ECS 10

2/22

Announcements

- Program 5 due **Tuesday 2/26 10pm**
- You do not have to get “United States of America” to match “United States”.

- Midterm **Friday 3/1**

Things to learn in Prog. 5

- Dictionary – putting data in, getting it out.
- String methods
    - strip()
    - split()
    - replace()
    - lots more….try Googling “Python string methods”

Today’s example

- Read in file, build dictionary
- Illustrates things you’ll need:
    - Dictionary
    - More string methods
- Data: my Twitter

Computer alphabets

- Everything in a computer is zeros and ones.
- A bit is either a zero or a one.
- There really are no strings, ints, floats, lists….just bits.
- Integers are numbers base 2 – this is binary.
- Letters are encoded as binary numbers.

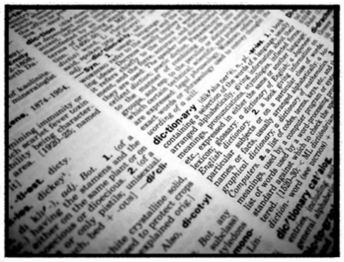
Character encodings

- How many bits to waste on a character?
- Used to be 8 (ASCII was the encoding)
- This allowed a total of 128 characters. Not enough!
- Unicode uses up to 21 bits.
- Utf-8 is a subset of Unicode (a codec) that most Web browsers handle.

```
twFile = open(“tweets.txt”,”r”,”encoding=“utf-8”)
```

Dictionary jargon (again)

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



Using the dictionary

- Make it:

```
twDict = {}
```

- Add an entry:

```
twDict[tweeter] = 1
```

- Check before you try to get an entry:

```
if tweeter in twDict:
```

- Change an entry:

```
twDict[tweeter] = twDict[tweeter]+1
```

For loop on a dictionary

```
for x in M:
```

- If M is a dictionary, then x is each key in turn

```
for tweeter in twDict:
```

- **tweeter** will be each twitter handle that is a key in the dictionary (in pseudo-random order)