# #Reading data in core python

# #load data using pandas DF

# #REST API

# #Querying Data Using Connector/Python

# #HTML document scraping using Beautiful soup

# #python API for face book .

# #Python API for twitter

#Reading data in core python

# 'rU' mode (read universal) converts different line endings into '\n'

f = open('airlines.csv', mode='rU')

file\_string = f.read()

f.close()

# use a context manager to automatically close your file

with open('airlines.csv', mode='rU') as f:

file\_string = f.read()

# read the file into a list (each list element is one row)

with open('airlines.csv', mode='rU') as f:

file\_list = []

for row in f:

file\_list.append(row)

# do the same thing using a list comprehension

with open('airlines.csv', mode='rU') as f:

file\_list = [row for row in f]

# read the data from yelp.json into a list of rows

# each row is decoded into a dictionary using using json.loads()

import json

# Download the file and save it in python working directory <https://raw.githubusercontent.com/TVMKISHORE/Analytics/master/data/yelp.json>

#we can load directly with the Url, but it dint work for some reason.

with open(“yelp.json.txt”, 'rU') as k:

data = [json.loads(row) for row in k]

yelp=pd.DataFrame(data)

# side note: splitting strings

'hello DAT students'.split()

'hello DAT students'.split('e')

**# split each string (at the commas) into a list**

with open('airlines.csv', mode='rU') as f:

file\_nested\_list = [row.split(',') for row in f]

# do the same thing using the csv module

import csv

with open('airlines.csv', mode='rU') as f:

file\_nested\_list = [row for row in csv.reader(f)]

# separate the header and data

header = file\_nested\_list[0]

data = file\_nested\_list[1:]

# #load data using pandas DF

import pandas as pd

**# can read a file from local computer or directly from a URL**

pd.read\_table('u.user')

pd.read\_table('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/u.user')

users = pd.read\_table('u.user', sep='|', index\_col='user\_id')

**#reading data with desired columns and sorting and indexing**

import pandas as pd

url = 'http://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.data'

col\_names = ['id','ri','na','mg','al','si','k','ca','ba','fe','glass\_type']

glass = pd.read\_csv(url, names=col\_names, index\_col='id')

glass.sort('al', inplace=True)

glass.head()

**#loading Drinks data**

drink\_cols = ['country', 'beer', 'spirit', 'wine', 'liters', 'continent']

url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/drinks.csv'

drinks = pd.read\_csv(url, header=0, names=drink\_cols, na\_filter=False)

**#loading ufo data**

url = 'https://raw.githubusercontent.com/justmarkham/DAT8/master/data/ufo.csv'

ufo = pd.read\_csv(url)

ufo['Time'] = pd.to\_datetime(ufo.Time) 🡨 changing to python datetime

ufo['Year'] = ufo.Time.dt.year <- Extracting Year from Time column

# #REST API

import requests

r = requests.get('http://www.omdbapi.com/?t=the shawshank redemption&r=json&type=movie')

r.status\_code -- 200 is success! 4xx is an error

r.text 🡨 raw response text

r.json()🡨JSON response body into a dictionary

r.json()['Year'] 🡨 Extracting year from the dictionary

**#Function to returnyear**

def get\_movie\_year(title):

r = requests.get('http://www.omdbapi.com/?t=' + title + '&r=json&type=movie')

info = r.json()

if info['Response'] == 'True':

return int(info['Year'])

else:

return None

**#load a data frame containing movies**

import pandas as pd

movies = pd.read\_csv('imdb\_1000.csv')

movies.head()

**#copy few movies**

top\_movies = movies.head().copy()

**# write a for loop to build a list of years**

from time import sleep

years = []

for title in top\_movies.title:

years.append(get\_movie\_year(title))

sleep(1)

**#check if the data frame and the years are same**

assert(len(top\_movies) == len(years))

**# save that list as a new column**

top\_movies['year'] = years

# #Querying Data Using Connector/Python

**#The following example shows how to connect to the MySQL server:**

import mysql.connector

cnx = mysql.connector.connect(user='scott', password='tiger',

host='127.0.0.1',

database='employees')

cnx.close()

**#To handle connection errors, use the try statement and catch all errors.**

import mysql.connector

from mysql.connector import errorcode

try:

cnx = mysql.connector.connect(user='scott',

database='testt')

except mysql.connector.Error as err:

if err.errno == errorcode.ER\_ACCESS\_DENIED\_ERROR:

print("Something is wrong with your user name or password")

elif err.errno == errorcode.ER\_BAD\_DB\_ERROR:

print("Database does not exist")

else:

print(err)

else:

cnx.close()

**#** [**Querying Data Using Connector/Python**](https://dev.mysql.com/doc/connector-python/en/connector-python-example-cursor-select.html)

import datetime

import mysql.connector

cnx = mysql.connector.connect(user='scott', database='employees')

cursor = cnx.cursor()

query = ("SELECT first\_name, last\_name, hire\_date FROM employees "

"WHERE hire\_date BETWEEN %s AND %s")

hire\_start = datetime.date(1999, 1, 1)

hire\_end = datetime.date(1999, 12, 31)

cursor.execute (query, (hire\_start, hire\_end))

for (first\_name, last\_name, hire\_date) in cursor:

print("{}, {} was hired on {:%d %b %Y}".format(

last\_name, first\_name, hire\_date))

cursor.close()

cnx.close()

**#other references**

**#https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html**

# #HTML document scraping using Beautiful soup

**#Beautiful Soup is essentially a set of wrapper functions that make it simple to select common HTML elements.**

**#creating a beautiful soup object**

from bs4 import BeautifulSoup

import urllib

r = urllib.urlopen('http://www.aflcio.org/Legislation-and-Politics/Legislative-Alerts').read()

soup = BeautifulSoup(r)

print type(soup)

<class 'bs4.BeautifulSoup'>

**#print data**

print soup.prettify()[0:1000]

**#reading a local file**

# read the HTML code for a web page and save as a string

with open('example.html', 'rU') as f:

html = f.read()

# convert HTML into a structured Soup object

from bs4 import BeautifulSoup

b = BeautifulSoup(html)

# print out the object

print b

print b.prettify()

**#using Requests API**

**# get the HTML from the Shawshank Redemption page**

**i**mport requests

r = requests.get('http://www.imdb.com/title/tt0111161/')

**# convert HTML into Soup**

b = BeautifulSoup(r.text)

print b

**#query data from Beautifulsoup object**

**# ResultSets can be sliced like lists**

len(b.find\_all(name='p'))

b.find\_all(name='p')[0]

b.find\_all(name='p')[0].text

b.find\_all(name='p')[0]['id']

**# iterate over a ResultSet**

results = b.find\_all(name='p')

for tag in results:

print tag.text

**# limit search by Tag attribute**

b.find(name='p', attrs={'id':'scraping'})

b.find\_all(name='p', attrs={'class':'topic'})

b.find\_all(attrs={'class':'topic'})

# limi**t search to specific sections**

b.find\_all(name='li')

b.find(name='ul', attrs={'id':'scraping'}).find\_all(name='li')

**#Reference links**

**#http://web.stanford.edu/~zlotnick/TextAsData/Web\_Scraping\_with\_Beautiful\_Soup.html**

**#https://github.com/TVMKISHORE/Analytics/blob/master/code/07\_web\_scraping.py**

# #python API for face book .

**#Documentation of Face book API**

**#https://github.com/mobolic/facebook-sdk/blob/master/docs/api.rst**

**#Example program to get posts from face book.**

**#https://github.com/mobolic/facebook-sdk/blob/master/examples/get\_posts.py**

**#Python SDK for Facebook's Graph API**

<https://github.com/mobolic/facebook-sdk>

# #Python API for twitter

**#A Python wrapper around the Twitter API.**

<https://github.com/bear/python-twitter>