1. **Lesson3 Introduction**  
   Video [**https://vimeo.com/823083903**](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84MjMwODM5MDM_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.3iTpRS54VZuX47U4XizxfhNI6OxoWDmCzhCub5zZ8FA)  
   Length is 1 minute
2. **Conditionals**  
   Video [**https://vimeo.com/823096706**](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84MjMwOTY3MDY_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.5yWJ5oXUpOd5IEt45TjCrY0rLT0nAmq0VE4apktmc2M)  
   Length is 6 minutes
3. **For Loops**  
   Video [**https://vimeo.com/823176272**](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84MjMxNzYyNzI_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.TI2GWKTJrBVX4UFEqCLT6AncaumutPquJjOwfYPnLro)  
   Length is 9 minutes
4. **While Loops**  
   Video [**https://vimeo.com/823499732**](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84MjM0OTk3MzI_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.iCOkZ8VP2v6z5gwXK39ypOI7QX18mNutk8QhlTSRcVM)  
   Length is 5 minutes
5. **​** **List Comprehensions**(\*optional/intermediate content)  
   Video [**https://vimeo.com/825283709**](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84MjUyODM3MDk_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.JpMTjTD-RZ4HoviMZSLQ7QoOkTOWMI9laSn0t9jnxZI)  
   Length is 5 minutes
6. **Generator Expressions**(\*optional/intermediate content)  
   Video [**https://vimeo.com/825313173**](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84MjUzMTMxNzM_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.4TdHZjlcsqTqUwO5QDhupFY3fAP_No-F_SvTCA2JrO8)  
   Length is 6 minutes

**\* Optional/intermediate content - this content is more intermediate and can be safely skipped.**In other words, feel free to skip this content if you are time limited or are struggling with the more fundamental content.  
  
  
  
  
**Collateral Material (programs used in the videos):**  
  
*In collateral, we try to provide the Python scripts and related files shown in the videos. In other words, we try to make it easy for you to reproduce the examples from the videos.*  
  
[Lesson3 Collateral Programs](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..6-BfLaWywjwgnvCdtQNoyMTU1i1YvLy44ucEc8yDXUo)  
  
  
  
  
**Additional Content:**  
  
**Conditionals**  
  
[Nick Parlante's Python Guide: If and Comparisons](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2NzLnN0YW5mb3JkLmVkdS9wZW9wbGUvbmljay9weS9weXRob24taWYuaHRtbD9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.i4b052LW69-4ZgkO4GBAnNBcmisfTw7CkmKMpIuUHEE)  
Notes from a Stanford Computer Science Lecturer. This section covers conditionals, comparison operators, and "truthy". The content is pretty similar to the content in my videos.  
  
  
**For Loops**  
  
[Nick Parlante's Python Guide: For Loop](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2NzLnN0YW5mb3JkLmVkdS9wZW9wbGUvbmljay9weS9weXRob24tZm9yLmh0bWw_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.9B-T6yoU4pnHzLy3MP139ZVoD1HRxsbn-1Jt4jU-GSI)  
Notes from a Stanford Computer Science Lecturer. This section is similar to the content in my videos. It does bring up the important idea that you should not modify the collection that you are looping over. It also references what happens if you loop over a null collection (for example, an empty list). In the final section, "Standard Loop Phrases", subsection 3, he uses the pattern range(len(some\_list)) to obtain the list indices--a better pattern for this is to use the 'enumerate' built-in.  
  
  
**While Loops**  
  
[Nick Parlante's Python Guide: While Loop](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2NzLnN0YW5mb3JkLmVkdS9wZW9wbGUvbmljay9weS9weXRob24td2hpbGUuaHRtbD9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.pASd1Zr4M3KpNNB5fAlrHSRTN_JX9axtTPYdc-yPAF0)  
Notes from a Stanford Computer Science Lecturer. He covers while loop syntax, incrementing a loop counter, accidentally creating infinite loops, and using "while True".  
  
  
**Flow Control (conditionals and loops)**  
  
[Documentation from python.org on flow-control](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2RvY3MucHl0aG9uLm9yZy8zL3R1dG9yaWFsL2NvbnRyb2xmbG93Lmh0bWw_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.EKO9607-RMoPXUdoRw2axl5NufcFFKyrrDoHx73_xbg)  
Sections 4.1 through 4.5 cover conditionals, for-loops, range, break/continue/else, and the pass statement.  
  
  
**List Comprehensions**  
  
[Digital Ocean Article: Understanding List Comprehensions in Python3](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..cUqVKZ1CY4-ebfY29stFMUO6Uo72lGZ_A4feLy9rTOE)  
Good article talking about the purpose and syntax of list comprehensions. It also provides various examples including list comprehensions that use conditionals and list comprehensions that use nested for loops.  
  
  
  
  
**Exercises**  
  
Reference code for these exercises is posted on GitHub at:  
    [https://github.com/twin-bridges/learning\_python/tree/main/lesson3/exercises](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU2ODEzNzQyLCJuYmYiOjE3NTY4MTM3NDIsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJ4ejYwd3lwd3ZxcjNvOXAyMDdjMSIsInRva2VuIjoieHo2MHd5cHd2cXIzbzlwMjA3YzEiLCJzZW5kX2F0IjoxNzU2ODI1MjAwLCJlbWFpbF9pZCI6NzUwNDk2MiwiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2dpdGh1Yi5jb20vdHdpbi1icmlkZ2VzL2xlYXJuaW5nX3B5dGhvbi90cmVlL21haW4vbGVzc29uMy9leGVyY2lzZXM_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.s1K9ejgf5BL0MbOoitWnFdAhDSvyWxrK7EeNnMnf6uY)  
  
  
1. This exercise expands on the subnet calculations from lesson2, exercise1.  
  
Create a base address variable and set it to "192.168.254.". Prompt a user to enter a subnet prefix length that ranges between 25 to 30 (i.e. the netmask length of the subnets). Save this input as an integer.  
  
From the entered subnet prefix length, calculate the size of the subnet (the number of total IP addresses in the subnet). Once we know the subnet size, we can calculate the number of hosts allowed in the subnet (subtract off the network number and the broadcast address).  
  
Use a loop to print out all of the subnet network numbers. For example, if your prefix length is 26, then your program should print out the following:

​Subnets:

Number of subnets: 4

Subnet Number: 192.168.254.0

Subnet Number: 192.168.254.64

Subnet Number: 192.168.254.128

Subnet Number: 192.168.254.192

Your program should print out the following:

1. The number of hosts in the subnet.
2. The total number of subnets and the network number for each of the subnets
3. The first and last host address in just the first subnet.

2. Create a Python program that reads in the file named [show\_ip\_int\_brief.txt](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..sgYySolOCaaZEV7XB5iTuIk0d7DD5yx5pKvj-R6DGvA).  
  
Loop over the file contents (in some way) and find the address that belongs to the "10.220" network. Note, since we have not talked about regular expressions yet, you can use simple string checks for finding this address.  
  
Once you find the line containing the "10.220.x.x" address, extract both the interface name and the IP address. Save these two items to variables named "intf\_name" and "ip\_addr". Print these variables out to standard output.  
  
  
3. Write a Python program that stays in a while loop until a timeout expires.  
  
Your program should set the timeout to be five seconds. Additionally, you should use "time.sleep(1)" to sleep for one second inside your while loop. You will need to "import time" to be able to call time.sleep().  
  
You can create a start\_time variable using "start\_time=time.time()".  
  
Once you have captured the start time, then you should be able to compare the start time to the current time. This will allow you to determine whether you have exceeded the five second timeout or not.  
  
While inside of your loop, print out a message indicating the amount of time elapsed since the start time (print this out each time through the loop).  
  
Additionally, print out a message indicating that you have exited the loop and also the total elapsed time (elapsed time since the start time).  
  
  
4. Create a Python program that reads the file named [show\_ip\_int\_brief.txt](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..sgYySolOCaaZEV7XB5iTuIk0d7DD5yx5pKvj-R6DGvA). Loop over the contents of this file (in some way).  
  
From the contents of this file, find all of the interfaces that are assigned an IP address. You can assume that any IP address will start with "10.".  
  
From this, create a list of all of the assigned IP addresses. Additionally, create a list of all the interfaces containing an IP address. The order of these two lists should match. In other words, the first IP address should correspond to the first interface in the interfaces list.  
  
Print both of these lists out to standard output.  
  
  
5. Create a Python program that reads the file named [junos\_show\_arp.txt](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..6lNbt6k1Y_4k4hkIHpPrKexIlYQE08_SCakO2XJd18k). From this Junos ARP information, extract all of the IP address to MAC-address bindings. Convert the MAC address format from '06:1a:97:b0:c4:89' to '06-1a-97-b0-c4-89'  
  
Print all of these pairings out to standard output.

**CLASS OUTLINE**

1. **Video1 - Lesson3 Introduction**
2. **Video2 - Conditionals​​​**
   1. Conditionals: Syntax [0:16]**​**
      1. Expression - evaluates boolean true or false [0:25]
         1. Example ‘if’, ‘in’, ‘:’, indented block executed if True [0:40]
   2. Conditionals: Making Choices [0:55]
      1. ‘elif’ if previous was False, then evaluate block [1:05]
      2. ‘else’ catchall if none of the preceding if/elif/elif were True [1:20]
      3. Only one block will happen [1:40]
   3. Comparison operators [2:05]
      1. == Comparison equals
      2. != Not equals
      3. >, >= Greater than
      4. <, <= Less than
   4. Logical-ors and logical-ands [2:44]
      1. ‘and’ items must be True
      2. ‘or’ if at least one of the items True
   5. Nesting conditionals [3:54]
      1. Conditionals inside conditionals
         1. Could be multiple levels of depth [4:02]
   6. Idiomatic Expressions [4:27]
      1. Linters will prefer “is None”, “is False” as compared to using “==” [4:44]
      2. Linters will prefer ‘is not None’ [5:06]
   7. Remember “truish” [5:21]
      1. Python assigns false values
         1. Integers: False value is 0
         2. Strings: False value is null string
         3. Lists: False value is blank list
         4. Floats: False value is 0.0
      2. Everything else is true… [5:50]
         1. Is it the behavior you want? [6:17]
         2. Is the data type correct?
3. **Video3 - For Loops**
   1. **​**Syntax “for”keyword, loop variable, in, and iterator (list/ collection of things to be looped over) [0:25]
      1. Iterator can be a list, strings, tuples, sets, and dictionaries [2:25]
      2. Looping over strings gets a single character [2:49]
   2. Break - Immediately Exit Loop [2:57]
   3. Continue - Immediately jump back to the top of the for loop [3:46]
   4. Nesting loops [4:23]
   5. Enumerate: Obtaining both the index and the item [6:00]
   6. Looping over an empty list [7:18]
   7. Pass - Python no-op [7:43]
   8. For Loop “else” clause [8:05]
4. **Video4 - While Loops**
   1. **​**Syntax: “while” keyword and expression (True or False)  [0:10]
      1. Stays in the loop as long as expression are True [0:24]
   2. Infinite Loops [0:40]
      1. Ensure that you always exit the loop
         1. False statement [0:42]
         2. Break statement [1:17]
   3. while loop: many similarities to for-loops [3:10]
   4. while True - stay in the loop until ‘break’ happens [3:46]
   5. Nesting loops [4:16]
   6. For (think “for each) vs While (event based) [4:43]
5. **​Video5 - List Comprehensions**
   1. Why do list comprehensions exist? [0:27]**​**
   2. Syntax: Square brackets, loop over iterator, what we do with the loop variable, and loop variable [1:45]
   3. List Comprehensions: can include a conditional [2:13]
      1. Ex. loop over list, only include elements that are True [2:22]
   4. List Comprehensions: Multiple loops [2:57]
   5. List Comprehensions: Multiple Loops and Conditionals [4:20]
6. **Video6 - Generator Expressions**
   1. **​**What is a generator? Why do they even exist in Python? [0:14]
      1. Python stores items in memory [0:40]
      2. Algorithm for how to get the next number [1:50]
   2. Syntax: “yield”, we know current state and how to return the next value [2:15]
   3. Generator Expressions: A shorthand way for creating generators [3:10]
      1. Similar to list comprehensions except you use parentheses to enclose generator expressions [3:30]
      2. Can use enormous values [4:45]
   4. Generator Expressions: Multiple loops [5:00]
   5. Generator Expressions: A Gotcha (generators remember previous state - they don’t reset, initialize again in a looping construct) [5:25]