List Comprehensions, Lists, Oh My!

- Python in the news? http://bit.ly/y7yeXX
  - How hard is it to program?
- TIMTOWTDI, aka skinning cats
  - Is there a "best" way?
- What is TxMsg about *conceptually*?
  - Easy to get lost in some details, which ones are they?
  - Breaking a string into "words": `.split()`
  - Putting a list of "words" back together: `.join()`
  - What are arguments to these methods? Methods of …?

Yahtzee APT interlude and motivation

- APT: http://bit.ly/yahhhtzee
  - How do we create these shortened URLs?
- How do we solve this?
  - What do we loop over?
  - What do we do for each iteration of loop?
- How do we transform data to make it easier to solve
  - What's the largest number in a list? `max(lst)`
  - Where does the list come from?

List Comprehensions

- Creating a list from another list, two decisions:
  - Is new list the same size as original, or smaller?
  - Are elements the same or related by some correspondence?

```
words = ["bear", "lion", "zebra", "python"]
w2 = [w for w in words if some_property(w)]
w3 = [f(w) for w in words]
w4 = [1 for w in words if some_property(w)]
```

- Once we have list can apply list functions
  - We have: len, sum, max, min
  - Can "invent" others by writing functions

List Comprehensions Again

- Transformative approach can scale differently
  - Functional programming: code generates and doesn't modify
  - Basis for (ultra) large scale mapreduce/Google coding

```
w = [expr for elt in list if bool_expr]
w = [f(w) for w in list if bool_expr(w)]
w = [list.count(x) for x in range(1,7)]
```

- Why are abstractions important?
  - Reason independently of concrete examples
    - Generalize from concrete examples
  - http://www.joelonsoftware.com/articles/LeakyAbstractions.html

danah boyd

Dr. danah boyd is a Senior Researcher at Microsoft Research, … a Visiting Researcher at Harvard Law School, … Her work examines everyday practices involving social media, with specific attention to youth engagement, privacy, and risky behaviors. She recently co-authored *Hanging Out, Messing Around, and Geeking Out: Kids Living and Learning with New Media.*

"From day one, Mark Zuckerberg wanted Facebook to become a social utility. He succeeded. Facebook is now a utility for many. The problem with utilities is that they get regulated."

http://bit.ly/ySwjyl

Compsci 6/101: I ❤ Python

- Techniques for looping
  - Loop over sequences by sequence value
  - Loop by indexing, or by index and value: enumerate
  - While loop: as long as condition holds, e.g., game not over
- Techniques for transforming data
  - One domain leads to solutions, other much harder
  - Identify music with sound-hound/shazaam
  - Encryption: transform data to hide it, but …
  - APT AnagramFree

Loop over sequence with index

- Index useful in accessing elements in order
  - Sometimes need adjacent elements, `i-1`, `i`, and `i+1`
  - Often need both index and element, see enumerate below

```
for i,fr in enumerate(['a','b','c']):
    print i,fr
```

- No more *powerful* than looping over range, why?
  - Idiomatic programming, helps to know vocabulary
    - Syntactic sugar
  - Not necessary, use `for i in range(0,len(seq)):`

Indefinite loop: while … *interactivity*

```
wrong = 0
while wrong < max_wrong:
    guess = raw_input()
    if not good_guess(guess):
        wrong += 1
    else:
        #process the guess here
```

- Suppose, for example, play http://www.hangman.no
  - What happens if you loop while True:
  - Break out of loop with break
  - See code in ***GuessNumber.py***

Interactive programs

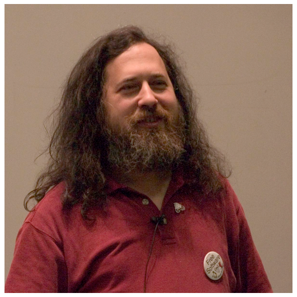
- How do you obtain input from the user?
  - If using the keyboard and a console?
  - If using a web-browser or a GUI program?
  - What about "bad" input?
- Developing and designing loops
  - Reasoning about loop "test", while: false loop done
  - What about initial evaluation of loop "test" or "guard"
- Formal reasoning can help, intuition too?
  - Hard to get better at intuition?

From guessing numbers to transforms

- With good-guessing, optimal number of guesses?
  - How do you reason about this?
  - Don't think of the number, but range of possibilities
- How did Watson do in Jeopardy?
  - http://to.pbs.org/fRQz6p
  - How does Watson transform questions so understandable?
- Sometimes changing data leads to solution
  - Transformations depend on problem and solution space
  - If the answer is 'yes', if the answer is 'Waterloo', …

Richard Stallman (b.1953, Hopper '90)

- Transformed programming
  - Free Software Foundation
- "World's Best Programmer"
  - Gnu/Linux: g++, emacs
  - Believes all software should be free, but like "free speech", not "free beer"
  - Won MacArthur award for his efforts and contributions
  - League for Programming Freedom
    - It's about free, not open



Aside: Transform for AnagramFree APT

- How do you know when two words are anagrams?
  - Possible to tell with letter-count fingerprint
  - `"apple"` -> [1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,2,0,0,0,0,0,0,0,0,0,0]
  - Can we create this fingerprint? How?
  - Alternative fingerprint: sort the letters

```
sorted("apple") is … why?
''.join(['a','b','c']) is "abc"
```

- If the data is transformed, still some work to do
  - #Anagrams in ['dgo', 'aet', 'dgo', 'aet', 'aet']?