Lecture 8 - Loops / For Loops / Range

To deal with data that is not individual data we have 2 different tools. The first is functions that call themselves - this is recursion.

The second is Loops. Loops are the more common.

```
for i in range(3):
        print ( "i = {}".format(i) )
```

So... This uses the “range” operator. range encodes a set of values that can be used in a loop.

```
print ( range(5) )
```

When it is specified like this it is the same as `range(0, 5)` .

A range is a start value that increments up to but not including the end value. So `range(0, 5)` will have values 0, 1, 2, 3, 4.

This is useful whenever we have a loop to control how many times the loop will go around.

Let’s walk through a loop:

```
ll = [ "dog", "cat", "goldfish", "parakeet" ]
for i in range ( len(ll) ):
        print ( "i = {} list[{}] = {}".format ( i, i, ll[i] ) )
```

So we can use this to search a list:

```
ll = [ "dog", "cat", "goldfish", "parakeet" ]
for i in range ( len(ll) ):
    if ll[i] == "cat":
        print ( "found 'cat' in list! list[{}] = {}".format ( i, ll[i] ) )
```

We can also use a for loop to go through the values in a list.

```
ll = [ "dog", "cat", "goldfish", "parakeet" ]
for i in range ( len(ll) ):
    if ll[i] == "cat":
        print ( "found 'cat' in list! list[{}] = {}".format ( i, ll[i] ) )
```

So... Let's search a bigger list.

```
 1:
 2: def readNameList(fn):
 3:     f = open(fn,"r")
 4:     if f == None:
 5:         print ( f"Invalid file {fn} - failed to open" )
 6:         return None
 7:     dt = f.readlines()
 8:     f.close()
 9:     for i in range (len(dt)):
10:         s = dt[i].rstrip()
11:         dt[i] = s
12:     return dt
13:
14: phone_list = readNameList("50000phone.csv")
15:
16: print ( "Enter a Name to Lookup" )
17: lookFor = input()
18: found = False
19: for v in phone_list:
20:     if lookFor in v:
21:         found = True
22:         print ( "Found {}".format(v) )
23: if not found:
24:     print ( "no names found" )
25:
```

So... What is a .csv file - you can save them from excl or from programs like numbers on a mac. If you double-click on a .csv it will take you into a spreadsheet.

Let's take a look in VI at the file.

CSV stands for comma separated value. Usually values with commas in them will need to be quoted.

It is very common for programs that need to produce structure data to output that data in CSV format.

For example the US Census has data in .csv format.

```
ID,STATE_CODE,STATE_NAME,CITY,COUNTY,LATITUDE,LONGITUDE
1,AK,Alaska,Adak,"Aleutians West",55.999722,-161.207778
2,AK,Alaska,Akiachak,Bethel,60.891854,-161.39233
3,AK,Alaska,Akiak,Bethel,60.890632,-161.199325
4,AK,Alaska,Akutan,"Aleutians East",54.143012,-165.785368
...
```

Notice in this data the first line is a “header” line.

Let’s modify our read-file to read in CSV data.

```
 1:
 2:
 3: import csv
 4:
 5: # readNameListCSV opens 'fn' and reads the CSV file into a dictory.
 6: def readNameListCSV(fn):
 7:
 8:     f = open(fn,"r")
 9:     if f == None:
10:         print ( f"Invalid file {fn} - failed to open" )
11:         return None
12:     csvR = csv.reader(f)
13:     dt = {}
14:     for row in csvR:
15:         dt[row[0]] = row[1]
16:     f.close()
17:     return dt
18:
19:
20: # Automated Test
21: if __name__ == "__main__":
22:     n_err = 0
23:
24:     got = readNameListCSV("2names.txt")
25:     # print ( "got= {}".format(got))
26:     expect = {
27:         "Gunter, Dolly R":"(072) 123-4760",
28:         "Polk, Hattie S":"(563) 404-0792"
29:         }
30:     if got["Polk, Hattie S"] != expect["Polk, Hattie S"]:
```

```
31:         n_err = n_err + 1
32:         print ( "Error: Test 1: file read error expected {} got {}".
33:            format (  expect["Polk, Hattie S"], got["Polk, Hattie S"] ) )
34:
35:     if n_err == 0 :
36:         print ( "PASS" )
37:     else:
38:         print ( "FAILED" )
39:
```

Let's read in a sample of the phone numbers as a dictionary.

```
 1:
 2: from readNameListCSV import readNameListCSV
 3:
 4: x = readNameListCSV("5-lines.phone.csv" )
 5:
 6: for key in x:
 7:     print ( "name={}         phone={}".format(key,x[key]))
 8:
```

And now use a dictionary to search it.

```
 1: #!/Users/philip/opt/anaconda3/bin/python
 2:
 3: from readNameListCSV import readNameListCSV
 4:
 5: phone_list = readNameListCSV("50000phone.csv")
 6:
 7: print ( "Enter a Name to Lookup\n=> ", end="" )
 8: lookFor = input()
 9:
10: if lookFor in phone_list:
11:     print ( "Found {}".format(phone_list[lookFor]) )
12: else:
13:     print ( "{} not found".format(lookFor) )
14:
```

Copyright