CS 205

**Majors at UIUC: Part 1**
Lecture #3 | August 30, 2016

## Data Set: Demographics by Department at UIUC

The dataset for this week involves demographic data from every department at UIUC from 2005-2015. Exploring the dataset, what are important columns in the data?

| Column Name | Description/ Format |
|---|---|
| Fall | |
| | |
| | |
| | |
| | |
| | |

## Python Pattern: Reading a CSV File

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

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

## Q: How many students were at UIUC each year?

- Option #1:

- Option #2:

## Python Dictionaries

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

`countByYear` ➔

| "2015" | "2014" | "2013" | "2012" | "2011" | "2010" | … |
|---|---|---|---|---|---|---|
| 44,087 | 43,603 | 43,398 | 42,883 | 42,606 | 41,949 | … |

We can access a specific year by indexing into our dictionary:

```
1: # Reading a value from a dictionary:
2: print( countByYear["2015"] )  # prints 44087
3:
4: # Setting/adding a value in a dictionary:
5: countByYear["2015"] = 100
6: countByYear["2015"] += 200
```

## Python Pattern: Dictionaries for Categorized Data

Create a Python program that finds the student count for every year:

```
# Create an empty dictionary:


for row in data:
  # Pull out useful fields:









  # Initialize our dictionary entry if it does
  # not already exist in our dictionary:









  # Update our dictionary with the current row:
```

## Multi-Level Dictionaries

A dictionary can contain any type of data within it, including other dictionaries! Counting the number of students in each major by year requires a multi-level dictionary:

**majors ➔**

| | | | | | | … |
|---|---|---|---|---|---|---|
| "2015" | "2014" | … | "2015" | "2014" | … | … |
| | | | | | | |

## Python: Programming Multi-Level Dictionaries

Create a Python program that finds the student count for every year:

```
# Create an empty dictionary:


for row in data:
```

## Python Lists

Besides dictionaries, a list is the other major data structure that will be extremely useful for data analysis and visualization. A list is an **ordered collection** of data:

**majorsList ➔**

| | |
|---|---|
| [0] | { "major": "Undeclared", "students": 2790 } |
| [1] | { "major": "Computer Science", "students": 1640 } |
| [2] | { "major": "Psychology", "students": 1480 } |
| [3] | { "major": "Accountancy", "students": 1422 } |
| ... | … |

## Creating Lists form Dictionaries

Similar to dictionaries, the general pattern is to start with an empty list and add data to the list from your dictionary:

```
# Create an empty list:
majorsList = []

# Navigate through your majors
for majorName in majors:

  # Add an entry for the major to the new list:




# Sort the list:
majorsList = sorted(majorsList,
                    key=lambda k: k["students"],
                    reverse=True)

# Print the sorted list:
for major in majorsList:
  print( major )
```

**Puzzle #1:** What major grew the most between 2010 and 2015...
...by percentage? ...by number of students?

**Puzzle #2:** List every major by percentage growth, sorting the majors by the largest/smallest percentage growth.