CompSci 101
Introduction to Computer Science

November 18, 2014

Prof. Rodger
Announcements

• Reading for next time on calendar page
  – RQ maybe

• Assignment 7 due Thursday

• APT 9 is out today

• Do not discuss exam until it is handed back
Snarky Hangman

• Demo
• 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
    • Matched letters
    • Letters guessed by not chosen
    • List shrinks in size each time
Regular Expressions

- Powerful language for matching text patterns
- Part of the compiler process
  - Can write a regular expression for each type of word in a programming language
  - Example
    - Key words – if, else, elif, while
    - Integers – 456, 78, 2, -56
    - Float – 3.14, 7856.2345
    - String – ‘word’, “this is a phrase”
    - Special symbols – [ ] + %
Regular Expressions

• a - a
• a* - a repeated 0 or more times
• a+ - a repeated 1 or more times
• a? – a 0 or 1 time, so a is optional
• ^ - match at the beginning of the string
• $ - match at the end of the string
• . – matches anything
• [abc] – match a, b, or c
• [a-z] – match any character from a to z
• [^a] – match any character but a
More on regular expressions

- | - or
- \b - word boundary
- \s - whitespace character
- \d – match any digit
- When using backslashes – must use r in front of string
Regular expressions with re

- import re
- re.sub(pattern, repl, str) – return string that replaces the pattern matches with repl in string str – looks from left end of string
- re.compile() – create a pattern
- re.findall()
- See code examples
More on sort

- Import operator
  - fruit = [("pear",5),("apple",9)]
    * fruit = sorted(fruit)
    * fruit.sort() OR fruit = sorted(fruit)
  - arguments
    * key=itemgetter(0)
    * reverse=True