Lecture 9 - Lists and Dictionaries

List Example

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

Dictionary

A dictionary is an association between a "key" value and a set of data that is efficient for looking thins 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:

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

Letter Paris

Paris of letters are the "phonetic" parts in English. Single letters do not make most sounds in English. Letter Paris do.

Let's build a program that counts the occurrences of pairs.

```
 1: import readNameList
 2:
 3: def main():
 4:
 5:     print ( "Enter File Name\n=> ", end="" )
 6:     fn = input()
 7:     # read in file to a list of lines
 8:     data = readNameList.readFileIntoList(fn)
 9:
10:     letter_pair = {}
11:
12:     for line in data:
13:
14:         # Split line up into a set of words
15:         words = line.split()
16:
17:         for w in words:
18:             for i in range(len(w)-1):
19:                 pair = w[i:i+2]
20:                 pair = pair.lower()
21:                 if pair in letter_pair:
22:                     letter_pair[pair] = letter_pair[pair] + 1
23:                 else:
24:                     letter_pair[pair] = 1
25:
26:
27:     # print ( "Count of Paris: {}".format(letter_pair) )
28:     print ( "Count of Paris:" )
29:     for p in letter_pair:
30:         print ( "{}: {}".format( p, letter_pair[p] ) )
31:
32: main()
```

Copyright

Copyright © University of Wyoming, 2021.