**Loops**

1. Loops are used to repeat a block of code a given number of times. Sometimes this is done while incrementing a counter of some kind or while processing data such as reading each line of a file or records from a database.
2. As with "if" statements a condition is tested and the loop runs until it is no longer true or some other condition forces the loop to break. Also like if statements the condition is terminated with a colon and the code block that runs while the condition is true is indented once.
3. Basic syntax

while <condition>:  
 code to run while condition is true  
 code to run while condition is true  
 code to run while condition is true

1. A break statement may be used to exit the loop early if some other condition proves true.
2. Continue statements cause the code execution to jump back to the beginning of the loop.
3. One mistake many beginners make when writing a loop is to make sure the condition that is being tested changes at some point. Failure to do so causes the loop to run forever. However, there are times when a seemingly unending loop can be useful. This can be accomplished using the True keyword.

while True:  
 code to run while condition is true  
 …  
 code to run while condition is true

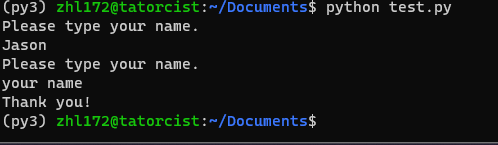
1. Example 1

x = 0

while x < 10:  
 print(x)  
 x = x + 1

1. Execute the code above. What is the result?
2. 0 1 2 3 4 5 6 7 8 9
3. Examine the following loop

name = ''  
while name != 'your name':  
 print('Please type your name.')  
 name = input()  
print('Thank you!')

1. When will the above loop end?
2. When user input ‘your name’
3. Execute it and paste a screen cap. Were you right?
4. 
5. A loop that eternally checks for the existence of a file

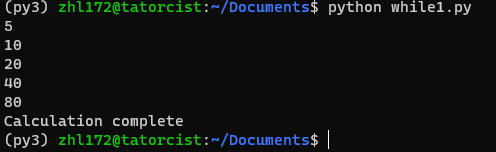
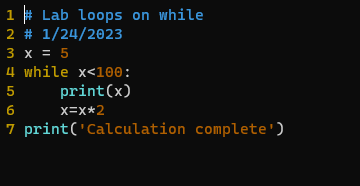
# Author: Robert Ellison  
# A simple file monitor

# load modules  
import os  
import time

# prompt for files  
myFile = input('Please enter file path: ')

i = 1

# eternal loop  
while True:  
 print('Check for file {}, attempt {}'.format(myFile, i))  
 if os.path.exists(myFile): # test if file exists and exit loop if it does  
 break  
 i = i + 1  
 time.sleep(5) # wait 5 seconds before continuing loop  
print('{} has been found!'.format(myFile))

1. Copy the code and update any indentation issues.
   1. Review the code and ask questions if you do not understand anything.
   2. Execute it and specify [/home/username@univ.pitt.edu/job.log](mailto:/home/username@univ.pitt.edu/job.log) for the path where username is YOUR pitt username.
   3. Open another terminal window or another tab and ssh into the python server.
   4. Use vim to create job.log in the root of your profile directory
      1. Enter ‘Hello World!’
      2. Save and exit
   5. Flip back to the window running the monitoring script
   6. What happened and why?
   7. The script found the job.log and stopped the loop
2. Create a while loop with the following conditions:
   1. Initialize a variable with the value of 5
   2. Run the loop until the variable has a value greater than 100
   3. Print the current value
   4. Multiply the variable by 2 and store it in itself
   5. Print "Calculation complete" and the ending value once the loop exits.
3. Insert a screenshot of your code and successful execution.
4. 
5. 
6. Modify the code from step 12 to check for removal of a file and report when it no longer exists.
   1. Post your code and successful execution here.
7. For loops can be used to iterate through a sequence, which can be a list, tuple, dictionary, set, or string. We will learn more about each of these in the coming weeks.
8. Syntax:

for <sequence>:  
 code to run

1. Example:

for i in range(1,10):  
 print('i')

1. Checking for a file using an eternal for loop:

# Author: Robert J. Ellison  
# Eternal For Loop Demonstration

import os  
import itertools  
import time

myFile = input('Enter file path to monitor: ')

# a cycle function never ends it repeats whatever value(s) given endlessly

for i in itertools.cycle('1'):  
 print('Checking for file {}'.format(myFile))  
 if os.path.exists(myFile):  
 break  
 time.sleep(5)  
print('File {} found!'.format(myFile))

1. Use a for loop to print values from 10 to 20.
   1. Post your code and successful execution here.