CIS 122 Project 4   Read, write text files
and web page text files

Winter 2015

Reference

Text, chapter 10.3 **Reading text files**

You'll find "The '**For Line in File**' Technique" very helpful

Text, chapter 10.4 **Files over the Internet**

Good news: Almost the same as reading a text file from your own computer.

Text, chapter 10.5 **Writing Files**

Notice that writing a file either creates a new file or overwrites an existing file.
Notice too the need to end each line with a '\n' newline character.

Text, chapter 7.3 String methods

```
letters = 'ann'
found = mystring.startswith(letters)  # Assigns True or False to foun
found = mystring.endswith(letters)
found = letters in mystring
```

Also see `mystring.strip(), mystring.split(',')`

Project

4a 5 points

Read the **planets.txt** file.
Store the data you read into a planets_list.
Print each planet's name on a separate line.

4b 5 points

Read the planets2.txt file.

Each line has 2 pieces of data, separated by a comma **,**

Planet name such as Jupiter
Average distance from sun, in millions of kilometers, such as 779 for Jupiter

```
Jupiter,779
```

Convert the string with the distance into an integer.

After reading the data into a list (or lists), print the data from those lists, one planet per line:

```
Jupiter    779 million km from Sun
```

XC 2 points

Define a km2miles function that given km **returns** miles.
Modify your print to show planet name, average millions of km from Sun, and average millions of miles from the Sun, somewhat like this:

```
Jupiter    779 million km  =  484 million miles from Sun, on average
```

4c 5 points

Read data from an Internet site, the **hopedale** data from a Canadian Internet site (text 10.4 lists it):

`http://robjhyndman.com/tsdldata/ecology1/hopedale.dat`

The data itself is a series of numbers relating to fur trapping in the mid 1800's.
Print the data as you read.
Typical problems: Typing errors with the URL.

4d 5 points

Read the **yob1994short.txt** file into a suitable structure a list of **sublists** (name, gender, count) or parallel lists (namelist, gender list, count list).
After reading in the data, use a **while loop** to allow user to **request a name to search for.**
Look for: **Dana**
Print the search results "Dana not found" or something similar to this

```
Dana   F   217
Dana   M    25
```

4e 5 points

Convert project 4d to project 4e by changing the filename to read from **yob1994short.txt** to **yob1994.txt**

Doing so means your program will read about 25,000 lines of text instead of 22 lines of text.

Did you notice any special slowdown in reading the data or searching for a name?

If your 4d printed out the entire contents of the file, you will need to suppress that printout.
Only print the results of your search.

4f 15 points

Add additional search options to your project 4e:

Starts with Dan
Dan, Dana, Daniel, Danielle would all match this request, but **Odana** would **not**.

Ends with ary
Ary, Gary, Mary would all match this request, but **Larry** would **not**.

Contains **ari**
Ari, Cari, Darielle, Mari would all match this request, but **Arail** would **not**.

Each time you conduct a search, your user must be able to tell you what kind of search to do:
Exact March
Starts with
Ends with
Contains

4g 5 points

Modify your 4f project to **write** starts with, ends with or contains **results to a text file**.

4h 5 points

Modify your 4f or 4g project to get its data from the Internet from this url:
http://www.cs.uoregon.edu/Classes/15W/cis122/data/yob1994.txt

XC 5 points

Modify 4f or 4g to use **babynames.txt**
You will find it both in Blackboard as a file, and on the UO CIS website – use either source.

It contains a **year** field, and a total of some 160,000 records.

Your searches will now retrieve 4 items from the data
name, gender, count, year
You should display all four items.

Could you notice the extra time needed to read and search 160,000 items?