ECS 10

2/20

Announcements

- Program 5 Due Tuesday.
- Review exercises for ideas from early in the course in "Slides, Readings and Handouts", under Mon 2/18.
- Can work on this with TAs in lab hours Thursday.
- Also can work on Program 5.
- **Midterm 2** on Friday 3/1.

Last time

- Program to look up Congressional Rep by district.
- Left for you to finish program, for practice. Don't hand this in.
- **Algorithm** – plan of the program.

First loop:
Read file, put relevant data in list
Second loop:
Get user input = index
Look up the requested data by index

What if…

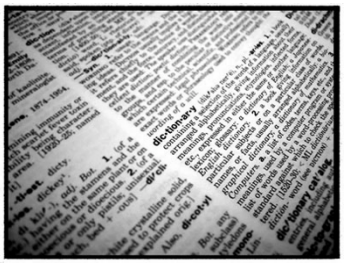
- …we wanted to loop up the District number, given the name of the Representative? (like a phone book)
- What would be the algorithm?

Does it "scale up"?

- This is one of the key questions we have about an algorithm in computer science. Does it work well as the data gets bigger?
- When there is one question per program and the list of data is short, this algorithm is fine.
- But what if SISWeb worked like this? Every time you wanted to check if a course was full, it would have to read the whole UCD course catalog.
- There's got to be a better way……

Dictionary jargon

- Words are **keys**.
- Definitions are **values**.



Dictionary Entry

myDict = {"ferret":"Ferdinand"}

key value

- Curly braces {} indicate a dictionary
- {} is the empty dictionary.

Looking up an Item

- Look up a value, use key as index.

name = myDict["ferret"]

variable holding value key

- Use square brackets for indexing, just like a list.

Bad index

```
name = myDict["ferret"]
```

- If there is no key "ferret" stored in the dictionary, then this causes the program to crash.
- Always check before indexing with a key!
- How to check:

```
if "ferret" in myDict:
```

Dictionaries are mutable

```
>>> hours = {}
>>> hours["Mon"] = 8
>>> hours["Tues"] = 9
>>> hours["Mon"] = hours["Mon"]-1
>>> hours
{'Mon': 7, 'Tues': 9}
```

- You can treat dictionary items as variables, and change them.
- You can add new items to the dictionary.
- You can also delete item, using the del() function

Dictionary Query Program

- Loop 1: Read in data, build dictionary.
  - Reads whole file. Only done once.
- Loop 2: get queries from user and answer them.
  - Goes right to the data it needs.
  - Does NOT have to look at all the data to answer each query.
  - This is important for huge data, eg. Google, airline reservations, SISWeb.