# Homework 8 Twitter Data and Caching

In this assignment, you will get the data using Twitter's API and extract certain information from that data. You will use Tweepy - a simple python library for accessing the Twitter API.

You will write code to search for tweets using three different user entered phrases and cache any results from each search in a dictionary and a JSON formatted file (cache file). We have given you the code for writing the dictionary into a JSON file. Use the provided function **write_cache** to save the search results.

You will complete the following functions in the file **HW-Twitter.py.** The code will read from the **cache_twitter.json** file and write to it. **We recommend making a copy of the cache_twitter.json file before you start testing this code so that you can go back to the original file.** ***Be sure to execute HW-Twitter.py in the same directory with the cache_twitter.json file.*** You can't run this in just any directory (so don't use the *Run Python File in Terminal* command in Visual Studio Code).

**1. def read_cache(cache_file):**
This function reads from the JSON cache file and returns a dictionary from the cached (saved) data. If the file doesn't exist, it returns an empty dictionary. You may need to add encoding="utf-8" on the open of the file for reading.
*Note: see cache_example.py for example code*

**2. def get_tweet_data_with_caching(searchTerm):**
This function searches for the search term in the dictionary returned by **read_cache**. If the search term is in the dictionary, it prints "**using cache**" and returns the data for the search term. If the search term doesn't exist in the dictionary, it prints "**fetching**" and makes a new call to the API to get the data. If the search returns a response, it will save the response for the search term in the dictionary and then call the *write_cache* function to save the dictionary in the cache file. The function will return the response. If there was no response, it returns None. If there was an error during the search, it should print a message to inform the user and return None.

**3. def print_tweets():**
This function loops three times and each time it asks the user to enter a search term. It gets the results of the search term using **get_tweet_data_with_caching**. If the result has at least 5 tweets it prints out the tweet *text* and *created_at* values from the first 5 tweets with a blank line after each. If the result was less than 5 tweets, it prints `"Found x tweets"` where x the number of tweets returned. The original cache file has saved searches for 'Hope', 'Thanksgiving', 'Space', 'horse', 'Trump', 'Halloween' and 'USMI'.

***Sample Output: The highlights below are just to help you notice requirements.***

```
Enter search term: horse
Using cache
Text: RT @tubooks: Here's an amazing Q&amp;A with author @g_neri
```

about the inspiration behind his newest graphic novel GRAND THEFT HORSE!
https://t.c…
Created At:Thu Nov 08 18:13:27 +0000 2018 None

Text: RT @TheRealRaNon: Good work @Badwolf44659299 , Open their eyes to "Behold a Pale Horse"....Many unknown truths that will have many losing s…
Created At:Thu Nov 08 18:05:30 +0000 2018 None

Text: RT @CityoftheWeak: City of the Weak has a show on 11/08/2018 at 07:00 PM @ The Drunk Horse Pub in Fayetteville, NC https://t.co/3xrTAPRZqG…
Created At:Thu Nov 08 17:51:40 +0000 2018 None

Text: RT @sapphocles: Oh, don't be silly... Why would a guy who resembles an older #JFKJr be standing in the midst of a #Q formation, wearing one…
Created At:Thu Nov 08 17:26:47 +0000 2018 None

Text: RT @tubooks: Here's an amazing Q&A with author @g_neri about the inspiration behind his newest graphic novel GRAND THEFT HORSE!
https://t.c…
Created At:Thu Nov 08 17:15:59 +0000 2018 None

Enter search term: baby
fetching
Text: @itsEmfranco kimmy sna mameet u din jn c baby @JELAYPILONES08 the 1st big 4 ng teen batch 1..alam q xa ang bet u s… https://t.co/25FRCjPzPn
Created At:Tue Mar 26 01:14:49 +0000 2019 None

Text: não tem música em inglês q eu goste mais de cantar do q baby do justin
Created At:Tue Mar 26 01:14:40 +0000 2019 None

Text: opa opa tenho uma amiga caloura de Direito sai da frente q ela ta com muita moral

Futura Advogata👏🏻👏🏻 amoo https://t.co/2Dj3GJTg4z
Created At:Tue Mar 26 01:14:23 +0000 2019 None

Text: Da Baby & Q money really been in my rotation heavy... I haven’t liked any down south rappers since T.I. & Luda we’r… https://t.co/JnfgxFZhqm
Created At:Tue Mar 26 01:14:13 +0000 2019 None

Text: RT @MEDIOSYOPINION: EDITORIAL DE BABY CONTRA LOS Q CONVOCARON PARA EL 24 DE MARZO https://t.co/3Z3mH6Lsda

```
Created At:Tue Mar 26 01:13:59 +0000 2019 None

Enter search term: kjlkdlks
fetching
Found 0 tweets
```

**Important!**
You should cache all of the data from this exercise in a JSON file, and *submit the cache file along with your source code in your github repository.* So, for example, if you have already searched for tweets about "rock climbing", when we run your code, the code should use the CACHED data, and *should not* need to make a new request to the Twitter API. But if, for instance, you have never searched for "bicycles" before you submitted your final files, then if we enter "bicycles" when we run your code, it *should* make a request to the Twitter API.

**Grading Rubric:**
Because it is dependent on user input, there are only a few unit tests for this -- we will run your assignments with different inputs to grade them thoroughly!

**test_read_cache(10)**
- 5 points for reading the cache file contents into a JSON object.
- 5 points for returning the cached dictionary from the JSON object.

**test_get_tweet_data_with_caching(5)**
- 5 points for getting data from the dictionary (cache) using the search term

**These have to be tested live (not with unit tests)**
5 points for printing the results in the required format when there are at least 5 tweets.
5 points for handling less than 5 tweets correctly.
5 points for asking the user to enter a search phrase 3 times.
5 points for correctly calling **get_tweet_data_with_caching** with the user search term.
5 points for making a call to the Twitter API if the search term doesn't exist in the dictionary (cache).
10 points for saving the new Twitter data in the dictionary and in the cache file.
5 points for printing an appropriate message when an exception occurs during the search and returning None.
5 points for having at least three new search results in the submitted cache file.

**Extra Credit (3):**
Return a list of tuples, where each tuple contains the search term and the number of tweets for that search term in the dictionary from the cached file, in descending order from most tweets to least tweets.

For instance, when this function is run with the original 'cache_twitter.json' file it would return:

*[('Hope', 15), ('Thanksgiving', 14), ('Space', 12), ('horse', 11), ('Trump', 10), ('Halloween', 9), ('USMI', 8)]*

where “Hope” is the most tweeted term present in the given cached file

**Submission:**

***On Github - do not turn in your* twitter_info.py file!**

- JSON file containing your cached results - **cache_twitter.json**
- Source code - **Hw-Twitter.py**

***On Canvas***

- Submit URL of your Github repo