

Lecture 6 - Lists / Dictionaries

First let's read in a file

```
def readNameList(fn):

    f = open(fn,"r")
    if f == None:
        print ( f"Invalid file {fn} - failed to open" )
        return None
    dt = f.readlines()
    f.close()
    for i in range (len(dt)):
        s = dt[i].rstrip()
        dt[i] = s

    return dt


# Automated Test
if __name__ == "__main__":
    n_err = 0

    got = readNameList("2names.txt")
    expect = [
        "\"Gunter, Dolly R\",(072) 123-4760",
        "\"Polk, Hattie S\",(563) 404-0792"
    ]
    if got[0] != expect[0]:
        n_err = n_err + 1
        print ( "Error: Test 1: file read error expected {} got {}".format ( expect[0]
    if got[1] != expect[1]:
        n_err = n_err + 1
        print ( "Error: Test 2: file read error expected {} got {}".format ( expect[1]

    if n_err == 0 :
        print ( "PASS" )
    else:
        print ( "FAILED" )
```

Let's remove the punctuation

```
import string

def removePunctuation(txt):
    for c in string.punctuation:
        txt = txt.replace(c,"")
    return txt

# Automated Test
if __name__ == "__main__":
    n_err = 0

    got = removePunctuation("this, and: that")
    expect = "this and that"
    if got != expect:
        n_err = n_err + 1
        print ( "Error: Test 1: file read error expected {} got {}".format ( expect, g

    if n_err == 0 :
        print ( "PASS" )
    else:
        print ( "FAILED" )
```

List Example

```
def readNameList(fn):
    f = open(fn,"r")
    if f == None:
        print ( f"Invalid file {fn} - failed to open" )
        return None
    dt = f.readlines()
    f.close()
    for i in range (len(dt)):
        s = dt[i].rstrip()
        dt[i] = s
    return dt

phone_list = readNameList("50000phone.csv")

print ( "Enter a Name to Lookup" )
lookFor = input()
found = False
```

```
for i in range(len(phone_list)):
    if lookFor in phone_list[i]:
        found = True
        print ( "Found {}".format(phone_list[i]) )
if not found:
    print ( "no names found" )
```

Dictionary

A dictionary is an association between a “key” value and a set of data that is efficient for looking things up by the key.

let’s do an “age” one...

```
>>> dd = {}
>>> dd["bob"] = 22
>>> dd["jane"] = 31
>>> dd["marry"] = 18
>>>
>>> dd["bob"]
>>> dd["jane"]
```

Now changing our phone search to use a dictionary:

```
#!/Users/philip/opt/anaconda3/bin/python

from readNameListCSV import readNameListCSV

phone_list = readNameListCSV("50000phone.csv")

print ( "Enter a Name to Lookup\n=> ", end="" )
lookFor = input()

if lookFor in phone_list:
    print ( "Found {}".format(phone_list[lookFor]) )
else:
    print ( "{} not found".format(lookFor) )
```

Requirements

Prompt for a file name. Then read in a file of text.

Split the file up into words. Remove any punctuation and convert each word to lower case. Count how many times each word occurs. Print out a sorted list of the words in the file with the number of occurrences of each word. Sort from the most frequent word to the least.

Requirements Broken Down

1. Prompt for a file name.
2. Read in a file of text.
3. Split the file up into words.
4. Remove any punctuation and convert each word to lower case.
5. Count how many times each word occurs.
6. Print out a sorted list
 - Print out a sorted list of the words in the file with the number of occurrences of each word.
 - Sort from the most frequent word to the least.

Let's count some words

```
#!/Users/philip/opt/anaconda3/bin/python

from readNameList import readNameList
from removePunctuation import removePunctuation

def main():

    print ( "Enter File Name\n=> ", end="" )
    fn = input()
    # read in file to a list of lines
    data = readNameList(fn)

    # create an empty dictionary, the key will be the word
    #the value will be the count.
    freq = {}

    for line in data:

        # Split line up into a set of words
        words = line.split()

        for word in words:

            word = word.lower()
            word = removePunctuation(word)

            if word in freq:
```

```
        freq[word] += 1
    else:
        freq[word] = 1

# Prints that would just print out in word sorted order.
#for key in sorted(freq.keys()):
#    cnt = freq[key]
#    print ( f"{key} = {cnt}" )

print ( "{name:5s} : {word}".format(name="Count",word="Word") )
print ( "{name:5s} : {word}".format(name="-----",word="-----") )
x = sorted(freq.items(), key=lambda x: x[1], reverse=True)
for p in x:
    print ( f"{p[1]:5d} : {p[0]}" )


if __name__ == "__main__":
    main()
```

Copyright

Copyright © University of Wyoming, 2021.