CS 205

Lecture Handout #3
January 26, 2016

Data Set: Majors at UIUC

The data set for this week will focus on the population of different majors at UIUC. The data set contains 11 columns, with three being most relevant to our analysis:

- **Fall**, the year of the Fall semester the data was collected
- **Major Name**, the name of the major
- **Total**, the total number of majors at that time

Python Pattern: Reading a CSV File

Last week, we developed the CSV reading pattern we will reuse:

```
 1: # Required import:
 2: import csv
 3:
 4: # Read the rows of the file:
 5: f = open("fileName.csv")
 6: reader = csv.DictReader(f)
 7: data = [row for row in reader]
 8:
 9: # Loop through each row of data
10: for row in data:
11:   print( row["Major Name"] )
```

...from the CS 205 Guidebook: "Reading a CSV File"

Q: How many students were at UIUC each year?

- Option #1:

- Option #2:

- Option #3:

Using dictionaries to store categorized data

Dictionaries are convenient structures to store all the information about a specific category in one place.

Variable	Key *Must be string*	Value Can be anything (array, dictionary, etc)
`yearsDict`	"2015"	...any data about 2015...
	"2014"	...any data about 2014...
	"2013"	...any data about 2013...
	"2012"	...any data about 2012...
	...	...

Python Pattern: Dictionaries for Categorized Data

Any time you want to categorize data into a dictionary, there is another Python pattern to follow.

The following instance of the pattern sums the values into our dictionary:

```
 1: # Create an empty dictionary
 2: myDictionary = {}
 3:
 4: # Loop through each row of data
 5: for row in data:
 6:
 7:   # Pull out the key, ensure it is a string
 8:   key = str( data["Field for key"] )
 9:
10:   # Check if the key doesn't exist in the dict
11:   if key not in myDictionary:
12:     # If not, initialize it
13:     myDictionary[ key ] = 0  # or whatever
14:
15:   # Add the row to our dictionary
16:   myDictionary[ key ] += row["Total"]
```

...from the CS 205 Guidebook: "Reading a CSV File"

What was UIUC's student population in 2005?

...in 2015?

...what was the percent growth?

Puzzle #1: Find the growth rate of all majors.
Examples:

- "Computer Science" grew from **1064** majors in 2005 to **1640** majors in 2015 *(a growth rate of **+54.14%**)*.
- "French" shrank from **102** majors in 2005 to only **42** majors in 2015 *(a growth rate of **-58.82%**)*.

Observations about solving the problem:

-

-

-

Puzzle #1(a): A function to populate growth rates
In order to find growth rates, it is great to decompose this logic into a function whose only purpose is to find growth rates.

```
 1: def populateGrowthRates(d):
 2:   # d is a dictionary in the format
 3:   # d = { "2015": { "French": 32,
 4:   #                 "Dance": 24, ... },
 5:   #        "2014": { "French": 37,
 6:   #                 "Dance": 26, ... }
 7:   #        ... }
 8:
 9:   # Loop through 2015 majors
10:
11:   for _________ in ____________:
12:      # Check if major exists in 2005:
13:
14:     if ____________________________________:
15:         # If so, compute growth rate:
16:
17:
18:
19:      else:
20:         # If not, give a default value
21:
22:
23:
```

Python Pattern: Sorting a Dictionary
Dictionaries (by default) in Python are *un-ordered*, so we need to use an `OrderedDict` to store a sorted (ordered) dictionary. *This pattern gets a little complex, but always works.*

```
1: # Import the OrderedDict
2: from collections import OrderedDict
3: ...
4:
5: # Assume a myDictionary with data
6: # myDictionary = { ... }
7:
8: sortedDictionary = OrderedDict(
     sorted( myDictionary.items(),
             key = lambda: d: d[1]["sort field"]
           )
   )
```

By default, the keys are sorted in ascending order. You can reverse this by placing this argument into the `sorted` function (Line 8):

```
             reverse = True
```

What department had the largest growth rate?

What was the growth rate of your department?

Before our next class...

1. Continue to develop Python skills by completing **the first lessons (14 parts)** in the following on codecademy.com:
   - **Units 5**: "Python Lists and Dictionaries"