CompSci 101
Introduction to Computer Science

Nov 14, 2017

Review for exam

Prof. Rodger

Announcements

- Exam 2 Thursday
- Reading and RQ start after Thanksgiving Break
- APT 7 due tonight
- No Lab this week!
- No Consulting hours Thursday night
- Yes we have class on Tuesday, Nov 21!

- Today:
  - Reviewing for the exam

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 RQ 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/101f17-1114-1

Unpacking a list comprehension

```
[f(x) for x in foo if condition with x]
[w for w in words if w.endswith('e')]
[(w,words.count(w)) for w in set(words)]
```

- Always possible to use a loop

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

```
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/101f17-1114-2

Now go over Test Practice problems