CompSci 101
Introduction to Computer Science

<table>
<thead>
<tr>
<th></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>

<table>
<thead>
<tr>
<th></th>
<th>[ 'Smith', 'Brandt', 'Rodger', 'Crackers']</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></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>

March 17, 2015
Prof. Rodger
Announcements

- For next time, Reading and RQ14
- APT 6 due today, APT 7 out
- Assignment 6 out soon!
Alice programming language
alice.org, Alice version 2.4
Nested Loop
Fair Ride – Octopus
Wac-A-Mole

World's details

```
properties  methods  function

Obj moles = mole, mole2, mole3, mole4, mole5, mole6, mole7, mole8, mole9, mole10, mole11, mole12
```
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']

1) Find the longest name that starts with that letter

2) Find the position of the longest name that starts with that letter

See longestName.py, DO NOT use enumerate
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!

• Redo find position of longest name with iterator
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
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  'Susan'</td>
<td>0  ['Smith', 'Brandt', 'Rodger', 'Crackers']</td>
</tr>
<tr>
<td>1  'Jackie'</td>
<td>1  ['Long', 'Johnson']</td>
</tr>
<tr>
<td>2  'Mary'</td>
<td>2  ['White', 'Rodger', 'Velios']</td>
</tr>
<tr>
<td>3  'Eric'</td>
<td>3  ['Long', 'Lund']</td>
</tr>
<tr>
<td>4  'Jack'</td>
<td>4  ['Frost']</td>
</tr>
</tbody>
</table>
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

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

- uniqueLastName

- correspondingLastName

- 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:"
Expanding the Problem

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

See processFiles in popular.py