**Unit Eleven 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. Run the NETCONF-GET-RUN2-INT.py script.

Graphical user interface, text, application, email

Description automatically generated

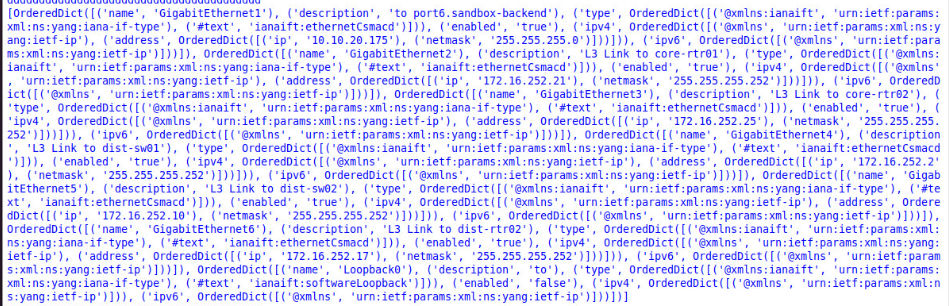
1. We are getting the running config, but filtering it using an XML filter so that we only see the interfaces. Line 23 makes the connection, using the parameters in the dictionary, and line 25 makes the call with the xml filter we have defined. Note the indentation for calls made inside the connection. The connection is automatically closed when we use the “with” on line 23. Once all lines have executed in the indented code, the connection closes nicely. The remaining lines display the data (a list of interfaces in an ordered dictionary) in various formats with separators for reference. Line 27, prints the XML in a readable format:



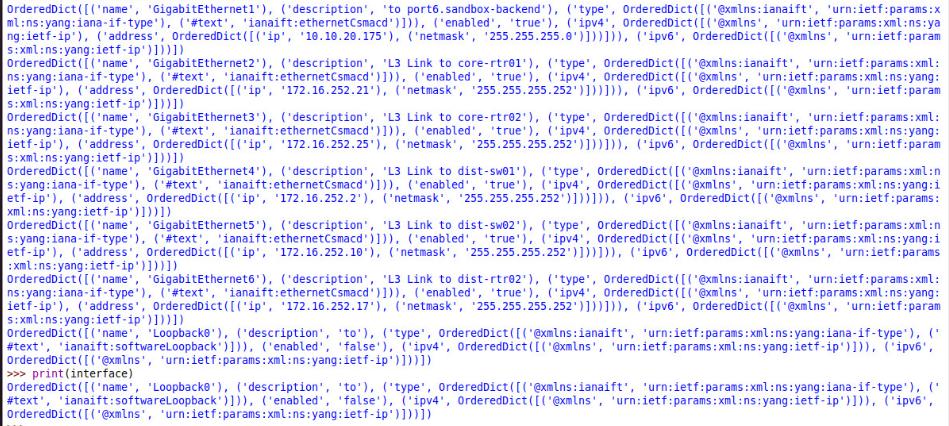
Line 33 parses the XML to an ordered dictionary:



Line 41 creates a list of dictionaries, each one, representing an interface:



Finally, lines 47 and 48 iterate the list, separating the dictionaries, printing each dictionary separately:



1. **Write a script that makes a NETCONF call to an IOSXE device and displays a list of interfaces, their IP addresses, subnet masks, and descriptions in a human-readable display table, similar to a show ip interface brief.**
2. **Modify the script and ask your user which interface they would like to change, and then ask them for an IP address, subnet mask, and description. Check the address and mask for validity, and then update the ordered dictionary. Once you have done this, re-print the dictionary as in step 3. Note that you will not actually change the address of the device yet. Like last week, your code should be heavily modularized. Your NETCONF call, validity checking, and printing should be in functions.**