Understanding Data

POTD 17 (http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/):

```
# Write a function to open and read a file named StarWars.txt
# from http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/StarWars.txt
#
# Let's practice opening a file locally and via a URL (internet).
# That is,
# - for the first solution, you will download the file and
#   save it to your machine (same folder where your .py file is)
# - for the second solution, you will open the file via a URL
#
# The function will return 2 things (in order)
#   1. The number of StarWars' fans
#   2. The total number of respondents
#
# There are many fields for each line. You only need the following fields to
complete this exercise:
#    RespondentID  (0)
#    Have you seen any of the 6 films in the Star Wars franchise?  (1)
#        -- possible value "Yes" or "No"
#    Do you consider yourself to be a fan of the Star Wars film franchise?  (2)
#        -- possible value "Yes" or "No"
#
#  Note: There are exactly 2 header lines.
# we want to count responders and star wars fans, to compare




def count_fans_local(filename):
   infile = open(filename, 'r')
   line_cnt = 0 # i.e., number of respondents
   number_of_fans = 0 # note, if this isn't outside of the for loop, will get 0
   for line in infile:
       line_cnt += 1
       # print(line_cnt, line)
       columns = line.strip().split(',')
       # print(columns)
       if columns[2] == "Yes":
           number_of_fans += 1
   infile.close()
   return number_of_fans, line_cnt
print(count_fans_local("StarWars.txt"))

# url solution
import urllib.request
def count_fans_online(url):
   data_stream = urllib.request.urlopen(url)
   respondents = 0
   fans = 0
```

```python
    for line in data_stream:
        columns = line.decode("UTF-8").strip().split(",")
        respondents += 1
        if columns[2] == "Yes":
            fans += 1
    return fans, respondents
print(count_fans_online("http://www.cs.virginia.edu/~up3f/cs1110/practice-of-the-day/StarWars.txt"))
```

Opening Files from the Internet:

import urllib.request

urllib.request is a built-in library that connects Python to the internet

data_variable = urllib.request.urlopen(“link”)

The link must be a string (in quotes if directly typed)

It comes in coded, must be decoded

The scheme we use in this class is “UTF-8”

Use “UTF-8” to decode anything off of the internet

A function that displays the content of links:

```python
import urllib.request
def open(link):
    stream = urllib.request.urlopen(link)
    for line in stream:
        decoded = line.decode("UTF-8")
        print(decoded.strip())

link = input("Web page: ")
open(link)
```

Once decoded, we still need to process the string (e.g. string.strip() to remove “/n” and such)

Examples (http://www.cs.virginia.edu/~up3f/cs1110/examples/file/) open_url:

```python
import urllib.request
# We'll work with a dataset from the Internet. This dataset contains history temperature
# http://www.wunderground.com/history/airport/KCHO/2012/03/17/DailyHistory.html?format=1"
# Typically, we'd want to be able to access multiple day-month-year of this history file,
# put day-month-year in variables and form a URL
# Or, sometimes we'd want to let the user specify what date to load data

year = "2012"
month = "03"
day = "17"
url = "http://www.wunderground.com/history/airport/KCHO/" + year + "/" + month +"/" + day + "/DailyHistory.html?format=1"
# can change year, month, and day to get a new url, new data
# could write a function that takes year, month, and day user inputs and outputs data

# open the specified URL to read
# store the response object received from the server in a variable
stream = urllib.request.urlopen(url)
print(stream)
# prints: http.client.HTTPResponse object
# this is the not-decoded version

print("processing data")
is_header = True
# not using this here, but could be used to not process the header
for line in stream:
   decoded = line.decode("UTF-8").strip().split(",")
   # look online and see how the data is separate to know what to split by
   print(decoded)
```