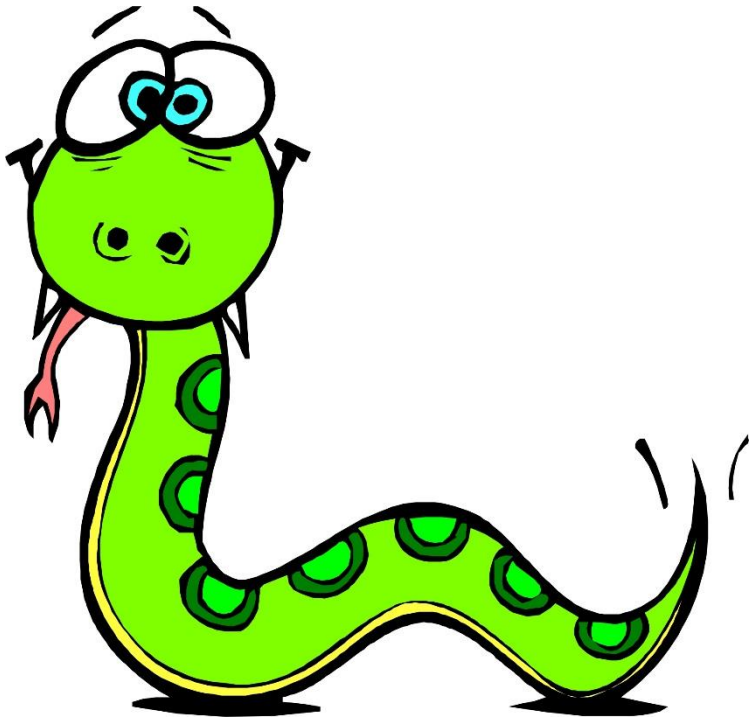
Python Twitter API

Session 2, Oct 25th, 2014



Twitter API – Getting Started

Twitter API Project

- Twitter authentication
- Download the Twitter API Project and extract all files
- Enter your API information
- Test the Twitter API code with twitter1.py

Mentor Check Off

- Run twitter1.py

- Status Indicator
  - Red not checked
  - Green for done!

Twitter API

- Tweepy (twee – pie)
- Documentation
- Remember to save your file before trying to run it!

Twitter API

- api.home_timeline

- tweet.text
- tweet.user

- user.screen_name
- user.followers_count

twitter1.py

```
public_tweets = api.home_timeline(count=100)
for tweet in public_tweets:
    print tweet.text
```

- What does it do?
- What more can we do with this?

twitter2.py

- Grabs the usernames of your followers

```
print user.screen_name + " has " +
str(user.followers_count) + " followers."


print "They include these 100 people: "


for friend in user.friends(count=100):
    print friend.screen_name
```

twitter3.py

```
public_tweets = api.search("data science", count=20)

for tweet in public_tweets:
    print (tweet.user.screen_name + "\t" +
      str(tweet.created_at) + "\t" + tweet.text)
```

- Note that a lot of names are intuitive, this increases readability
  - Very helpful for large projects with many contributors

twitter4.py

```
output_file = codecs.open("MY_DATA.tsv", "w", "utf-8")

public_tweets = api.search("data science", count=10)

for tweet in public_tweets:
    print >>output_file, tweet.user.screen_name + "\t" +
                str(tweet.created_at) + "\t" + tweet.text
```

- What would a line in the output file look like?

twitter-stream1.py, twitter-stream2.py

```
class StreamListener(tweepy.StreamListener):
    def on_status(self, tweet):
        print (tweet.user.screen_name + "\t" +
        tweet.text.encode('ascii', 'ignore'))

     def on_error(self, status_code):
        print 'Error: ' + repr(status_code)
        return Falsel = StreamListener()


streamer = tweepy.Stream(auth=auth, listener=l)
streamer.sample()


keywords = ['python', 'perl']
streamer.filter(track = keywords)
```

- Grabs tweets in real time!

Next steps

- twitterfun.py

- Altered from twitter2.py to see if a user is “popular”

Next steps

- Play and have fun!
- Decide what you want to do with the information you collect
  - Sort information, filter with keywords/hashtags/users, pattern recognition, basically whatever you can think of
- Ask yourself what's required of the code to do what you want
  - Recall the skills you gained from previous sessions: variables, If, elif, and else statements, loops
- Manipulate existing code or write your own.
  - Refer to documentation, your peers, and your mentors
- Consider yourself a coder!