

PFTW: lists, loops, accumulation

- We want to convert Shakespeare to pig-latin
 - Change each word, need to loop over words
 - Change one word, find vowel and make changes
- `f y cn rd ths y cn rd nthng`
 - How would we create this string from ...
- How do we find all links from a web page?
 - How do we find one link in a web page
- How do run a random walk simulation 1000 times?
 - Recording statistics as we go...

Compsci 101, Fall 2012

7.1

Start simple, solve problems

- Ultimately, how do we get to Ngram Viewer?
 - books.google.com/ngrams
 - Try: North Carolina, South Carolina, Virginia, Georgia
- Count how many words start with 't', how many times 'coward' occurs, how many times ...
 - Loop over collection of words
 - Increment a count in certain conditions
- How many times is a song played on Spotify?
 - Loop over songs, increment a count

Compsci 101, Fall 2012

7.2

Standard accumulation idiom

```
def count(collection, word):  
    total = 0  
    for elt in collection:  
        if elt == word:  
            total = total + 1  
    return total
```

- How do we count 'scarlet' in *Scarlet Letter*?
 - Or dagger in *Hamlet* or *Romeo*?
 - Or friend in *Little Brother*?
 - Or CGAT in a genome?

Compsci 101, Fall 2012

7.3

Accumulation revisited

```
def count(collection, letter):  
    total = []  
    for elt in collection:  
        if elt.startswith(letter):  
            total.append(elt)  
    return total
```

- How do we find words that start with 't' in bible?
 - Or words that start with 'U' in *The Illiad*?

Compsci 101, Fall 2012

7.4

Anatomy of a list

- **Lists are collections of elements**
 - Create: `[1,2,3]`, `["apple", "pear"]`, `[]`
 - Heterogeneous, lists of
 - Index and slice (like a string, starting at 0)
 - Append (and extend) by *modifying* the list
- **Lists are iterable sequences**
 - `for x in lst:`
- **Many functions in common with strings**
 - `len`, `max`, `min`
 - Also methods after `.`, `lst.count(1)`, `lst.index(elt)`

CompSci 101, Fall 2012

7.5

Zephyr Teachout

- **Fordham Law Prof**
 - Duke law school
- **Internet Campaign**
 - Howard Dean 2004
 - <http://cnet.co/GAGR0V>



We have so many different parts of our site, right now. We actually build software here. The Internet is really essential for people to achieve some more power with the process.

We have three full-time programmers and our database team has people with (scripting language) PHP and we have a bunch of volunteer coders, as well as a whole community of over 100 people working in open source on Dean-related projects. I am not the person that knows hardware.

CompSci 101, Fall 2012

7.6

Debugging APTs: Going green

- **TxMsg APT: from ideas to code to green**
 - What are the main parts of solving this problem?
 - Transform words in original string
 - Abstract that away at first
 - Finding words in original string
 - How do we do this?
- ```
def getMessage(original):
 ret = ""
 for word in original.split():
 ret = ret + " " + transform(word)
 return ret #initial space?
```

CompSci 101, Fall 2012

7.7

## Why do we write code this way?

- **Structure of code is easier to reason about**
  - Harder to develop this way at the beginning
  - Similar to accumulate loop, build on what we know
- **We can debug pieces independently**
  - What if transform returns "" for every string?
  - Can we test transform independently of getMessage?
- **How is this similar to Bagels APT?**
  - Loop over ...
  - Accumulate ...
  - Return ...

CompSci 101, Fall 2012

7.8