Announcements

• Exam 2 Thursday
• Reading and RQ for next week – coming…
• Assignment 7 due April 14
• APT 9 due today
  – Doing extra ones – good practice for exam
• No Lab this week!

• Today:
  – Reviewing for the exam
Snarky Hangman

• Version of Hangman that is hard to win.

• Program keeps changing secret word to make it hard to guess!

• User never knows!

• Once a letter is chosen and shown in a location, program picks from words that only have that letter in that location

• Program smart to pick from largest group of words available
Snarky Hangman - Dictionary

• Builds a dictionary of categories
• Start with list of words of correct size
• Repeat
  – User picks a letter
  – Make dictionary of categories based on letter
  – New list of words is largest category
    • Category includes already matched letters
    • List shrinks in size each time
Snarky Hangman Example

• Possible scenario after several rounds

(secret word: lucky ) # words possible: 33
Progress: _ u _ _ _
letters missed: a b e i o s
guess a letter: c

• You currently have a list of all words with u the second letter. From that build a dictionary of list of words with no c and with c in different places (show count of number of words in each list):

| cu___  | 2 |
| _uc__  | 2 |
| _u___  | 21 |
| _u_c_  | 8 |

Only 2 words of this type
Only 2 words of this type
Choose “no c”, most words, 21
Only 8 words of this type
Exam logistics

• Only need a pen or pencil
• No scratch paper
• See the reference sheet of Python information you will get with the test (see resources page)
• Closed book, closed notes, closed neighbor
• Covers lecture, lab and assigned reading
• Have put old quizzes back up as quiz review
  – This is NOT for a grade, for studying only
Understand old and new topics

• Old topics: if, for, while, lists, strings
• list comprehension, enumerate
• Files – write code - Will give you a file already opened and ready for reading
• Sets, Dictionaries – write code – create and use
• Understand items on Python review sheet on resources page
• HAVE NOT COVERED TOPICS – regular expressions or recursion
The best way to study

• Write code on paper!

• Resources page has old tests and solutions
  – Try writing code, then look at solutions

• Rewrite an APT

• Rewrite code we did in lecture

• Rewrite code we did in classwork or lab
Looping by index or by element

- Strings and lists: use either
  - range(len(x)) for index, can get element
  - enumerate(somelist)

- Sets and Dictionaries: element only
  - Loop over d or d.keys() for dictionary
  - The keys are a set, so similar to set loop

- Which is best when choice? It depends!
  - Can you get element from index?
  - Can you get index from element?
Questions
bit.ly/101sp16-0405-1
Unpacking a list comprehension

\[ \{ f(x) \mid x \in \text{foo} \text{ if condition with } x \} \]
\[ \{ w \mid w \in \text{words} \text{ if } w.\text{endswith('e')}, w \} \]
\[ \{(w, \text{words.}\text{count}(w)) \mid w \in \text{set(words)}\} \]

– Always possible to use a loop

```python
build = []
for x in foo:
    if condition with x:
        build.append(f(x))
```

```python
build = []
for w in set(words):
    build.append((w, words.count(w)))
```
Set Concepts

- Set union, intersection, difference
  - s.intersection(t) is the same as s & t
  - s.union(t) is the same as s | t
  - s.difference(t) is the same as s - t

- Sets aren't in order during iteration
  - Convert to list, create from list
  - Sets are really, really efficient for add/search
Dictionaries

- Build a dictionary
  - Counting dictionary
    - string to number
  - Grouping dictionary
    - string to list of items related

- Use a dictionary
  - Get values
  - Get keys
  - Get key, value pair
Questions
bit.ly/101sp16-0405-2
Now go over Test Practice problems