Announcements
- Next assignment is on Web.
- Due WdS, Dec 5. No extensions!
- Uses a dictionary, which we will not discuss in class until WdS. You can get started on other parts of the program though.
- Start reading Chapter 5 on dictionaries.

Database
- A database is a program that
  - answers questions about some data,
  - lets you change it in specific ways.
- You organize the data differently depending on the kinds of questions you have to answer.
- On our CA congressional data, two kinds of queries: Who won in district 14? What district does Nancy Pelosi represent?

New Concepts
- How to change a global variable in a function
- How to stop a for loop early
- Both of these are ways of avoiding Python’s biases about how to do things.
- Python’s biases are there to keep you organized, so use these new ideas rarely and carefully.

Structure of Database Program
- Two parts:
  1. Read input file and store relevant data in some kind of data structure (eg, list of lists)
  2. Have a loop which asks for queries and answers them using the data structure.
- We’ll start with Part 1, same for both kinds of query.
- Extract names and votes from each line, pick winner in each district, and store winners in a list of lists -- [district, name]
ReDo the `getVotes()` function
- Extract names instead of parties.
- Name might be two or three words.
- Get rid of those stars!
- Return a list [name, votes]
- This is all review. If you really need to review, don’t look at it and try and write your own version.

Find Winner - Warm Up
L = [56,3,78,23,45,87,12]
max = None
for num in L:
    if max == None or max < num:
        max = num
print max

- Trick question: Does this program crash when `max` has value `None`?
- No! When `max == None` is True, then `max < num` never gets evaluated.

Find Winner
- Just work on one district first. Break down problem into smaller pieces!
- Similar to finding maximum in list.
- Store winner in a list (the “dataBase”) as a record which is itself a list [district, name].

Every Congressional District
- When we start a new district
  - Store the winner of the current district in the data base.
  - Reset winner to None.

But there are 53 districts…?
- What happened to the last one?
- Need to repeat code to store the winner.
- Make a function to avoid repeating code.
- Woops! We want to modify a global variable…

Changing a Global Variable
```python
def store (pair,dist):
    global database
    if pair != None:
        item = [dist,pair[0]]
        database = database+item
```
- `global database` asks permission to change a global variable in the function.
- Like exceptions, use it carefully, only when you know what you are doing.
- You usually don’t have permission because it is really easy to make a big mess.
Easy Queries

- Who is the representative in District 4?
- District is index into list
- Detail: list items are numbered 0,1,2,…52, but districts are numbered 1,2,3…53. So really \texttt{district - 1} is index of the record for that district.

More difficult queries

- What if we want to answer:
  Select a representative: Pelosi
  Nancy Pelosi is the Representative of District 8
- Read through all districts until we find the string "Pelosi"

Read list, look for name

- Use another for loop.
- Check for last name match.
- What if the name is not found?

Ending Early

- After we found the last name, why read the rest of the list? Doesn't hurt, but wastes time.
- Use the \texttt{break} statement to end loop early

Break

- Immediately jumps out of the loop.
- Usually used with for loops. Avoid with while.

```python
import random
num = random.randint(1,10)
print("I'm thinking of a number between 1 and 10")
print("You have three guesses.")
for i in range(3):
    guess = raw_input("Guess: ")
    if int(guess) == num:
        print "Right!"
        break
    else:
        print "Wrong!"
print "My number was", num
```

Practice and Study

- Try writing the program on the previous slide using a while loop with no break statement.
- Try writing the version of the getVotes function that returns the name and number of votes.
- Start reading pages 147-154 so you understand Wds's lecture better.