Searching for tweets

- Problem:
  You want to search Twitter for tweets using specific keywords and query constraints.
- Solution:
  Use the Search API to perform a custom query.

Searching for tweets

- Twitter's Search API returns results in batches, and we can configure the number of results per batch to a maximum value using the **count** keyword parameter.
- Generally the count is taken as 200. It is possible that more than 200 results (or the maximum value that you specify for count) may be available for any given query.
- In Twitter's API, we also have **cursor** to navigate to the next batch of results.

Searching for tweets

- Cursors are a new enhancement to Twitter's API and provide a more robust scheme than the pagination paradigm.
- The essence of the cursor paradigm is that it is able to better accommodate the dynamic and real-time nature of the Twitter platform.
- It could be the case that while you are navigating a batch of query results, relevant information becomes available that you would want to have included in your current results while you are navigating them, rather than needing to dispatch a new query.

```
def twitter_search(twitter_api, q, max_results=200, **kw):

    # See https://dev.twitter.com/docs/api/1.1/get/search/tweets and
    # https://dev.twitter.com/docs/using-search for details on advanced
    # search criteria that may be useful for keyword arguments

    # See https://dev.twitter.com/docs/api/1.1/get/search/tweets
    search_results = twitter_api.search.tweets(q=q, count=100, **kw)

    statuses = search_results['statuses']

    # Iterate through batches of results by following the cursor until we
    # reach the desired number of results, keeping in mind that OAuth users
    # can "only" make 180 search queries per 15-minute interval. See
    # https://dev.twitter.com/docs/rate-limiting/1.1/limits
    # for details. A reasonable number of results is ~1000, although
    # that number of results may not exist for all queries.

    # Enforce a reasonable limit
    max_results = min(1000, max_results)

    for _ in range(10): # 10*100 = 1000
        try:
            next_results = search_results['search_metadata']['next_results']
        except KeyError, e: # No more results when next_results doesn't exist
            break

        # Create a dictionary from next_results, which has the following form:
        # ?max_id=313519052523986943&q=NCAA&include_entities=1
        kwargs = dict([ kv.split('=')
                        for kv in next_results[1:].split("&") ])

        search_results = twitter_api.search.tweets(**kwargs)
        statuses += search_results['statuses']

        if len(statuses) > max_results:
            break

    return statuses

# Sample usage

twitter_api = oauth_login()

q = "CrossFit"
results = twitter_search(twitter_api, q, max_results=10)

# Show one sample search result by slicing the list...
print json.dumps(results[0], indent=1)
```