

CSCI5070 Advanced Topics in Social Computing

Python and APIs

Irwin King

The Chinese University of Hong Kong

king@cse.cuhk.edu.hk

©2012 All Rights Reserved.



Tutorial Overview

- Python basics and APIs
- R (basic and advanced, graphics, etc.)
- Web crawler
- PageRank, HITS, etc.
- NLTK and NetworkX
- Recommender systems
- Crowdsourcing/human computation
- Information extraction



Why Python?

- An agile programming language
- Scripting, interpreted, and high-level
- Fast prototyping and development
- Portable and cross-platform
- Object-oriented, easily extensible
- Powerful standard libs and extensive packages
- Stable and mature
- **FREE!**



Basic Requirements

```
YPCMC09006:~ iking$ python
Python 2.6.6 (r266:84374, Aug 31 2010, 11:00:51)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- Use Python 2.6 or 2.7 (using 2.6.6)
- idle, MacPython, IDE, etc.
- easy_install vs. pip
 - pip install --upgrade



Installing Packages

- pip install APackage
 - All packages are downloaded before installation.
 - Care is taken to present useful output on the console.
 - The reasons for actions are kept track of.
 - Useful error messages.
- The code is relatively concise and cohesive, making it easier to use programmatically.
- Packages don't have to be installed as egg archives, they can be installed flat (while keeping the egg metadata).
- Native support for other version control systems (Git, Mercurial and Bazaar)
- Uninstallation of packages.



Installing pip using getpip.py

```
YPCMC09006:Desktop iking$ python getpip.py
Checking for setuptools...
Downloading http://pypi.python.org/packages/2.6/s/setuptools/setuptools-0.6c11-py2.6.egg
Processing setuptools-0.6c11-py2.6.egg
Copying setuptools-0.6c11-py2.6.egg to /Library/Frameworks/Python.framework/Versions/
2.6/lib/python2.6/site-packages
Adding setuptools 0.6c11 to easy-install.pth file
Installing easy_install script to /Library/Frameworks/Python.framework/Versions/2.6/bin
Installing easy_install-2.6 script to /Library/Frameworks/Python.framework/Versions/2.6/
bin

Installed /Library/Frameworks/Python.framework/Versions/2.6/lib/python2.6/site-packages/
setuptools-0.6c11-py2.6.egg
Processing dependencies for setuptools==0.6c11
Finished processing dependencies for setuptools==0.6c11

Installing pip...
Searching for pip
Reading http://pypi.python.org/simple/pip/
Reading http://www.pip-installer.org
Reading http://pip.openplans.org
Best match: pip 1.0.2
Downloading http://pypi.python.org/packages/source/p/pip/
pip-1.0.2.tar.gz#md5=47ec6ff3f6d962696fe08d4c8264ad49
Processing pip-1.0.2.tar.gz
Running pip-1.0.2/setup.py -q bdist_egg --dist-dir /var/folders/pI/
pIYpFThQWoC5c6z2zZZ0Ik++DBs/-Tmp-/easy_install-oeoRqL/pip-1.0.2/egg-dist-tmp-x5aLQr
error: None
YPCMC09006:Desktop iking$
```



Output of easy_install pip

```
YPCMC09006:Desktop iking$ easy_install pip
Searching for pip
Reading http://pypi.python.org/simple/pip/
Reading http://www.pip-installer.org
Reading http://pip.openplans.org
Best match: pip 1.0.2
Downloading http://pypi.python.org/packages/source/p/pip/pip-1.0.2.tar.gz#md5=47ec6ff3f6d962696fe08d4c8264ad49
Processing pip-1.0.2.tar.gz
Running pip-1.0.2/setup.py -q bdist_egg --dist-dir /var/folders/pI/pIYpFThQWoC5c6z2zZZ0Ik++DBs/-Tmp-/easy_install-MvvaiM/pip-1.0.2/egg-dist-tmp-uAxq6v
warning: no files found matching '*.html' under directory 'docs'
warning: no previously-included files matching '*.txt' found under directory 'docs/_build'
no previously-included directories found matching 'docs/_build/_sources'
Adding pip 1.0.2 to easy-install.pth file
Installing pip script to /Library/Frameworks/Python.framework/Versions/2.6/bin
Installing pip-2.6 script to /Library/Frameworks/Python.framework/Versions/2.6/bin

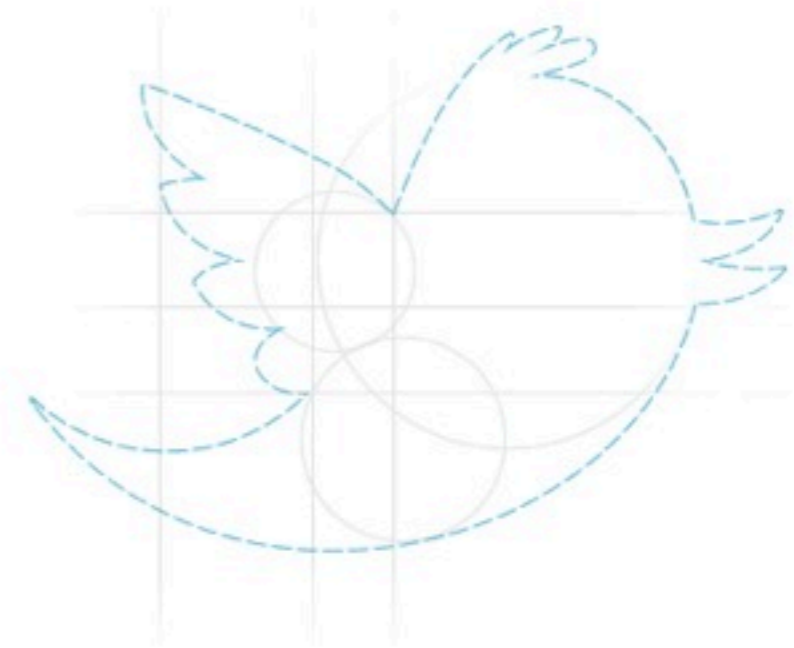
Installed /Library/Frameworks/Python.framework/Versions/2.6/lib/python2.6/site-packages/pip-1.0.2-py2.6.egg
Processing dependencies for pip
Finished processing dependencies for pip
```



<https://dev.twitter.com/>

Extend your reach.
Multiply your audience.

Add Twitter to your website



Recent posts from Twitter Developer Blog

Aug 25 [Changing the trends API](#)

Aug 19 [Bootstrap from Twitter](#)

Aug 15 [A Photo Upload API](#)

Aug 5 [Next steps with the t.co link wrapper](#)

Create applications that integrate Twitter

[Get started with the API](#)

Explore all of Twitter's API documentation

[Create an app](#)

Create an application to start using the Twitter API

[Discuss](#)

Get in touch with the API team and the community of developers



Twitter Libraries in Python

twitter developers

Search



API Status

Blog

Discussions

Documentation

Sign in

Python

- [Tweepy](#) by [@applepie](#) – a Twitter API library
- [Python OAuth2](#) by [Brian Rosner](#) – an OAuth library
- [Python Twitter](#) by [DeWitt Clinton](#) – a Python wrapper for the Twitter API (cf also [OAuth Python Twitter2](#))
- [Twitty Twister](#) by [Dustin Sallings](#) – a twisted client for Twitter API library
- [Twython](#) by [Ryan McGrath](#) – a Python wrapper for the Twitter API

Ruby

- [Grackle](#) by [Hayes Davis](#) – a lightweight Ruby wrapper for Twitter REST and Search APIs
- [Twitter](#) by [John Nunemaker](#) – a Ruby wrapper for Twitter REST and Search APIs
- [Twitter4R](#) – a community-supported Ruby wrapper for Twitter REST.
- [OAuth-Ruby](#) – an OAuth gem for clients and providers
- [Twitter_oauth](#) by [Richard Taylor](#) – an OAuth library for Twitter

Scala

- [DataBinder Dispatch](#) – an HTTP service library with OAuth

Inclusion in the **Twitter Libraries** is not an endorsement or recommendation of those organizations by Twitter. In addition, such inclusion is not intended to imply, directly or indirectly, that those organizations endorse or have any affiliation with Twitter.

[← Support](#)

[Getting Started](#)

[History of the REST & Search API →](#)

[Follow @twitterapi](#)

[API Terms](#) [API Status](#) [Blog](#) [Discussions](#) [Documentation](#) A Drupal community site supported by [Acquia](#)



```
>pip install twitter

import twitter

twitter_api = twitter.Twitter(domain="api.twitter.com",
api_version='1')
trends = twitter_api.trends()
[ trend['name'] for trend in trends['trends'] ]
twitter_search =
twitter.Twitter(domain="search.twitter.com")
search_results = []
for page in range(1,6):

search_results.append(twitter_search.search(q="barackobama"
, rpp=100, page=page))

import json
print json.dumps(search_results, sort_keys=True, indent=1)
```



```
tweets = []
tweets = [ r['text'] \
           for result in search_results \
           for r in result['results'] ]

words = []
for t in tweets:
    words += [ w for w in t.split() ]

len(words) # total words
len(set(words)) # unique words
1.0*len(set(words))/len(words) # lexical diversity
1.0*sum([ len(t.split()) for t in tweets ])/len(tweets) # avg
words per tweet
```



```
import cPickle
```

```
f = open("myData.pickle", "wb")  
cPickle.dump(words, f)  
f.close()
```

```
>pip install nltk
```

```
import cPickle  
import nltk
```

```
words = cPickle.load(open("myData.pickle"))  
freq_dist = nltk.FreqDist(words)  
freq_dist.keys()[:10] # 10 most frequent tokens  
freq_dist.keys()[-10:] # 10 least frequent tokens
```



```

import networkx as nx
import re

g = nx.DiGraph()
all_tweets = [ tweet
    for page in search_results
        for tweet in page["results"] ]

def get_rt_sources(tweet):
    rt_patterns = re.compile(r"(RT|via)((?:\b\W*@\w+)+)", re.IGNORECASE)
    return [ source.strip()
        for tuple in rt_patterns.findall(tweet)
            for source in tuple
                if source not in ("RT", "via") ]

for tweet in all_tweets:
    rt_sources = get_rt_sources(tweet["text"])
    if not rt_sources: continue
    for rt_source in rt_sources:
        g.add_edge(rt_source, tweet["from_user"], {"tweet_id" : tweet["id"]})

g.number_of_nodes()
g.number_of_edges()
g.edges(data=True)[0]
len(nx.connected_components(g.to_undirected()))
sorted(nx.degree(g))

```



```

> pip install pygraphviz

import pygraphviz

OUT = "search_results.dot"
try:
    nx.drawing.write_dot(g, OUT)
# except ImportError, e: # some problems here!

# Help for Windows users:
# Not a general-purpose method, but representative of
# the same output write_dot would provide for this graph
# if installed and easy to implement
dot = ['"%s" -> "%s" [tweet_id=%s]' % (n1, n2, g[n1][n2]
['tweet_id']) \
    for n1, n2 in g.edges()]
f = open(OUT, 'w')
f.write('strict digraph {\n%s\n}' % (';\n'.join(dot),))
f.close()

> circo -Tpng -Osnl_search_results snl_search_results.dot

```



```

> pip install pygraphviz

import pygraphviz

OUT = "search_results.dot"
try:
    nx.drawing.write_dot(g, OUT)
# except ImportError, e: # some problems here!

# Help for Windows users:
# Not a general-purpose method, but representative of
# the same output write_dot would provide for this graph
# if installed and easy to implement
dot = ['"%s" -> "%s" [tweet_id=%s]' % (n1, n2, g[n1][n2]
['tweet_id']) \
    for n1, n2 in g.edges()]
f = open(OUT, 'w')
f.write('strict digraph {\n%s\n}' % (';\n'.join(dot),))
f.close()

> circo -Tpng -Osnl_search_results snl_search_results.dot

```

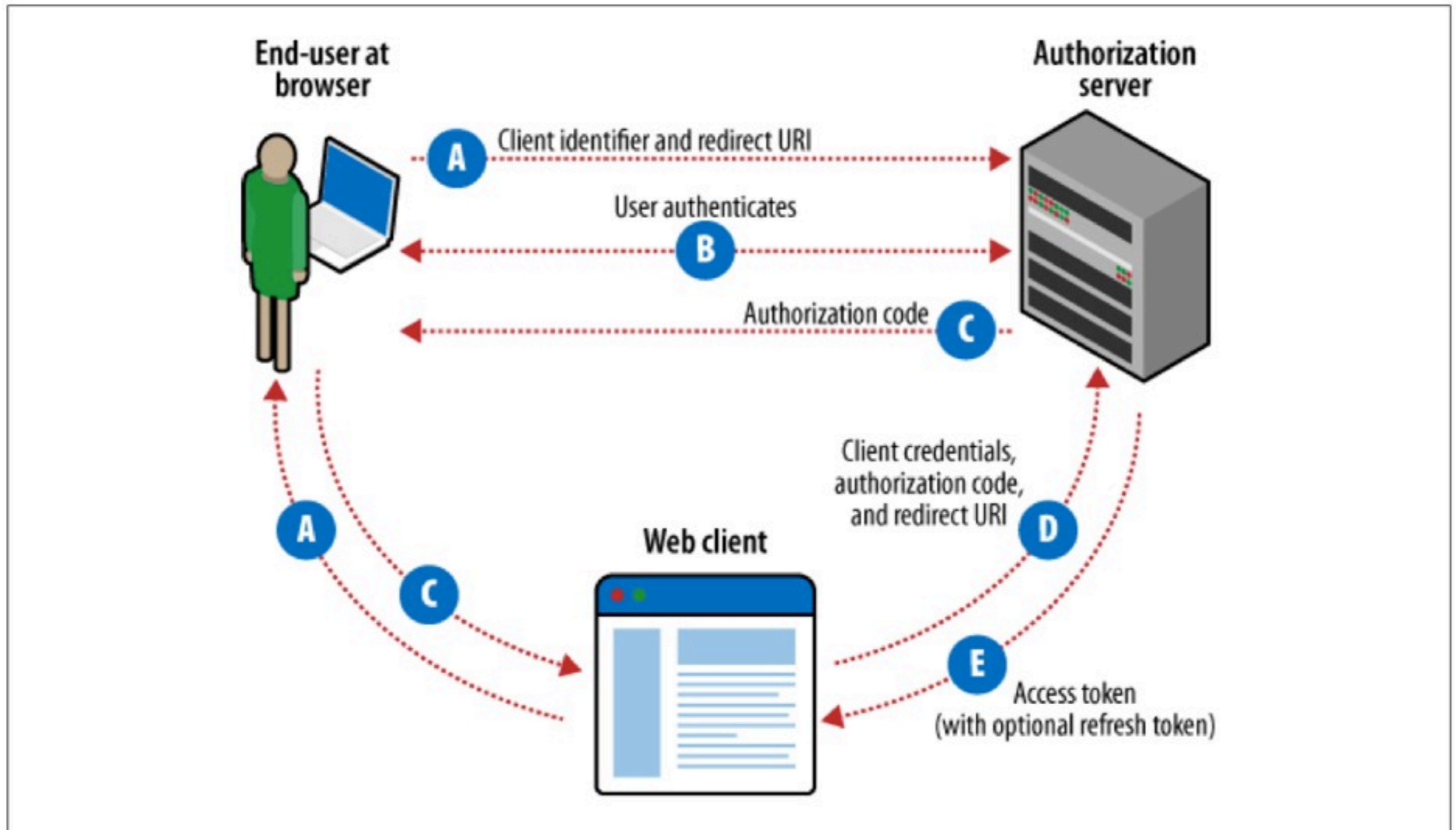


```
import twitter
import json

screen_name = 'iMacKing'
t = twitter.Twitter(domain='api.twitter.com', api_version='1')
response = t.users.show(screen_name=screen_name)
print json.dumps(response, sort_keys=True, indent=4)
```



OAuth 2.0



<https://dev.twitter.com/apps/new>

Home

Sign in with your Twitter account

Username: *

New to Twitter? [Sign up](#) @P

Password: *

Log in

Home → My applications

Create an application

Application Details

Name: *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens.

Description: *

Your application description, which will be shown in user-facing authorization screens.

WebSite: *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL:

Where should we return after successfully authenticating? For @Anywhere applications, only the domain specified in the callback will be used. OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.



Share your phone, IM and more with your connections. Update it now »

Filter Connections Select: All, None abc

All Connections (84)

- Tags: friends (15), colleagues (6), partners (4), classmates, group members, untagged (59)
- Companies
- Locations
- Industries
- Recent Activity


Name	Profile Picture	Job Title	Company	Connections
Lystad, Erik	[Profile Picture]	Medical Doctor - Lancaster General Hospital	Lancaster General Hospital	148
M				
Mak, Guy	[Profile Picture]	Senior Information Security Risk Officer - HSBC	HSBC	98
Melli, Gabor	[Profile Picture]	Sponsorship co-Chair - ICDM-2011	ICDM-2011	436
Metzler, Don	[Profile Picture]	Research Assistant Professor - University of Southern California	University of Southern California	348
Miller, Tobin	[Profile Picture]	teaching pastor - Watermark Community Church in Hong Kong	Watermark Community Church in Hong Kong	79
N				
Najork, Marc	[Profile Picture]	Principal Researcher - Microsoft	Microsoft	257
Ngan, Alice	[Profile Picture]	Chairman - LES China - HK Subchapter	LES China - HK Subchapter	68

Quickly view and organize your connections? Select a category or individual to see contact info, send a message and more.

0 outstanding sent invitations | Export connections

AdChoices

REWARD YOUR SUCCESS



GET A DECISION IN 60 SECONDS

APPLY NOW



```
YPCMC09006:Tutorial-01 iking$ python linkedin__analyze_companies.py  
mylinkedin.csv
```

Company	Freq
Microsoft	5
The Chinese University of Hong Kong	3
Cisco Systems	2

```
YPCMC09006:Tutorial-01 iking$ python linkedin__analyze_titles.py  
mylinkedin.csv
```

Title	Freq
Associate Professor	5
Assistant Professor	2
Executive Director	2
Managing Director	2
PhD Candidate	2
Professor	2
Systems Engineer	2



<https://developer.linkedin.com/>

LinkedIn Developers ▼ Irwin King

Home Plugins APIs Apply with LinkedIn Showcase Discuss

New Plugin!
Apply with LinkedIn
Now it's easy for candidates to apply to your jobs using their LinkedIn Profile
[Get Started](#)

Share Button
153
in Share
Help users share your website with LinkedIn members and drive traffic back to your site.
[Get it](#)

Member Profile
Adam Nash
VP Product Management, LinkedIn
San Francisco, Bay Area
View Profile
How you're connected to Adam
Bring LinkedIn member profiles to your site to help users discover common professional connections.
[Get it](#)

Company Insider
LinkedIn: 1,045 employees in your network
Following
Jonas Laiten, Enterprise Account Executive, LinkedIn
Kiran Prasad, Director of Engineering, LinkedIn
Greg Arnold, Engineering Director, LinkedIn
Brian Guarnaci, Process Engineer, LinkedIn
Enhance your content and show rich personalized insights about companies featured on your site.
[Get it](#)



<http://developers.facebook.com/>

The screenshot shows the Facebook Developers website homepage. At the top, there is a navigation bar with the text "facebook DEVELOPERS" and links for "Documentation", "Support", "Blog", and "Apps". A search bar on the right contains the text "Search for documentation".

The main content area features a large banner for "Facebook's f8 Developer Conference" in San Francisco, CA, scheduled for September 22, 2011. The banner includes a green "Register" button and a link to "learn more about f8". Below the banner are three circular icons representing different development areas: "Build for Websites", "Build for Mobile", and "Build Apps on Facebook".

The "Build for Websites" section describes driving growth and engagement through Facebook Login and Social Plugins. The "Build for Mobile" section describes letting users find and connect to their friends in mobile apps and games. The "Build Apps on Facebook" section describes integrating with the core experience by building apps that operate within Facebook.

Below the main content are three columns: "Latest Updates" with a list of recent news items, "Learn Social Design" with icons and a brief explanation of social design, and "Showcase" featuring logos of various companies like SportingNews, Flipboard, Eventbrite, and Airbnb.

At the bottom, there is a "Like" button, a "Send" button, and a notification that "87,685 people like this. Be the first of your friends." The footer contains the text "Facebook © 2011" and links for "About", "Platform Policies", and "Privacy Policy".



A Geocoding Toolbox

```
>pip install geopy
```

```
from geopy import geocoders
g = geocoders.Google()
place, (lat, lng) = g.geocode("1 Infinite Loop in Cupertino")
print "%s: %.5f, %.5f" % (place, lat, lng)
```

```
g = geocoders.Google(domain='maps.google.co.uk')
```

```
gn = geocoders.GeoNames()
place, (lat, lng) = gn.geocode("Cleveland, OH 44106")
print "%s: %.5f, %.5f" % (place, lat, lng)
```

```
from geopy import distance
_, be = g.geocode('Berkeley, CA')
_, bo = g.geocode('Boston, MA')
_, cu = g.geocode('Cupertino, CA')
distance.distance(be, bo).miles
distance.distance(be, cu).miles
```



Matplotlib

```
import matplotlib.pyplot as plt
import numpy as np
```

```
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```

```
plt.plot([1,2,3,4], [1,4,9,16])
plt.show()
```

```
plt.plot([1,2,3,4], [1,4,9,16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```

```
# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)
# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
plt.show()
```




```
def f(t):  
    return np.exp(-t) * np.cos(2*np.pi*t)  
  
t1 = np.arange(0.0, 5.0, 0.1)  
t2 = np.arange(0.0, 5.0, 0.02)  
  
plt.figure(1)  
plt.subplot(211)  
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')  
  
plt.subplot(212)  
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')  
plt.show()
```



```
# Working with text

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 50, normed=1, facecolor='g',
alpha=0.75)

plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```



```
# Annotating text

import numpy as np
import matplotlib.pyplot as plt

ax = plt.subplot(111)

t = np.arange(0.0, 5.0, 0.01)
s = np.cos(2*np.pi*t)
line, = plt.plot(t, s, lw=2)

plt.annotate('local max', xy=(2, 1), xytext=(3, 1.5),
            arrowprops=dict(facecolor='black', shrink=0.05), )
plt.ylim(-2,2)
plt.show()
```



Working Under Mac OS X Lion

Switching python version to 2.6 by default:

- > % defaults write com.apple.versioner.python Version 2.6

Installing nltk:

- > sudo pip-2.6 install pyyaml
- > sudo pip-2.6 install <http://pypi.python.org/packages/source/n/nltk/nltk-2.0.1rc1.tar.gz>

Installing pygraphviz:

- > /usr/bin/ruby -e "\$(curl -fsSL <https://raw.githubusercontent.com/gist/323731>)"
- > brew install graphviz
- > sudo pip-2.6 install pygraphviz

Install matplotlib:

- > git clone <https://github.com/matplotlib/matplotlib.git>
- > cd matplotlib
- > python2.6 setup.py build
- > sudo python2.6 setup.py install
- > sudo pip-2.6 install numpy



References

- M.A. Russell, Mining the Social Web: Analyzing Data from Facebook, Twitter, LinkedIn, and Other Social Media Sites, 1st ed. O'Reilly Media, 2011.
- <http://www.acm.uiuc.edu/sigunix/workshops/crashpython/crashpython.pdf>
- <http://dubroy.com/blog/so-you-want-to-install-a-python-package/>
- <https://dev.twitter.com/docs/twitter-libraries#python>

