

Plan for the week in Compsci 6/101

- **Guidance for APTs**
 - How to leverage the power of Python sorting
 - How to 'customize' sorting
 - Toward understanding recursion, more later
- **Model View Controller and Global Variables**
 - Making the Jotto program work
 - Lightening assignment
- **Working to understand the pickle module**

Compsci 06/101, Spring 2011

14.1

APTs Sorted and Sortby Frequencies

- **What are the organizing principles in SortedFreqs?**
 - Alphabetized list of unique words?
 - Count of number of times each occurs?
 - Is efficiency an issue? If so what recourse?
- **What are organizing principles in SortByFreqs?**
 - How do we sort by frequency?
 - How do we break ties?
 - `[(t[1],t[0]) for t in dict.items()]`

Compsci 06/101, Spring 2011

14.2

Sorting from an API/Client perspective

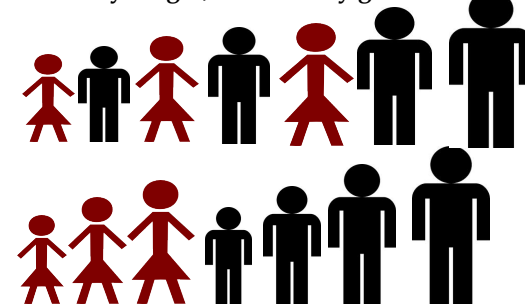
- **API is Application Programming Interface, what is this for `sorted(..)` and `.sort()` in Python?**
 - Sorting algorithm is efficient, part of API?
 - Sorting algorithm is stable, part of API?
 - `sorted(list, reverse=True)`, part of API
 - **Idiom:**
 - Sort by two criteria: use a two-pass sort, first is secondary criteria (e.g., break ties)
- ```
[("ant",5), ("bat", 4), ("cat",5), ("dog",4)]
[("ant",5), ("cat", 5), ("bat",4), ("dog",4)]
```

Compsci 06/101, Spring 2011

14.3

## Stable sorting: respect re-order

- **Women before men ...**
  - First sort by height, then sort by gender



Compsci 06/101, Spring 2011

14.4

## Python Sort API by example, (see APT)

- Sort by frequency, break ties alphabetically

```
def sort(data):
 d = {}
 for w in data:
 d[w] = d.get(w,0) + 1
 ts = sorted([(p[1],p[0]) for p in d.items()])
 print ts
 return [t[1] for t in ts]
```

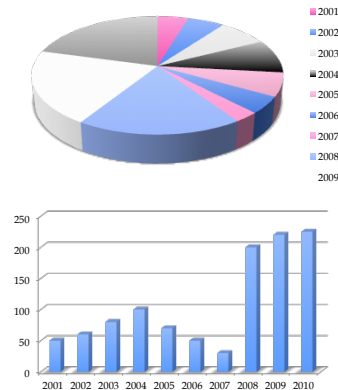
- How to change to high-to-low: `reverse=True`
- How to do two-pass: `itemgetter(1)` from operator

## Jotto: The Program Architecture

- You write `jottoModel.py`
  - This is the guts, brains, state of the program
  - Keeps track of how the game is progressing
  - Functions communicate via global state
    - Maintain state *between* function calls
- We provide two views: **command-line** and **GUI**
  - Allow you to *view* and *control* the model
  - Both view-controllers: `jottoMain.py` and `jottoGui.py` know about the model, but model doesn't know about them (atypical MV/MVC)

## Model View Controller (MVC)

|      |     |
|------|-----|
| 2001 | 50  |
| 2002 | 60  |
| 2003 | 80  |
| 2004 | 100 |
| 2005 | 70  |
| 2006 | 50  |
| 2007 | 30  |
| 2008 | 200 |
| 2009 | 220 |
| 2010 | 225 |



## What are global variables?

- Accessible to all functions in a module
  - Declared as `global varname` in functions
    - Mandatory for writing global variables
    - Good idea for reading global variables
  - Defined at 'top' of module for reading code
- Why are global variables a bad idea?
  - Very difficult to reason about correct code
  - Very difficult to extend, modify code
  - Non-locality makes maintenance difficult