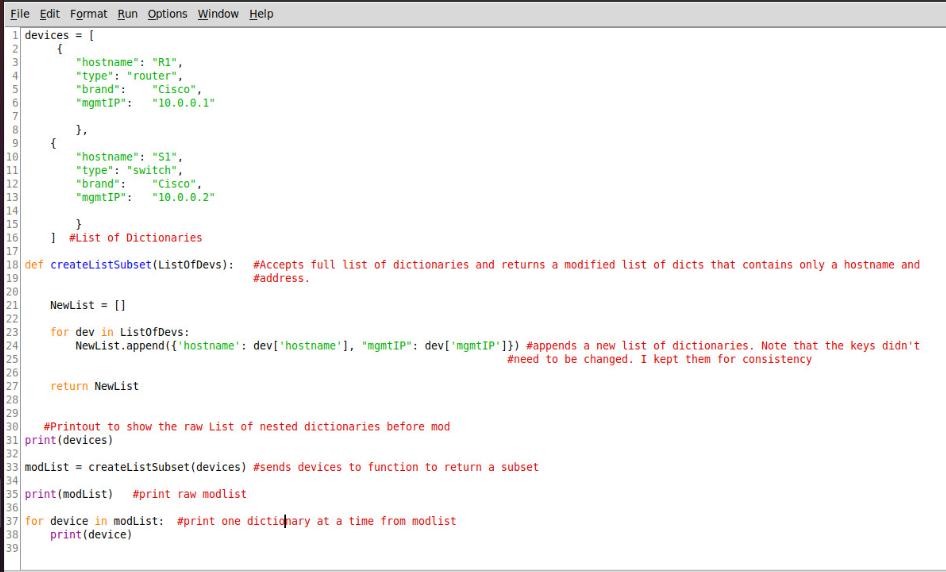
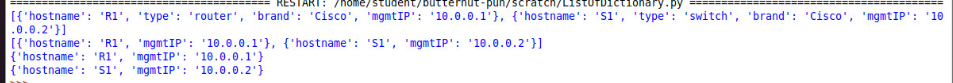
**Unit Ten Lab**

Directions: Complete each of the scripts below and submit them by the date specified in the assignment sheet and Blackboard. Submit **copies of your code** and **screenshots of the code running with each task**. Also, be sure to use the document, **Script Requirements as a guide** to writing good code. **Full credit will not be earned if you do not meet these script requirements.** **40 points**

1. Practice creating a List of dictionaries as in the following:



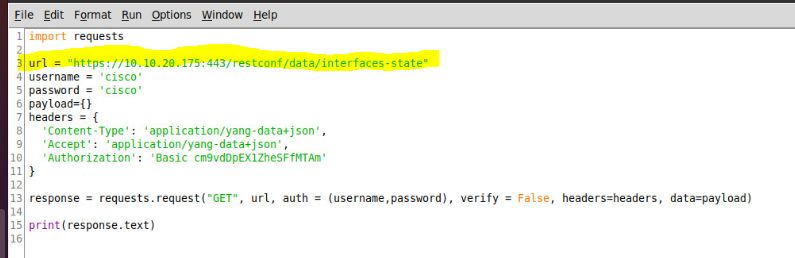


The output corresponds to print statements on lines 31, 35, and 38, respectively. Note that the CreateListSubset() function above takes a list of dictionaries and returns a modified list of dictionaries. You will do something like this in numbers 2 and 3 below.

1. Write a script that give a list of interfaces on an IOSXE device and shows the interface name, the IP address and the MAC address for each interface in a table. Remember to skip the Loopback address that is defined but contains no IP address. Hint: Use the type key in the interface dictionary to only gather data on “EthernetCsmacd” values for the type key. The table will layout as follows:

Int IP Physical

To accomplish this, you will make two RESTCONF API requests to your IOSXE device. The first will be like the one from your lab this week that called your interfaces using the ietf-interfaces:interfaces module, and the second will use the interfaces-state module as shown below:



You will use functions to accomplish this. Your main script must look something like the following:

import requests

import json

ipAddr = ’10.10.20.175’

#Returns a list of Dictionaries, one for each interface, that contains only an interface #name and IP address

intList = getIntRest(ipAddr)

#Returns a list of Dictionaries, one for each interface, that contains only an interface #name and MAC address

intStateList = getIntRestMAC(ipAddr)

#Combines both lists of dictionaries into one List of dictionaries, one for each interface, #that contains an Interface name, IP Address, and Mac Address

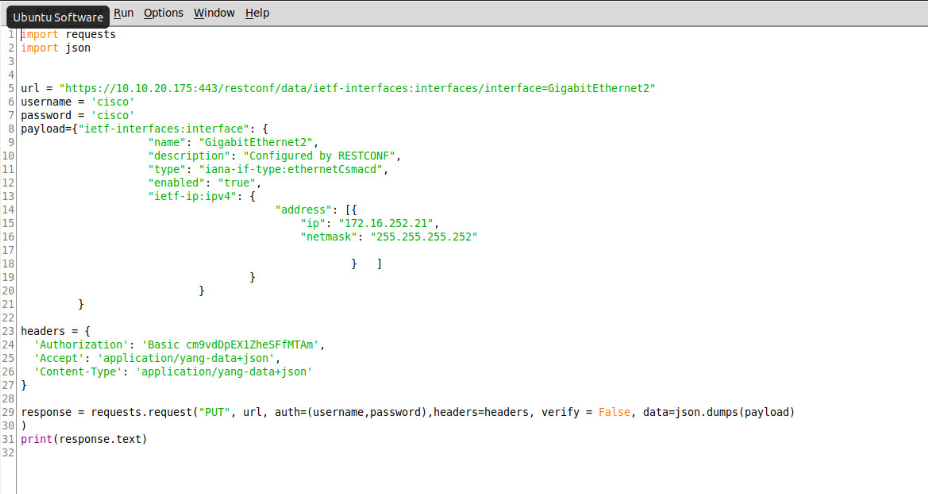
combinedIntList = combineIntLists(intStateList, intList)

#Prints combined list in desired format

printList(combinedList)

You will write all the functions that accomplish the tasks shown above.

1. Below is an example of how to change an IP address on an IOSXE device. This code is in the turnipTheBeet repository as ChangeAddressYang.py. If you do not see it, git pull the updates to download the script below. Note the headers dictionary is the same as previous examples from earlier this week:

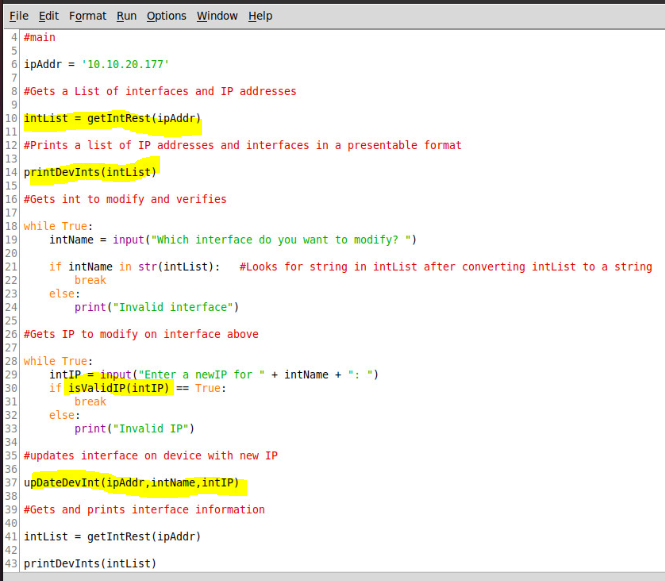


A screen shot of a computer

Description automatically generated with medium confidence

Write a script that prints the interfaces and IP addresses. Then have it ask the user which interface they would like to modify. Ask the user for an IP address, validate it, and then have it change the IP address on that interface. Finally, have the script re-print the interfaces with the IP addresses.

Your main must be modularized as follows (You can add functions, but you cannot subtract them):



Note that the highlighted functions will be written by you. Some may already be written in previous labs.

A picture containing text, screenshot

Description automatically generated

Lastly, while testing this, you will need to manually remove the IP address with the CLI. Each new address you add to an interface, while testing, will add as a new primary address and the old tests will be moved to a secondary address. This can cause issues if you try to add an address on the same network as an earlier secondary address. So, for now, make sure that the new address being entered is on a different network. You don’t need your code to verify this, but you will want to ensure you do this while testing. The fix for this is using a different interface YANG model, the Cisco-IOS-XE-native model, which we will use next week.