CompSci 101  
Introduction to Computer Science

Feb 24, 2015  
Prof. Rodger

Announcements

• Reading and RQ11 due next time
• Assignment 5 out today - Hangman
• APT 4 due, APT 5 out
• Exam 1 back today
  – Exam regrades go to Prof. Rodger
  – Write on the front the problem and what the issue is
• Today
  – sets
  – Finish lecture notes from last time

More on List Comprehensions

• What is the list for the following:
  1) [j+1 for j in range(20) if (j%3) == 0]
  2) [i*2 for i in [j+1 for j in range(20) if (j%3) == 0] if i*i > 19]

More on List Comprehensions
bit.ly/101S15-0224-01

• Problem: Given a list of strings, return the longest string. If there are more than one of that length, return the first such one. ['kiwi', 'plum', 'orange', 'lemon', 'banana']
Write a list comprehension for this problem
Python Sets

- Set – unordered collection of distinct items
  - Unordered – can look at them one at a time, but cannot count on any order
  - Distinct - one copy of each
- Operations on sets:
  - Modify: add, clear, remove
  - Create a new set: difference(-), intersection(&), union (|), symmetric_difference(^)
  - Boolean: issubset <=, issuperset >=
- Can convert list to set, set to list
  - Great to get rid of duplicates in a list

Summary (from wikibooks)

```python
# A new empty set
set1 = set()
# Add a single member
set1.add("cat")
# Add several members
set1.update(["dog", "mouse"])
# Remove a member - error if not there
set1.remove("cat")
# Iteration or "for each element"
for item in set1:
    print item
# Length, size, item count
print "Item count":, len(set1)
# Test for emptiness
isempty = len(set1) == 0
# Initialize set from a list
set2 = ["cat", "dog"]
# Intersection
set3 = set1 & set2
# Union
set4 = set1 | set2
# Set difference
set5 = set1 - set3
# Symmetric difference (elements in either set but not both)
set6 = set1 ^ set2
# Subset test
issubset = set1 <= set2
# Superset test
issuperset = set1 >= set2
# A shallow copy (copies the set, not the elements)
set7 = set1.copy()
# Clear, empty, erase
set8.clear()
```

Creating and changing a set

```python
colorList = ["red", "blue", "red", "red", "green"]
colorSet = set(colorList)
smallList = list(colorSet)
colorSet.clear()
colorSet.add("yellow")
colorSet.add("red")
colorSet.add("blue")
colorSet.add("yellow")
colorSet.add("purple")
colorSet.remove("yellow")
```

- See setsEasy.py

Set Operations

```python
UScolors = set(["red", "white", "blue"])
dukeColors = set(["blue", "white"])
print dukeColors.union(UScolors)
print dukeColors | UScolors
print dukeColors.intersection(UScolors)
print dukeColors & UScolors
print dukeColors.difference(UScolors)
print dukeColors - UScolors
print UScolors - dukeColors
print dukeColors ^ UScolors
print UScolors ^ dukeColors
```

- See setsEasy.py
poloClub = set(['Mary', 'Laura', 'Dell'])
rugbyClub = set(['Fred', 'Sue', 'Mary'])

Question 1:
print [w for w in poloClub.intersection(rugbyClub)]
Question 2:
print [w for w in poloClub.union(rugbyClub)]

lista = ['apple', 'pear', 'fig', 'orange', 'strawberry']
listb = ['pear', 'lemon', 'grapefruit', 'orange']
listc = [x for x in lista if x in listb]
listd = list(set(lista)|set(listb))

Question 1:
print listc
Question 2:
print listd