Homework 3: Extracting information with Dictionaries

Due Wednesday, October 15, 7:00a

What to hand in via Canvas

- hw3-exercises.py program (all your code from part 1)
- hw3-feedprocessor.py program (all your code from parts 2, 3, and 4)

Part 0: Download the starter code

You'll need the homework files. Get them by typing the following lines in terminal:

```
cd ~
./getHW.sh 3
```

Don't forget that if you already have Eclipse open, you may need to right click on the myhw project and "refresh" it, to make the new folder 3 appear.

Part 1: Exercises

The following code should be at the top of your starter file, hw3.py:

```
pet_count = {'dog': 3, 'cat': 2, 'rabbit': 7, 'fish': 5}
```

The dictionary `pet_count` represents the number of pets in a household: 3 dogs, 2 cats, 7 rabbits, 5 fish. Complete the exercises embedded in the code.

When you work on earlier exercises, the code in exercise 8 will produce an error – that is intentional and something for you to fix in exercise 8.

Your output should look like:

```
== 1 ==
7
== 2 ==
8
== 3 ==
4
== 4 ==
There are 2 cats.
== 5 ==
fish
hamster
dog
rabbit
cat
== 6 ==
fish: 5
hamster: 4
dog: 3
rabbit: 8
cat: 2
== 7 ==
there are 4 hamsters
== 8 ==
cat: 2
mouse: 0
pig: 0
velociraptor: 0
frog: 0
fish: 5
hamster: 4
hippo: 0
```

Part 2: Printing posting users

Your next task is to print out all the names of users who posted to the Facebook group. See the file for more details.

The *first* part of the of part 2 should look like (It's a long list, so I have clipped it!):

```
== part 2 ==
Danny Cohen
Wei Kong
Perry Lin Meas
Wei Kong
Perry Lin Meas
Wei Kong
Sean Munson
Wei Kong
Samuel Marks
Sean Munson
Samuel Marks
Perry Lin Meas
Sean Munson
Perry Lin Meas
Sean Munson
Perry Lin Meas
Sean Munson
Sean Munson
Perry Lin Meas
Sean Munson
Perry Lin Meas
Sean Munson
Perry Lin Meas
Sean Munson
Perry Lin Meas
Sean Munson
Kyle Nesburg
Chia-Fang Chung
Kyle Nesburg
Sean Munson
Yoanna Dosouto
```

… and so on

Part 3: Counting poster contribution frequency

Your next task is to count how often each user posts to the group. You will need to use a dictionary to keep a running count of how many times each user posted or commented (a combined count for each) to the group:

- The starter code begins by creating an empty dictionary, called `post_count`. Note that you will need to open the file again, because looping through its lines in Part 2 will have "consumed" those lines.

- Once you've extracted a user name (the same thing that you printed out in part 2), think of it as a key in the dictionary, whose value is the number of posts encountered by that user so far.

  If the key is not in the dictionary, set the value to 1. If the key is already in the dictionary, set its value to 1 more than its current value. (In the language of the accumulation pattern, the count is our accumulator variable for that user.)

- Once your program finishes reading through the file, print how many times each user posted or commented.
  - To do this, create a string of the form: "`<name> <X>`", where <name> is the user's name, and X is the number of times they posted. (See example screenshot for what it should look like.)
  - Print that string.

Put your code in `hw3-feedprocessor.py`.

The output of part 3 should look like the following, though your usernames may be in different order (remember that dictionaries are in arbitrary order!):

```
== part 3 ==
Yoanna Dosouto 10
Justin Woodum 8
Anne Zheng 2
Sean Munson 29
Max Schreiber 1
Autumn Grassel 2
Wei Kong 4
Perry Lin Meas 11
Kendall McGinnis Avery 2
Samuel Sun 1
Tristan Shi 1
Jess Landquist 1
Sabrina Weschler 2
Nitaya Munkhong 2
Chip Connor 1
Xiaochen Yu 5
Samuel Marks 2
Nicole Tilly 2
Jessica Bao 3
Candelario Peraza 1
Danny Cohen 3
Natalee Ouzts 1
Kyle Nesburg 2
Chia-Fang Chung 8
Sami West 7
Lisa Hu 1
```

Part 4: Counting word frequency in posts

For part 4, we will **count the word frequency in posts (*not* comments!)** to the Facebook group.

For this part:

- Use stripWordPunctuation() to remove punctuation from each word. This function will only work if you pass it a word (i.e., not a line).
  (we will briefly cover how to use functions such as this in class on Wednesday)
- Convert all text to lower case before counting.

Put your code in `hw3-feedprocessor.py`.

Example output below. It has been trimmed and yours may be in a different order.

```
== part 4 ==
 8
all 2
code 1
consider 1
supplement 1
go 1
disk 3
rocking 1
issues 4
http://www.tomshardware.com/faq/id-1957309/install-virtualbox-guest-additions-linux-mint.html 1
environment 1
answered 1
finally 1
does 3
under 2
sorry 3
updated 1
downloading 5
lecture 4
telling 1
trouble 3
we'll 1
prize 1
first_program.py 1
wednesday 1
try 1
team 2
unexpected 1
says 1
10a 1
second 1
video 1
pass 1
download 2
click 2
vmware 6
what 2
waiting 1
```

and so on… . Please note that your program may also leave a blank string in as a “word” because of some flaws in how we remove punctuation.

P0:Identify an API

Please see a separate assignment on Canvas -> Assignments -> Project (**to be posted in class on Wednesday**). That (short!) assignment will also be due next Wednesday at 7:00am.

Just For Fun:

Remember, you don't have to do this. Modify your code from part 3 to:

1. Separately keep track of comments and posts per user.
2. For an extra challenge, print the users in alphabetical order by name.