**--- xml1.py**

import xml.etree.ElementTree as ET

data = '''

<person>

<name>Chuck</name>

<phone type="intl">

+1 734 303 4456

</phone>

<email hide="yes"/>

</person>'''

tree = ET.fromstring(data)

print 'Name:',tree.find('name').text

print 'Attr:',tree.find('email').get('hide')

**--- xml2.py**

import xml.etree.ElementTree as ET

input = '''

<stuff>

<users>

<user x="2">

<id>001</id>

<name>Chuck</name>

</user>

<user x="7">

<id>009</id>

<name>Brent</name>

</user>

</users>

</stuff>'''

stuff = ET.fromstring(input)

lst = stuff.findall('users/user')

print 'User count:', len(lst)

for item in lst:

print 'Name', item.find('name').text

print 'Id', item.find('id').text

print 'Attribute', item.get("x")

**--- json1.py**

import json

data = '''

{

"name" : "Chuck",

"phone" : {

"type" : "intl",

"number" : "+1 734 303 4456"

},

"email" : {

"hide" : "yes"

}

}'''

info = json.loads(data)

print 'Name:',info["name"]

print 'Hide:',info["email"]["hide"]

**--- json2.py**

import json

input = '''

[

{ "id" : "001",

"x" : "2",

"name" : "Chuck"

} ,

{ "id" : "009",

"x" : "7",

"name" : "Chuck"

}

]'''

info = json.loads(input)

print 'User count:', len(info)

for item in info:

print 'Name', item['name']

print 'Id', item['id']

print 'Attribute', item['x']

**--- geojson.py**

import urllib

import json

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'

while True:

address = raw\_input('Enter location: ')

if len(address) < 1 : break

url = serviceurl + urllib.urlencode({'sensor':'false',

'address': address})

print 'Retrieving', url

uh = urllib.urlopen(url)

data = uh.read()

print 'Retrieved',len(data),'characters'

try: js = json.loads(str(data))

except: js = None

if 'status' not in js or js['status'] != 'OK':

print '==== Failure To Retrieve ===='

print data

continue

print json.dumps(js, indent=4)

lat = js["results"][0]["geometry"]["location"]["lat"]

lng = js["results"][0]["geometry"]["location"]["lng"]

print 'lat',lat,'lng',lng

location = js['results'][0]['formatted\_address']

print location

**--- Program output**

Enter location: Ann Arbor, MI

Retrieving http://maps.googleapis.com/...

Retrieved 1669 characters

lat 42.2808256 lng -83.7430378

Ann Arbor, MI, USA

Enter location:

**--- hidden.py**

def oauth() :

return { "consumer\_key" : "h7Lu...Ng",

"consumer\_secret" : "dNKenAC3New...mmn7Q",

"token\_key" : "10185562-ein2...P4GEQQOSGI",

"token\_secret" : "H0ycCFemmwyf1...qoIpBo" }

**--- Augmented URL**

https://api.twitter.com/1.1/statuses

/user\_timeline.json?count=2

&oauth\_version=1.0&oauth\_token=101...SGI

&screen\_name=drchuck&oauth\_nonce=09239679

&oauth\_timestamp=1380395644

&oauth\_signature=rLK...BoD

&oauth\_consumer\_key=h7Lu...GNg

&oauth\_signature\_method=HMAC-SHA1

**--- twitter2.py**

import urllib

import twurl

import json

TWITTER\_URL = 'https://api.twitter.com/1.1/friends/list.json'

while True:

print ''

acct = raw\_input('Enter Twitter Account:')

if ( len(acct) < 1 ) : break

url = twurl.augment(TWITTER\_URL,

{'screen\_name': acct, 'count': '5'} )

print 'Retrieving', url

connection = urllib.urlopen(url)

data = connection.read()

headers = connection.info().dict

print 'Remaining', headers['x-rate-limit-remaining']

js = json.loads(data)

print json.dumps(js, indent=4)

for u in js['users'] :

print u['screen\_name']

s = u['status']['text']

print ' ',s[:50]

**--- twitter2.py output**

Enter Twitter Account:drchuck

Retrieving https://api.twitter.com/1.1/friends ...

Remaining 14

{

"users": [

{

"status": {

"text": "@jazzychad I just bought one .\_\_.",

"created\_at": "Fri Sep 20 08:36:34 +0000 2013",

},

"location": "San Francisco, California",

"screen\_name": "leahculver",

"name": "Leah Culver",

},

{

"status": {

"text": "RT @WSJ: Big employers like Google ...",

"created\_at": "Sat Sep 28 19:36:37 +0000 2013",

},

"location": "Victoria Canada",

"screen\_name": "\_valeriei",

"name": "Valerie Irvine",

],

}

leahculver

@jazzychad I just bought one .\_\_.

\_valeriei

RT @WSJ: Big employers like Google, AT&amp;T are h

ericbollens

RT @lukew: sneak peek: my LONG take on the good &a

halherzog

Learning Objects is 10. We had a cake with the LO,