Introduction to Using APIs with Python

NaLette Brodnax

Indiana University Bloomington

School of Public & Environmental Affairs

Department of Political Science

Center of Excellence for Women in Technology

CEWiT

CENTER OF EXCELLENCE
FOR WOMEN IN TECHNOLOGY

We started a conversation about women in technology, ensuring all women have a seat at the table in every technology venture.

CEWiT addresses the global need to increase participation of women at all stages of their involvement in technology related fields.

Faculty, staff, alumnae and student alliances hold events, host professional seminars, and give IU women opportunities to build a community.

CONNECT WITH US

iucewit @IU_CEWiT iucewit

cewit.indiana.edu

what is an API?

An application programming interface (api) is a tool that allows computers to exchange data.

1

static webpage

2

dynamic webpage

3

application programming interface (api)

uses of APIs

- Social – Twitter, Facebook, etc.
- Internet – bit.ly, domain registration
- Mapping – Google Maps, Bing Maps, etc.
- Search – Google, Yahoo, etc.

APIs make information transferred across the web digestible for a computer.

key protocols

- HTTP – communicating with web server
- OAuth – accessing secure information

part 2: python overview

getting the tools

- Use Python on SSRC computer
  - Search for IDLE

- Use Python on your laptop
  - Install requests package using the command line

Sample installation code for Mac OS X using Terminal:

```
$ pip3 install requests_oauthlib
```

integrated development environment

```
Python 3.4.2
>>>
>>> print(“Hello, world.”)
Hello world.
>>>
```

❶

```
#My first script

print(“Hello, world.”)
```

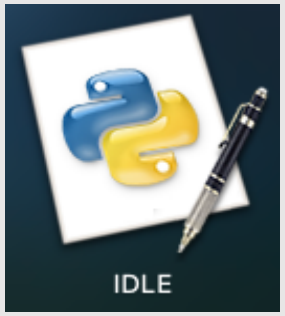
❷

```
Last login: Tue Nov 1 13:05:22
149-160-200-169:~ nbrodnax$
```

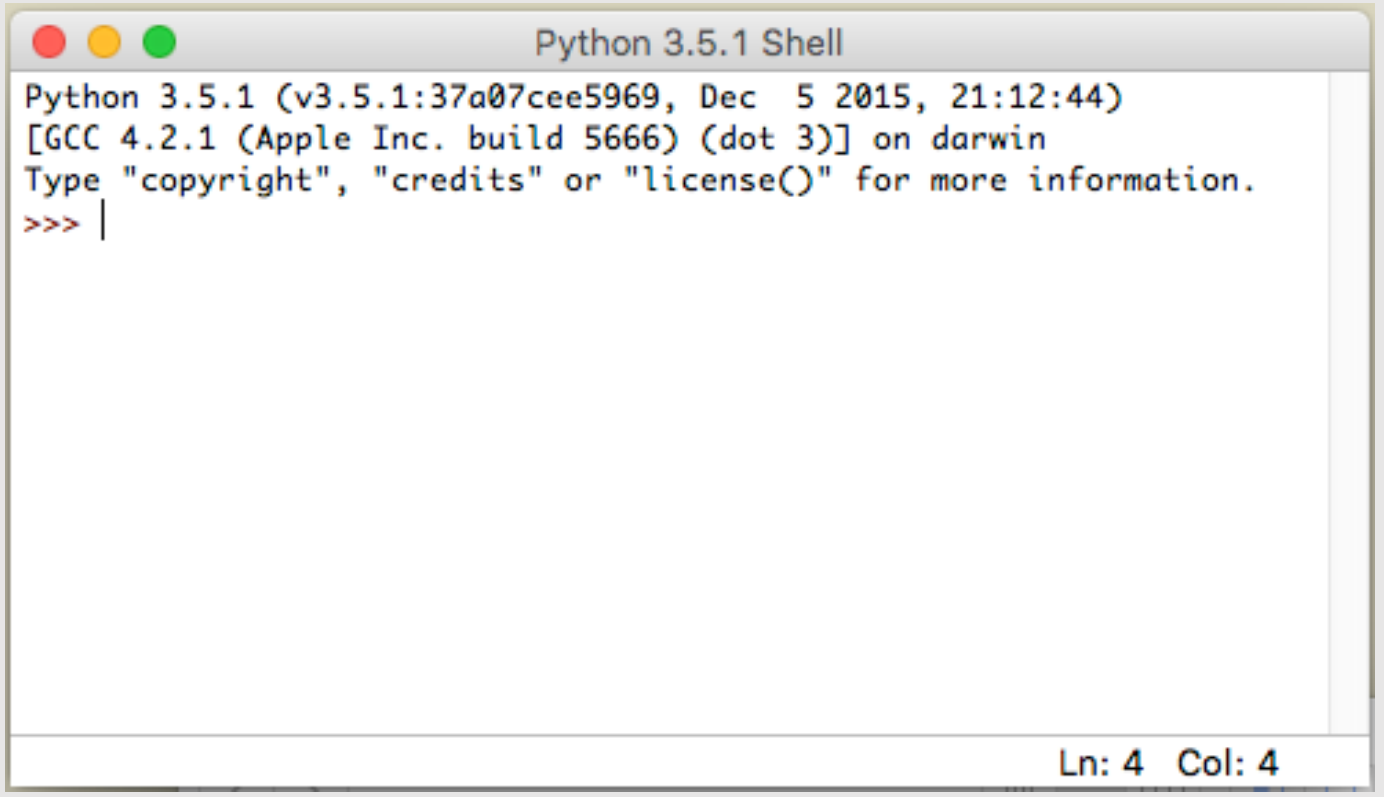
❸

1. Python interpreter (required)
2. Text editor (optional)
3. Command line (optional)

python development environment



IDE → **IDLE**



Interact with Python

Write programs in a separate screen: **File → New File**
Run the program in the interpreter: **F5** or **Run → Run Module**

data types: sequences

String—ordered sequence of characters

```
‘happy’
```

List—ordered sequence of items

```
[‘Leia’, ‘Rey’, ‘Maz’]
```

Dictionary—unordered sequence of key-value pairs

```
{‘name’:‘Kylo’, ‘side’:‘dark’}
```

part 3: data collection!

the API we will access

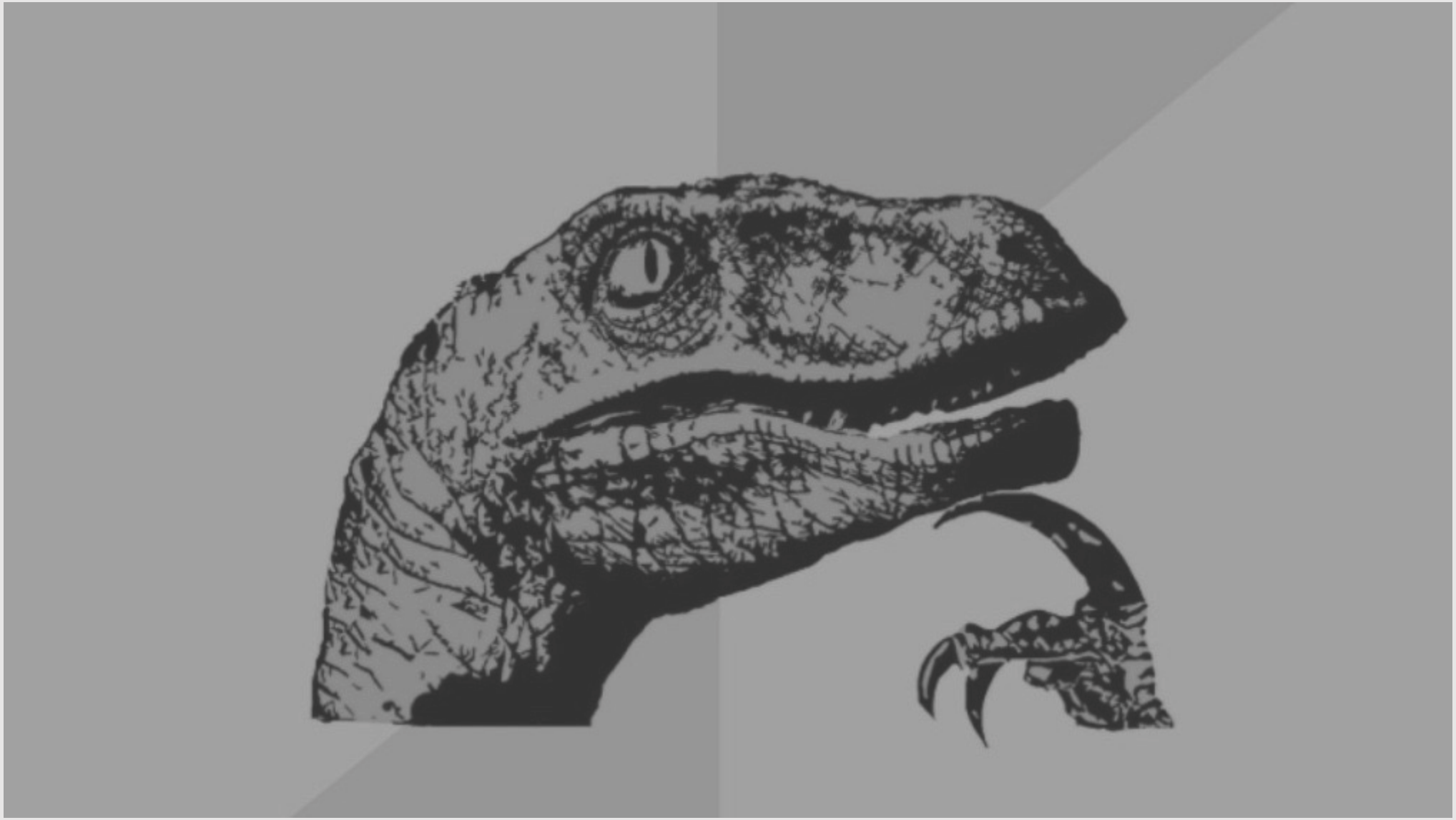
Twitter REST API

http://dev.twitter.com

Workshop Code

https://github.com/nmbrodnax/wim-workshop

- twitter_api.py
- twitter_auth_example.txt



Review • **A**ccess • **P**arse • **T**ransform • st**OR**e

RAPTOR

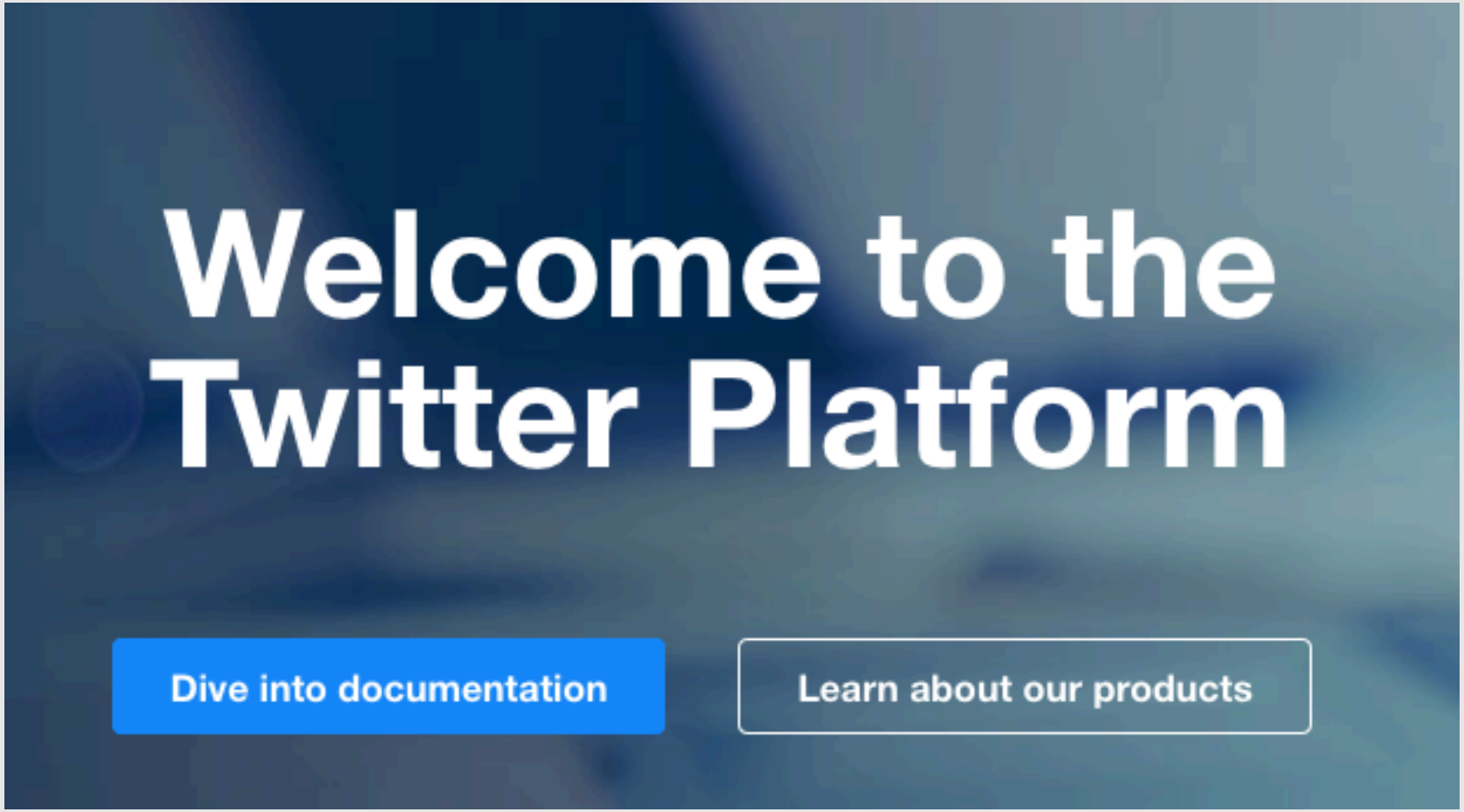
	Web Server	**Web Server + API**
Review	HTML structure (tags, attributes, etc.)	Parameters and structure from documentation
Access	No registration, no authentication	Registration and sometimes authentication
Parse	HTML	JSON or XML
Transform	Nested tables, lists	Nested dictionary
St**OR**e	Text, CSV	Text, CSV

next steps

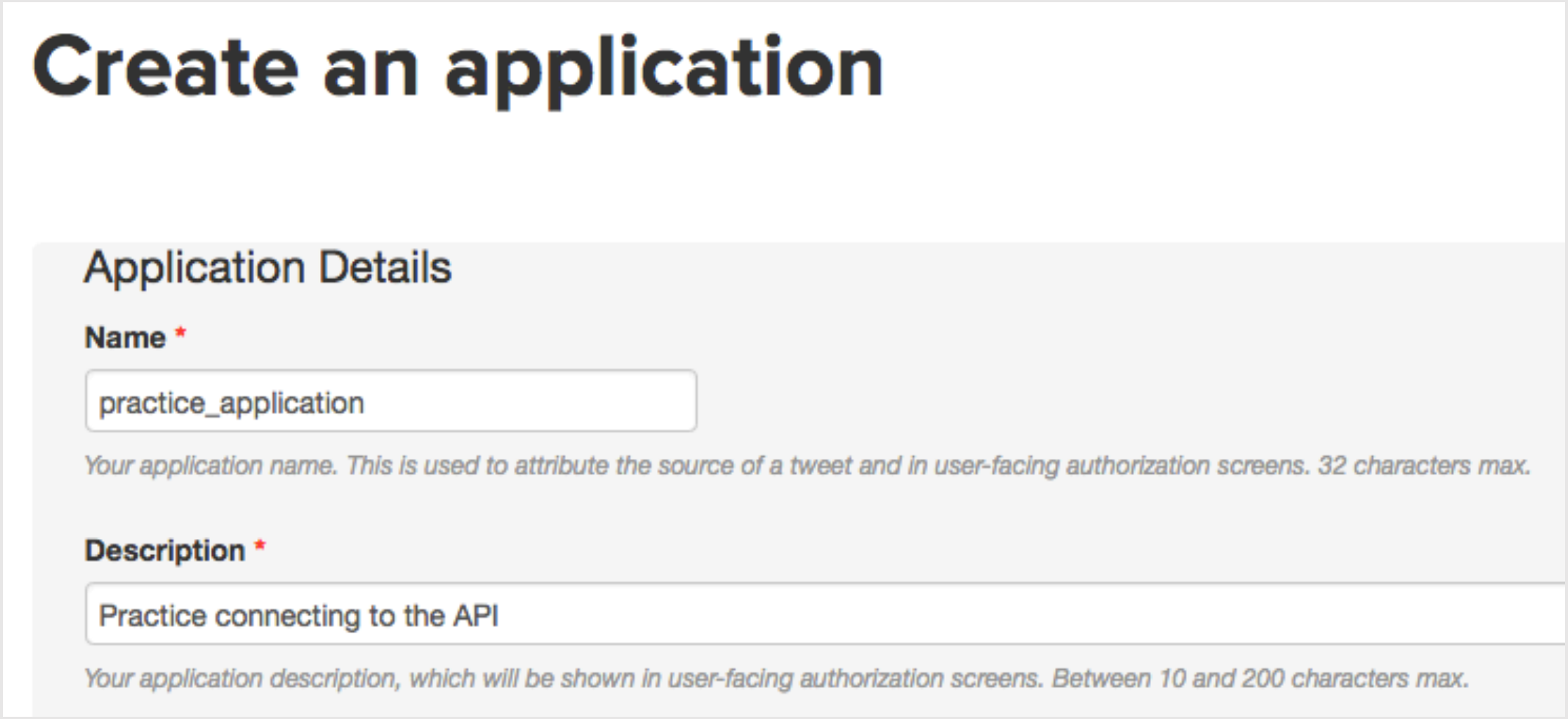
- Register as a developer
- Create an application
- Create an authentication document
- Use the API

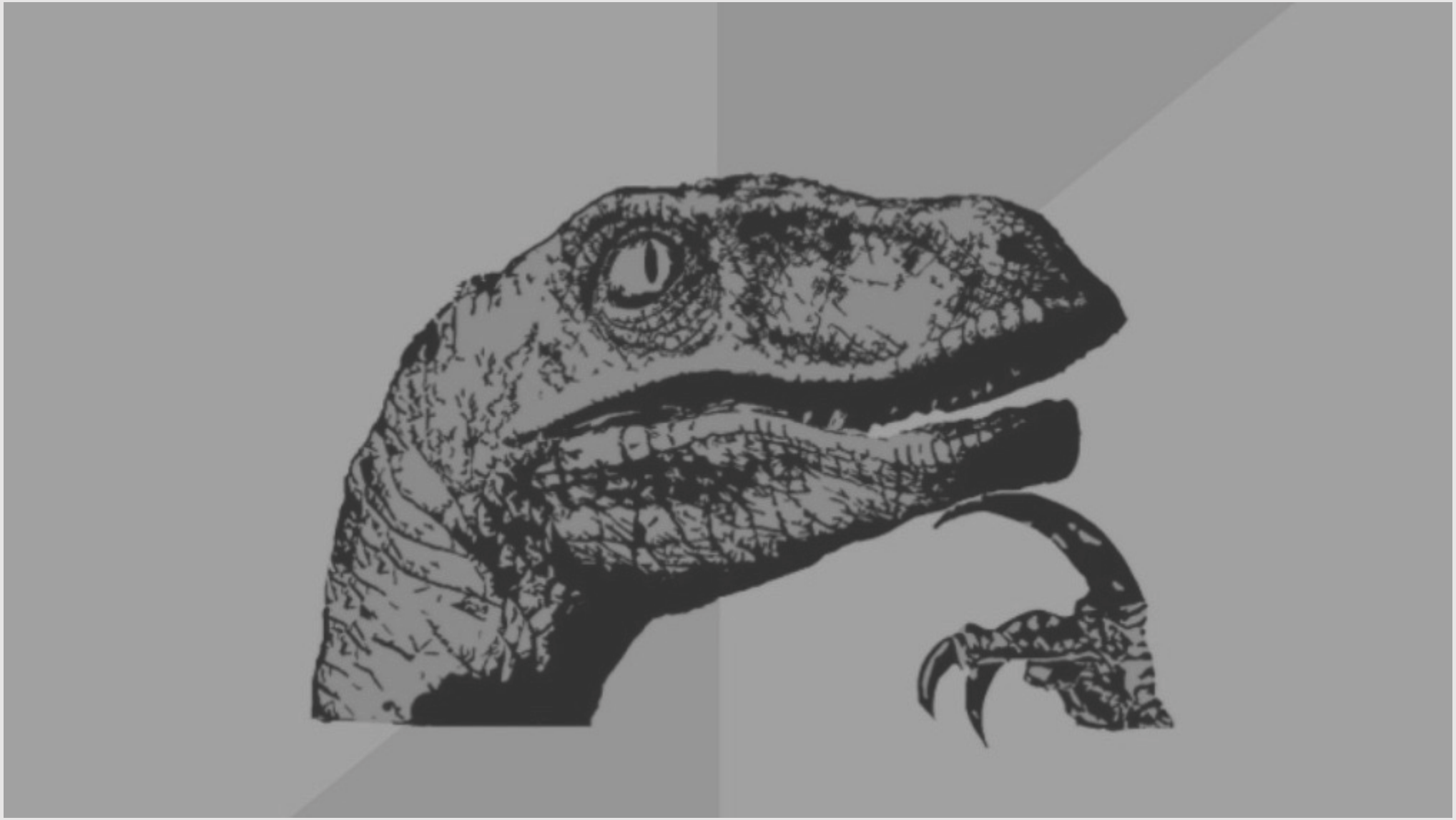
registration

https://dev.twitter.com/#



your application



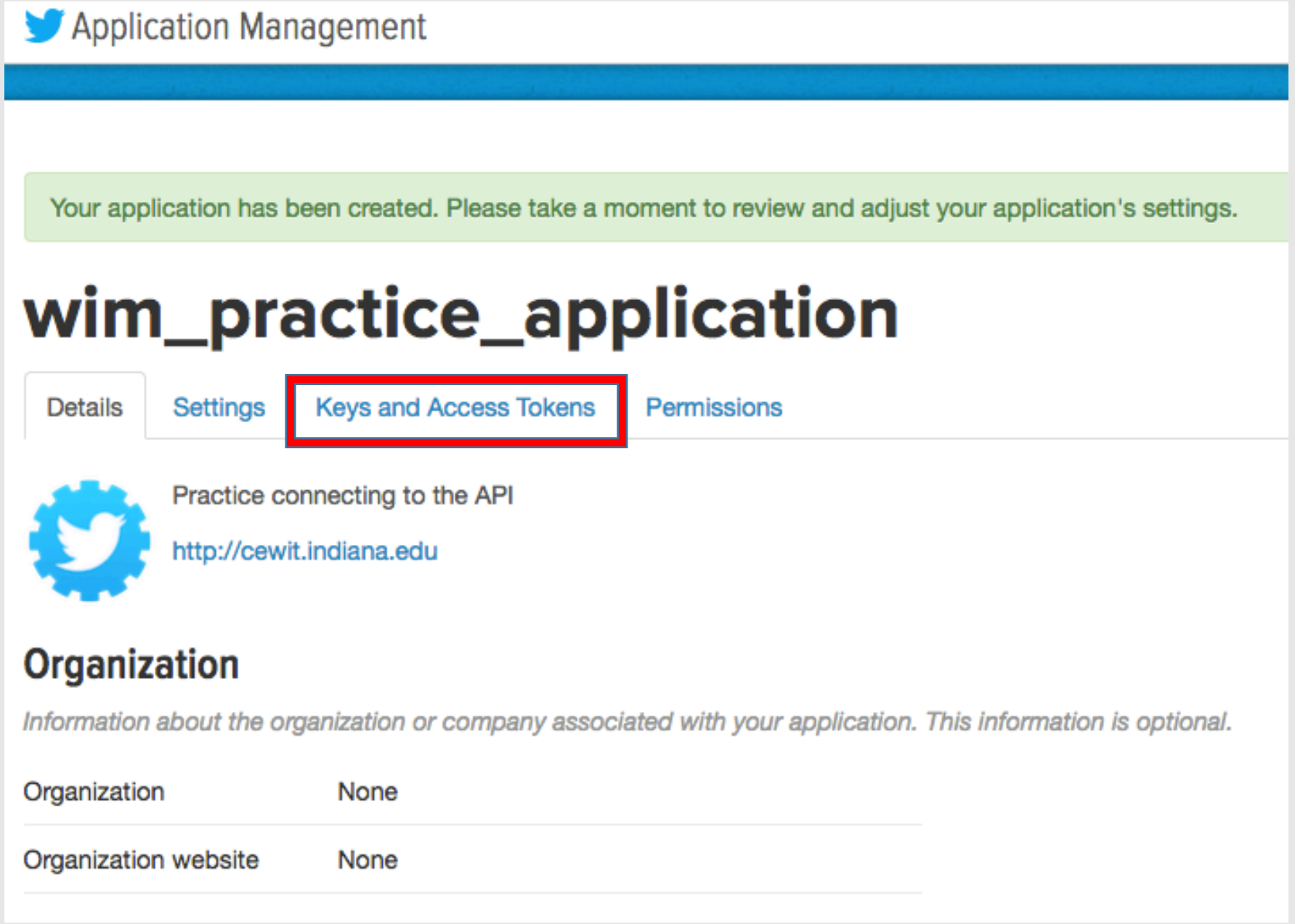


Review • **A**ccess • **P**arse • **T**ransform • st**OR**e

review

https://dev.twitter.com/rest/public/search

access


Application Management
Your application has been created. Please take a moment to review and adjust your application's settings.
wim_practice_application
Details
Settings
Keys and Access Tokens
Permissions
Practice connecting to the API
http://cewit.indiana.edu
Organization
Information about the organization or company associated with your application. This information is optional.
Organization
None
Organization website
None

access

Your Twitter developer credentials

```
{'consumer_key': 'your_consumer_key',
'consumer_secret': 'your_consumer_secret',
'access_token': 'your_access_token',
'access_secret': 'your_access_secret'}
```

let’s take a 10-minute break!

access

Import statements allow you to add functions

```
import sys
import csv
import from requests_oauthlib import OAuth1Session
```

access

```
# get authentication parameters from local file
local_file = ‘your_file_path’
with open(local_file) as txtfile:
     contents = txtfile.readline()
     credentials = eval(contents.strip('\n'))
```

access

```
# api OAuth 1.0 authentication
twitter = OAuth1Session(
    credentials.get('consumer_key'),
    client_secret=credentials.get('consumer_secret'),
    resource_owner_key=credentials.get('access_token'),
    resource_owner_secret=credentials.get('access_secret')
)
```

access

```
# host location of api
host = 'https://api.twitter.com’

# api GET request for user ids of followers
get_path = '/1.1/search/tweets.json?q=%40IUBloomington’
url = host + get_path
response = twitter.get(url)
```

parse

```
# check the HTTP response code
print(response)
# parse the JSON data into a python object
tweets = response.json()
```

transform

```
# check the structure of the data
print(len(tweets))
print(type(tweets))
print(tweets.keys())
print(len(tweets['statuses']))

# encode uncommon characters
non_bmp_map = dict.fromkeys(range(0x10000, sys.maxunicode + 1),
                            0xfffd)
print(str(tweets['statuses']).translate(non_bmp_map))
```

store

- text file
- CSV file
- other formats

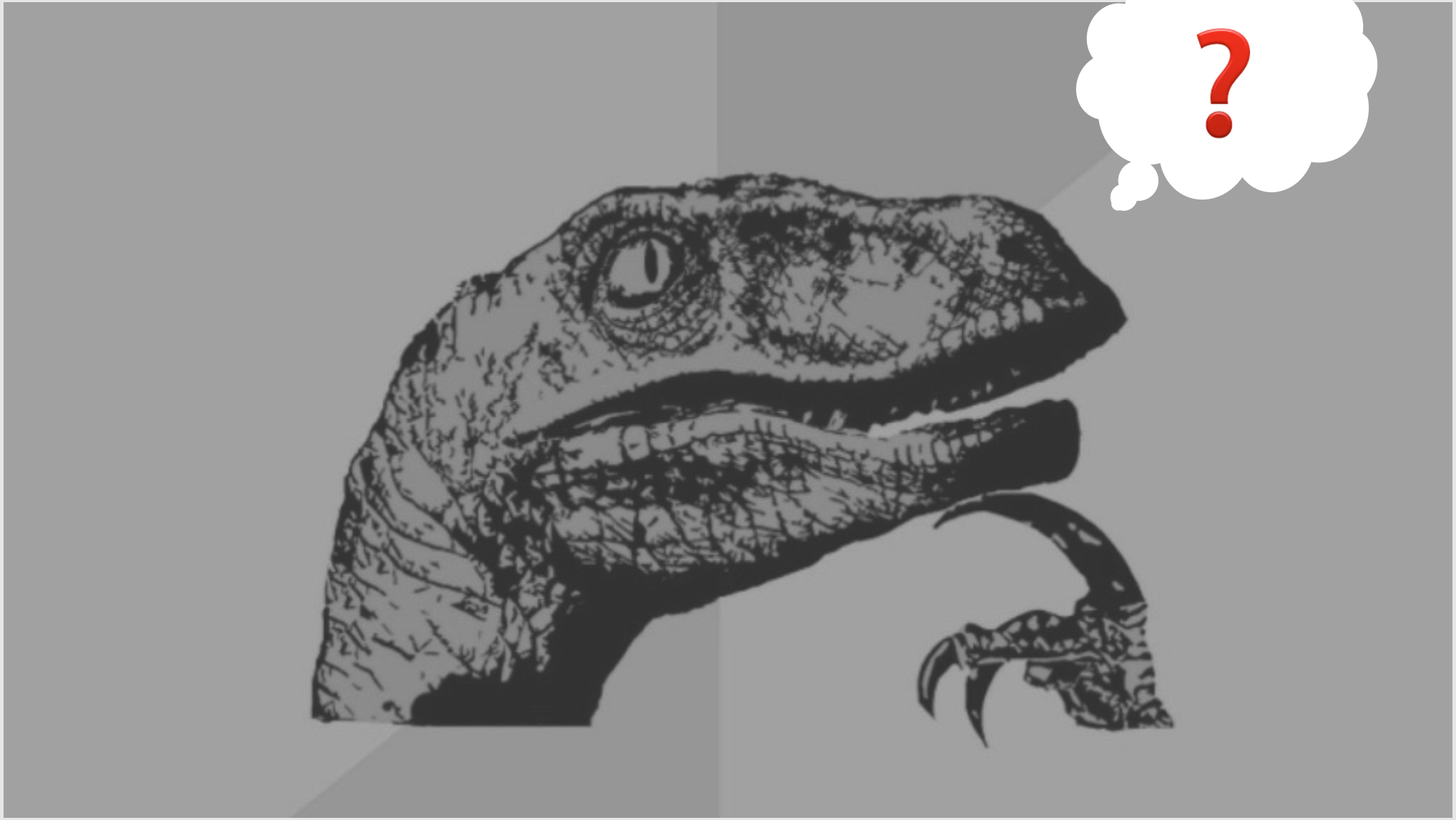
run your script!

From IDLE:
Run the program in the interpreter: **F5** or **Run → Run Module**

OR

From the Command Line:

```
$ python3 twitter_api.py
```



Review • **A**ccess • **P**arse • **T**ransform • st**OR**e

Thank you!

email: nbrodnax@indiana.edu

linkedin: nalettebrodnax
github: nmbrodnax
twitter: @nbrodnax