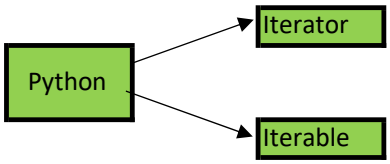
Some theory on python



In type of data i.e list,tuple,str,int for loop first tries to convert an iterable into an iterator. Data will be first available in the form of a iterable.
Only if data is iterable it can be converted to iterator.

Eg. Consider a list = [1,2,3,4]
then the list object is iterable but not in iterator form

How for loop internally works ?
In case of for loop:

If the item is iterable, the function first tries to convert iterable to iterator using **iter()** command.
It then uses **next()** command to extract each and every data.

Not all kind of data is iterable. Therefore if the object is not iterable you will not be able to convert those data to iterator.

list, tuple, Set, dict, Str objects by default are iterable so you can convert it to iterator using iter() command.

For loop uses **len()** command to stop the function at the end of object.

How does the range function work?
In case of range function:

it tries to generate a data set for us but the output is not directly displayed.
Yield command in place of print will store data in the form of generator object and will not give the output directly. same as in case of range function.
Therefore to convert any kind of function to a generator function use yield command.

Why is yield command used ?
When we define a function with return command and input very big value, the CPU in your system will take quite a long time processing to create those dataset and give the output and the laptop may even get hanged.

Whereas using yield command will give the output of each loop from the time the code runs and works in background to calculate and compute all the balance data without much affecting the processing power of CPU .

File operation

ls	command will open the default directory in drive C.
f = open("aditya1.txt")	will give an error if there is no file with that name in the directory
f = open("aditya1.txt",'w')	will create a new file in the directory with that name where 'w' means write
	r' means read, 'w+' means read and write, 'r+' means read and write.
f.write("this is my first file operation")	This will open the file and enter the data into the it to save the same.
f.close()	This command will close the file. With ls command if you open the
	file the input data size will reflect automatically

This operation is useful to save the data which you are temporirily holding in your juputer notebook

l = [1,2,3,4,5]	To enter list objects into a text file via python
f = open("text1.txt",'w')	
f.write(str(l))	In order to write list object you need to first convert it to a string variable
f.close()	You need to close the file for the data to reflect in the wordpad

f2 = open("text.adit",'w')	No matter in what format you save the file, it will always save in string format and will open in wordpad.

%%writefile text1.txt **this is my first python program to write into a file in a different way**	This will ***overwrite*** the text in the already existing file

File read operation	
f = open("text1.txt",'r')	It will display what ever data that is there in the file.
f.read()	Performing read operation the second time will not yield any dataas the first read operation has exhausted all the data and the pointer is at the end of very last object and there is no data after the last object
f.read()	
f.seek(0)	Will mover the pointer at the very same index in a string
f.read()	Only then read operation will yield data from start of the string. You can also provide index as per requirement to read the data.
f.tell(0)	It will show the pointer location in file.
f.read(15)	It will show the data upto the 15th index
f.readline()	will show one single line from the pointer
f.readlines()	will show all the lines present in the file.

Appending data in file operation	
f2 = open("text1.txt",'a')	a' stand for appending the data or else everytime
f2.write("ada dsaad")	it will overwrite the data.
f.name	will show the name of the file.
f.closed	to check whether the file is closed or not.
f.mode	will show a/r/r+/w/w+ mode

Mapping	**=map(function,iterable)**	It will implement the function operation to each and every element of the iterable.

Additional modes

It's fairly ineffecient to open the file in a or w and then reopening it in r to read any lines. Luckily we can access the file in the following modes:

r+ : Reading and writing. Cannot truncate the file.
w+ : Writing and reading. Truncates the file.
a+ : Appending and Reading. Creates a new file, if none exists.
tell() - returns the current position in bytes
seek(offset,from) - changes the position by 'offset' bytes with respect to 'from'. From can take the value of 0,1,2

corresponding to beginning, relative to current position and end