Homework 2: Extracting information

Due Wednesday, October 8, 9:00am

What to hand in via Canvas

- Screenshot of execution of your hw2-part3.py program, showing the output
- hw2-exercises.py program (all your code from part 2)
- hw2-part3.py program (all your code from part 3)

Part 0: Prepare

First, go back to the home directory and use getHW.sh to get hw2-part3.py and hw2-exercises.py:

```
cd ~
./getHW.sh 2
```

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 2 appear, with `hw2-exercises.py`, `hw2-part3.py`, `hw2feed.txt`, and `sherlock.txt`

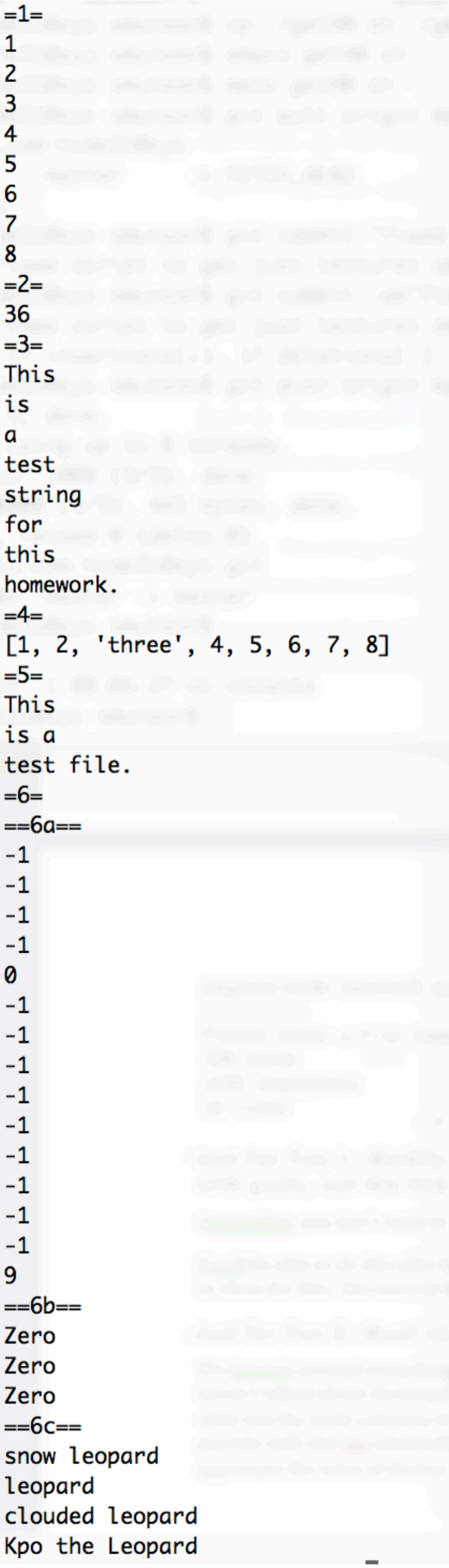
Part 1: Review additional string methods

Please review the slides from Supplement 2 (in the files section under Supplements) at your own pace. There is also code in the HW folder that will let you follow along in Python if that is your preferred learning style. Please make sure you are getting the CYU exercises right. We're available in office hours, or you can post to the Facebook group, if you run into trouble.

Don't skip over this, or you'll probably waste a lot of time in the exercises.

Part 2: Exercises

See the instructions in `hw2-exercises.py`. Your output should look something like:

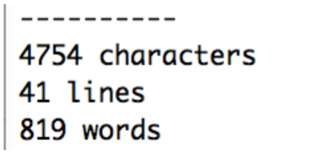


```
=1=
1
2
3
4
5
6
7
8
=2=
36
=3=
This
is
a
test
string
for
this
homework.
=4=
[1, 2, 'three', 4, 5, 6, 7, 8]
=5=
This
is a
test file.
=6=
==6a==
-1
-1
-1
-1
0
-1
-1
-1
-1
-1
-1
-1
-1
-1
9
==6b==
Zero
Zero
Zero
==6c==
snow leopard
leopard
clouded leopard
Kpo the Leopard
```

Part 3: Counting lines, characters, and words

See the instructions in `hw2-part3.py`

Here is an example of what your output for this should look like:



```
----------
4754 characters
41 lines
819 words
```

Just For Fun 1: Modify your code from part 3 to count only the lines with posts, not the lines with authors.

(remember, you don't need to do just for fun exercises)

You'll be able to do this after Monday's lecture. You'll need to use conditionals and find() or slices for this. See exercise 6 for some ideas.

Just for Fun 2: Most common word in Sherlock Holmes?

We haven't covered everything you need in Python to do this yet (in particular, we haven't talked about dictionaries yet). But think about how a program could figure out what was the most common word in a document, say the Sherlock Holmes text that I provide with this hw (sherlock.txt). Thinking about this problem now will help you appreciate the value of dictionaries when we learn about them.