ECS 10

2/27

Announcements

- MIDTERM 2 -- Friday in class
- Open book, open notes. Bring sample programs from class, your programs, the book, etc.
- Bring a Scantron 2000 form
- On SmartSite: last year's midterm.

Resources

- Practice exam
  - Will be more on functions on the test
  - Question 12 is e, not d.
- Homework problems – solution to Prog 5 up right after class.
- Class slides, example programs.
- Book:
  - Up through Chapter 7
  - Stuff not covered in lecture can be ignored
- Python tutor

Topics

- Lists, indexing, append method
- Dictionaries
- Files: reading and writing, open and close
- Try-except
- for loops on files, lists, dictionaries, range
- String methods (split, replace, strip)
- Functions, arguments/parameters, local variables, return values
- Break, continue, return

Programming problem

- Will involve reading a file
- Will involve building a dictionary
- Will be required to be a collection of functions, with main() the only command outside

Programming example

Read in a file containing one country per line, and, for each country, a list of major exports. The program should then let the user enter products, and it should report all the countries producing that product.

Solve this problem using a dictionary, in which the keys are the exports and the values are lists of countries.

Programming problem

Sample input file "exports.csv":

Albania, textiles, metals, oil

Algeria, oil, natural gas

Argentina, edible oils, oil, grain, motor vehicles

Sample output:

if user enters oil: Albania, Algeria, Argentina

How to approach this problem?

- The dictionary is the main idea; and the output is the clue to the structure of the dictionary.
- How do items get taken out of dictionary?
  - For now, assume they get in by magic.
  - What are queries from user?
  - What is produced in response to the queries?
  - Which are keys and which are values?
  - Data type of keys? values?
  - Make it clear what you want keys and values to be.

Building the dictionary

- How do items get into the dictionary?
  - Read file.
  - Break down lines from file.
  - Put data into dictionary entries.
  - Check to see if entry exists; if so, add to it; if not, start it (example from Friday/Monday program).

Adding an item to a list in a dict

```
# country is a country and prod is an export
    if prod in exportD:
            countries = exportD[prod]
            exportD[prod] = countries.append(country)
    else:
            exportD[prod] = [country]
```

Two-loop program

- Do this on scrap paper, then copy.
- Start at the bottom, with main()
- Write main() function, calling two functions, one to build dictionary, one to take stuff out of dictionary.
  - What is input to each function?
  - What is output of each function?
  - Possibly no inputs or no outputs.
- Then fill in functions

Functions – what does it print?

```
def dropH(s):
    s = s.replace("h", "")
    return s
def main():
  inStr = "hurricanes hardly happen"
  inStr = dropH(inStr)
  print(inStr)
main()
```

Functions – what does it print?

```
def dropH(s):
    s = s.replace("h", "")

def main():
    inStr = "hurricanes hardly happen"
    dropH(inStr)
    print(inStr)
main()
```

Functions – what does it print?

```
def dropH(s):
    if "can" in s:
        return s
    s = s.replace("h", "")
    return s

def main():
    inStr = "hurricanes hardly happen"
    inStr = dropH(inStr)
    print(inStr)
main()
```

List indexing – fill in the blank

```
from random import randrange
suits = ['Hearts', 'Clubs', 'Spades', 'Diamonds']
faces = ['Jack', 'Queen', 'King']
i = randrange(4) # choose suit randomly
num = randrange(13)  # choose number
if num < 10:
    _____
else:
    _____
```

Dictionary – fill in the blank

```
giftDict = {}
giftDict["Lion"] = "Courage"
giftDict["Tin Man"] = "a Heart"
giftDict["Scarecrow"] = "Brains"
character = input("Character name: ")
if _____
    _____
    print ("The gift of the",character,"is",gift+".")
```

Dictionary usage

- Putting a key,value pair into the dictionary:

```
giftDict["Lion"] = "Courage"
```

- Dictionary location on the left is a variable; it is a `labeled box' we put data into.
- Data value on the right can be anything.
- Getting data out of the dictionary:

```
gift = giftDict[character]
```

- **Data from dictionary goes** into the variable on the left.

Changing a dictionary item

```
x = x+1
```

- Takes value in x, adds one to it, puts result back into x, over-wrting the old value.
- A dictionary or a list is a big collection of variables, just with a different way of labeling them.

```
myDict["a"] = 4
myDict["a"] = myDict["a"]+1
```

Checking for a Key

```
if traveller in giftDict:
```

- **If you try to access a key that is not in the dictionary, the program crashes!**
- **If there is any doubt that your key will be there, check first.**
- **Use the in operator. Produces a Boolean, True if key on left is in dictionary on right.**

For Loops

```
for line in myFile:
```

- If myFile is a file, the variable line will contain each each line in turn.

```
for word in wordList:
```

- If wordList is a list of strings, the variable word will be each string in the list in turn.

```
for i in range(0,len(a)):
```

- Whether a is a list or a string, i will be the integers from 0 through len(a)-1, in turn.

```
for k in carDict:
```

- If carDict is a dictionary, k will be each key in turn.

Append, for loop with range

```
digits = [ ]
for i in range(0,10):
    digits.append(str(i))
print digits
```

- **digits** begins as the empty list
- i takes on the values 0,…,9
- Function **str(i)** converts integer **i** to a string.

String Methods

- **string = string.strip()**
  - Removes leading and trailing whitespace
  - Mainly used to remove newline characters from a file
- **words = string.split()**
  - **words = string.split(“,”)**
  - **words = string.split(“\t”)**
  - Returns **list** of substrings
- **string = string.replace(“%”, “”)**
  - Replaces all copies of one substring with another

String Processing Example

```
string = "Once bitten, twice shy.\n"
string = string.strip()
words = string.split()
for i in range(0,len(words)):
    w = words[i]
    char = w[-1]
    if not char.isalpha():
        words[i] = w.replace(char,"")
print(words)
```

Files

```
inFile = open(“myFile.txt”,”r”)
for dataStr in inFile:
    print ‘Next line is: ’, dataStr
inFIle.close()
```

- Files are read from beginning to end; no going backwards!
- dataStr takes on the value of each line of the file in turn.

File reading Loops

```
while True:
    line = inFile.readline()
    if line == "":
        break
```

- A little more flexible

```
for line in inFile:
```

- Usually sufficient

Try-except construction

```
while True:
    filename = input("enter file name: ")
    try:
        inFile = open(filename,"r")
        break
    except:
        print("Cannot find file")
```

Try-except for data conversion

```
while True:
    s = input("Enter a number: ")
    try:
        float(s)        # Try to do the conversion
        break
    except:
        print("Bad input")  # Conversion failed!
```

Writing Files

```
myFile = open('toExcel.tsv', 'w')
for i in range(10):
    floatNum = float(i)
    myFile.write('item'+str(i)+'\t%.2f\n'%floatNum)
myFile.close()
```

- Open file to write with 'w'
- write() method has one **string** argument.
- **Use the str() function to convert numbers to strings.**