### CompSci 101
Introduction to Computer Science

<table>
<thead>
<tr>
<th></th>
<th>'Susan'</th>
<th>['Smith', 'Brandt', 'Rodger', 'Crackers']</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>'Jackie'</td>
<td>['Long', 'Johnson']</td>
</tr>
<tr>
<td>2</td>
<td>'Mary'</td>
<td>['White', 'Rodger', 'Velios']</td>
</tr>
<tr>
<td>3</td>
<td>'Eric'</td>
<td>['Long', 'Lund']</td>
</tr>
<tr>
<td>4</td>
<td>'Jack'</td>
<td>['Frost']</td>
</tr>
</tbody>
</table>

Oct 19, 2017
Prof. Rodger
Announcements

• Reading and RQ due next time
• APT 4 extend one day to Oct 20
  – But note no consulting hours on Friday!
• Assignment 5 due Oct 26

• Today:
  – Review Sets
  – Tuples/generators
  – Enumerate
  – Processing data – how to organize it?
Latanya Sweeney

Former Chief Technologist at FTC. I am a computer scientist with a long history of weaving technology and policy together to remove stakeholder barriers to technology adoption. My focus is on "computational policy" and I term myself a "computer (cross) policy" scientist. I have enjoyed success at creating technology that weaves with policy to resolve real-world technology-privacy clashes.

http://latanyasweeney.org/

Identify 87% of US population using (dob,zip,gender). Director of Harvard Data Privacy Lab, instrumental in HIPAA because of de-identification work
Aboutmyinfo.org

• Entered my data

How Unique are You?

Enter your ZIP code, date of birth, and gender to see how unique you are (and therefore how easy it is to identify you from these values).

Date of Birth: [Month] [Day] [Year]
Gender: [Male] [Female]
5-digit ZIP: [Enter ZIP code]

Submit
aboutmyinfo.org

How Unique are You?

Enter your ZIP code, date of birth, and gender to see how unique you are (and therefore how easy it is to identify you from these values).

- Entered my data
- Easily identifiable by birth date (about 1)
- Lots with my birth year (about 273)
- Lots of people in my age range (of four years) – (1,365)
Set Operations from pictures
bit.ly/101-101917-1

Question: Which operation does the red represent?

A)  
B)  
C)  
D)  
E)
Tuples

• Like a list, but cannot change them
  – Define them with “,”
    (5, 7, 8) or 5, 7, 8
    without ()’s has some limitations!

• Use most list operations on them
  – they are a type of list
  – But immutable

• Examples
Example

\[ x = (4, 6, 8) \]
\[ y = 9, 5, 6 \]
\[ \text{print } x \]
\[ \text{print } y \]
\[ \text{print } x[1] \]
\[ \text{print } y[1] \]
\[ y[0] = 2 \]
\[ z = ([5,6], [7,8]) \]
\[ \text{print } z \]

\[ z[0][1] = 12 \]
\[ \text{print } z \]
\[ z[0].append(4) \]
\[ \text{print } z \]
\[ z[0].remove(5) \]
\[ z[0].remove(12) \]
\[ z[0].remove(4) \]
\[ \text{print } z \]
\[ v,w = 8, 3 \]
Problem: Longest Name

Given a list of names (one word only) and a letter (assume names start with capital letter, and letter is capital)

names = ['Helen', 'Bob', 'Bart', 'Hugh']

Find the longest name that starts with that letter
Code for longest name

def longestName(alist, letter):
    longest = ''
    for name in alist:
        if letter == name[0] and
            len(name) > len(longest):
            longest = name
    return longest

How do you modify to find the location (position) of the longest name?
Problem: Find the position in a list of the longest name that starts with that letter
Enumerate

- An iterator, generates a sequence
- Generates tuples of (index, item)
- Used with for loop to get both index and item
- for (index, item) in enumerate(somelist):
  - You get both at the same time!
Solve previous problem with enumerate

• Show enumerate examples

for (index, item) in enumerate(w):
    for g in enumerate(w):
        print enumerate(w)
Problem: Popular Name

- Given a list of names, determine the most popular first name and print that name with all of its last names.
- Input: Names are always two words, names are in a file. If multiple names are on the same line they are separated by a “:”
- Output: Most popular first name, followed by a “:”, followed by corresponding last names separated by a blank
Example Input File with 5 lines

Susan Smith:Jackie Long:Mary White
Susan Brandt
Jackie Johnson:Susan Rodger:Mary Rodger
Eric Long:Susan Crackers:Mary Velios
Jack Frost:Eric Lund

Corresponding Output

Susan: Smith Brandt Rodger Crackers
What do you need to solve this problem?
bit.ly/101-101917-3
How might one organize the data to solve this problem?

How many different ways to solve this problem?
One way to solve

• Create a list of unique first names
• Create a list of lists of last names that are associated with each first name
Example – two lists

<table>
<thead>
<tr>
<th>Unique First names</th>
<th>Corresponding Last names</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[‘Smith’, ‘Brandt’, ‘Rodger’, ‘Crackers’]</td>
</tr>
<tr>
<td>1</td>
<td>[‘Long’, ‘Johnson’]</td>
</tr>
<tr>
<td>2</td>
<td>[‘White’, ‘Rodger’, ‘Velios’]</td>
</tr>
<tr>
<td>3</td>
<td>[‘Long’, ‘Lund’]</td>
</tr>
<tr>
<td>4</td>
<td>[‘Frost’]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>’Susan’</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>‘Jackie’</td>
</tr>
<tr>
<td>2</td>
<td>‘Mary’</td>
</tr>
<tr>
<td>3</td>
<td>‘Eric’</td>
</tr>
<tr>
<td>4</td>
<td>‘Jack’</td>
</tr>
</tbody>
</table>

0: 19compsci 101, fall 2017
### Example – two lists

#### Unique First names

<table>
<thead>
<tr>
<th>0</th>
<th>’Susan’</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>‘Jackie’</td>
</tr>
<tr>
<td>2</td>
<td>‘Mary’</td>
</tr>
<tr>
<td>3</td>
<td>‘Eric’</td>
</tr>
<tr>
<td>4</td>
<td>‘Jack’</td>
</tr>
</tbody>
</table>

#### Corresponding Last names

<table>
<thead>
<tr>
<th>0</th>
<th>[‘Smith’, ‘Brandt’, ‘Rodger’, ‘Crackers’]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[‘Long’, ‘Johnson’]</td>
</tr>
<tr>
<td>2</td>
<td>[‘White’, ‘Rodger’, ‘Velios’]</td>
</tr>
<tr>
<td>3</td>
<td>[‘Long’, ‘Lund’]</td>
</tr>
<tr>
<td>4</td>
<td>[‘Frost’]</td>
</tr>
</tbody>
</table>

**Jackie** in position 1

**Jackie’s last names in position 1**
Now can we solve the problem?

• Compute those two lists that are associated with each other
  – List of unique first names
  – List of corresponding last names
• Compute the max list of last names
• Now easy to print the answer.
• See popular.py
Look at the code for popular.py
www.bit.ly/101-101917-4

• Which datafile is read in?
• What format is namelist in?
• Write the code for uniqueFirstNames
Write the code:

• allLastNames

• correspondingLastNames

• printFirstWithLasts
maxnum = max([len(item) for item in lastNames])
print maxnum

lastIndex = [index for (index, v) in enumerate(lastNames) if len(v) == maxnum]
print "first name with most last names is:"

Another way – list of lists

First word in each list is a first name
The rest are last names.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>['Susan', 'Smith', 'Brandt', 'Rodger', 'Crackers']</td>
</tr>
<tr>
<td>1</td>
<td>['Jackie', 'Long', 'Johnson']</td>
</tr>
<tr>
<td>2</td>
<td>['Mary', 'White', 'Rodger', 'Velios']</td>
</tr>
<tr>
<td>3</td>
<td>['Eric', 'Long', 'Lund']</td>
</tr>
<tr>
<td>4</td>
<td>['Jack', 'Frost']</td>
</tr>
</tbody>
</table>
Expanding the Problem

• Suppose we want to read from multiple data files
  
names1.txt, names2.txt, names3.txt

See processFiles in popular.py