

Data analysis

GETING DATA FROM YAHOO GOOGLE APIs

```
import urllib2

url =
"http://chartapi.finance.yahoo.com/instrument/1.0/F/chartdata?type=quote;range=1y/csv"
sourceCode = urllib2.urlopen(url).read()
splitSource = sourceCode.split("\n")
for eachLine in splitSource:
    splitLine = eachLine.split(',')
    if len(splitLine) == 6 and len(splitLine[0]) == 8:
        print splitLine
```

```
import urllib2

url =
"http://chartapi.finance.yahoo.com/instrument/1.0/F/chartdata?type=quote;range=1y/csv"
sourceCode = urllib2.urlopen(url).read()
splitSource = sourceCode.split("\n")
for eachLine in splitSource:
    splitLine = eachLine.split(',')
    if len(splitLine) == 6 and len(splitLine[0]) == 8:
        print splitLine
```

GETING DATA FROM YAHOO GOOGLE APIs AND COPY IT TO LOCAL FILE

```
import urllib2

count = 0
with open('mystocks.csv', 'w') as f:
    for stk in ['AAPL', 'F']:
        url = "http://chartapi.finance.yahoo.com/instrument/1.0/" + stk
```

```

+ "/chartdata?type=quote;range=1y/csv"
    sourceCode = urllib2.urlopen(url).read()
    splitSource = sourceCode.split("\n")
    for eachLine in splitSource:
        splitLine = eachLine.split(',')
        if len(splitLine) == 6 and len(splitLine[0]) == 8:
            print splitLine
            f.write(stk + "," + eachLine + "\n")
            count = count + 1

f.close()
print count

```

IT PRINTS THE PRICE STOCK FOR GOOGLE

```

import urllib2
import json

stks = ['AAPL', 'F']
for stk in stks:
    url = "http://finance.google.com/finance/info?client=ig&q=" + stk
    gf = urllib2.urlopen(url).read()
    gf = gf.replace("//", "")
    json_data = json.loads(gf)[0]
    price = json_data["1_cur"].replace("'", "\\'")
    print price

```

COUNT THE NUMBER IN THE FILE

```

count = 0
with open('mystocks.csv', 'r') as f:
    for line in f:
        print line
        count = count + 1
f.close()
print count

```

Download AAPL FILE AND TEST WITH IT

```

# looking at year stock
max = 0.0
with open('aapl.csv', 'r') as f:
    first = f.readline()

```

```
for line in f:
    splitline = line.split(",")
    if float(splitline[4]) > max:
        max = float(splitline[4])
```

```
f.close()
print max
```

PRINTS THE FIRST RECORDS

```
with open('aapl.csv', 'r') as f:
    first = f.readline()
    for line in f:
        print line
f.close()
```

LISTS

```
myStocks = []
myStocks = ['QCOM', 'AAPL', 'F']
print myStocks
myStocks.append('MSFT')
print myStocks
myStocks.insert(0, 'GOOG')
print myStocks
myStocks.pop()
x = sorted(myStocks)
print x
if 'GM' in x:
    print "GM is in stocks"
else:
    print "GM not in stocks"

if 'F' in x:
    print "F is in stocks"
else:
    print "F is not in stocks"
```

USING LISTS IN THE DATA COLLECTED FROM YAHOO AND GOOGLE APIs

#using list

```
import urllib2
```

```
count = 0
```

```
twoPercent = [] #list
```

```
for stk in ['AAPL', 'F', 'MSFT']:
```

```
    url = "http://chartapi.finance.yahoo.com/instrument/1.0/" + stk  
    + "/chartdata?type=quote;range=1y/csv"  
    sourceCode = urllib2.urlopen(url).read()  
    splitSource = sourceCode.split("\n")  
    for eachLine in splitSource:  
        splitLine = eachLine.split(',')  
        if len(splitLine) == 6 and len(splitLine[0]) == 8:  
            #print splitLine  
            cls = float(splitLine[1])  
            opn = float(splitLine[4])  
            chg = (cls-opn)/opn  
            if chg > 0.02:  
                print stk + " " + str(chg) # it convert the value to string  
                #twoPercent.append(stk + " " + str(chg))  
                twoPercent.append(splitLine[0] + " " + stk + " " + str(chg))#adding date  
            count = count + 1  
  
print count  
for x in sorted(twoPercent):  
    print x
```

DECTIONARY

```
import csv
```

```
dataFile = "aapl.csv"
```

```
f = open(dataFile, "r")
```

```
reader = csv.reader(f)
```

```
count = 0
```

```
for data in reader:
```

```
    #print data
```

```
print data[0], data[1]
count = count + 1
if count > 10:
    break
```

```
import csv
```

```
def getStockDict():
    dataFile = "aapl.csv"
    f = open(dataFile, "r")
    reader = csv.reader(f)
    myDict = {}
    for data in reader:
        myDict[data[0]] = data[1]
    return myDict
```

```
myDict = getStockDict()
myStocks = ['IBM', 'F', 'MSFT', 'DIS']
for x in myStocks:
    if x in myDict:
        print myDict[x]
    else:
        print "not found"
```

USING DECTIONARY IN YAHOO AND GOOGLE APIs

```
import urllib2
```

```
stockDict = []
for stk in ['AAPL', 'MSFT']:
    url =
"http://chartapi.finance.yahoo.com/instrument/1.0/F/chartdata?type=quote;range=1y/csv"
    sourceCode = urllib2.urlopen(url).read()
    splitSource = sourceCode.split("\n")
    for eachLine in splitSource:
        splitLine = eachLine.split(',')
        if len(splitLine) == 6 and len(splitLine[0]) == 8:
            #Date,close, high,low,open,volume
            myTupleKey = (stk, splitLine[0])

            stockDict[myTuplekey] = splitLine[1]
```

```
print stockDict[('AAPL', '20170412')] #closing price
print stockDict[('AAPL', '20170103')]#staring price
print stockDict[('AAPL', '20170412')]
print stockDict[('AAPL', '20170103')]
```

BASIC SETS

```
Python 2.7.6 (default, Oct 26 2016, 20:30:19)
[GCC 4.8.4] on linux2
Type "copyright", "credits" or "license()" for more information.
>>> sl = set()
>>> sl.add(1)
>>> sl.update([1,2,3])
>>> print si
```

```
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    print si
```

```
NameError: name 'si' is not defined
```

```
>>> print sl
set([1, 2, 3])
>>> sl.add(4)
>>> print sl
set([1, 2, 3, 4])
>>> s2 = set([4,5,6])
>>> print s2
set([4, 5, 6])
>>> sl.union(s2)
set([1, 2, 3, 4, 5, 6])
>>> sl.intersection(s2)
set([4])
>>> 7 in sl
False
>>> 4 in sl
True
>>>
```

SETS PROGRAM

```

watchList = ['MSFT', 'AAPL', 'QUCOM']
openPos = [('AAPL', 100.00, 100), ('F', 8.35, 100), ('DIS', 87.35, 100)] #{sym, buy, sh}
closedPos = [('MSFT', 25.50, 32.65, 100), ('INTC', 35.35, 27.89, 100)] #{sym, buy sell, sh}

myStocks = set()
for x in watchList:
    myStocks.add(x)

for x in openPos:
    myStocks.add(x[0])

for x in closedPos:
    myStocks.add(x[0])

print myStocks

```

PERSISTING DATA IN DATABASE AND FILES

Printing the email from the database

```

import sqlite3
from os import path
conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/simple.db")
c = conn.cursor()
c.execute('SELECT email FROM users')
for row in c.fetchall():
    print row[0]

```

printing all the data from database

```

import sqlite3
from os import path
conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/simple.db")
c = conn.cursor()
c.execute('SELECT email, last, first FROM users')
for row in c.fetchall():
    print row[0], row[1], row[2]

```

Select one records from the database

```

import sqlite3
from os import path
conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/simple.db")
c = conn.cursor()
x = 'wise',

```

```
c.execute('SELECT email, last, first FROM users WHERE last=?', x)
for row in c.fetchall():
    print row[0], row[1], row[2]
```

Insert data into the database

```
import sqlite3
from os import path
conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/simple.db")
c = conn.cursor()
x = 'opio@gmail.com', 'fazali', 'bed'
c.execute('Insert into users values(?,?,?)',x)
conn.commit()
conn.close()
print c.rowcount
```

Insert many data into the database

```
import sqlite3
from os import path
conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/simple.db")
c = conn.cursor()
employees = [('jill@mail.com', 'Jili', 'AppleTree'),
              ('frank@mail.com', 'Frank', 'AppleTree'),
              ('desi@mail.com', 'Desi', 'AppleTree'),]

c.executemany('Insert into users values(?,?,?)',employees)
conn.commit()
conn.close()
print c.rowcount
```

Updating data from the database

```
import sqlite3
from os import path
conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/simple.db")
c = conn.cursor()
x = 'refuge','homie@gmail.com',
c.execute('Update users set first=? where email=?',x)
conn.commit()
conn.close()
print c.rowcount
```


Delete data from the database

```
import sqlite3
from os import path
conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/simple.db")
c = conn.cursor()
x = 'homie@gmail.com',
c.execute('Delete from users where email=?',x)
conn.commit()
conn.close()
print c.rowcount
```

Importing data from yahoo and google APIs and save them to local database

```
import urllib2
import sqlite3

myList = []
for s in ['F', 'MSFT']:
    url = "http://chartapi.finance.yahoo.com/instrument/1.0/" + s
    url = url + "/chartdata?type=quote;range=1y/csv"
    sourceCode = urllib2.urlopen(url).read()
    splitSource = sourceCode.split("\n")
    for eachLine in splitSource:
        spl = eachLine.split(',')
        if len(spl) == 6 and len(spl[0]) == 8:
            tup = (spl[0],spl[1],spl[2],spl[3],spl[4],spl[5],s)
            myList.append(tup)

conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/stoks.db")
c = conn.cursor()
c.executemany('Insert into stockprices values(?,?,?,?,?,?,?)',myList)
conn.commit()
conn.close()
print c.rowcount
```

MISCELLANEOUS TOOLS AND PICKLE FILES

MISCELLANEOUS TOOLS

```
>sqlite3
>.open stock.db
>.tables
>.header ON
>select * from stock;
>select * from stock LIMIT 5;

>.mode list
>select * from stock LIMIT 5;
>.mode csv
>select * from stock LIMIT 5;
>.mode insert
>select * from stock LIMIT 5;
>schema #you all the table structure

>.once output.txt saves the data in the files
>select * from stock;
```

USING PICKLE

```
import sqlite3
import cPickle

#create list of tuples
def getStocks():
    stockList = []
    conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/stoks.db")
    c = conn.cursor()
    c.execute('Select date, close, high, low, open, volume, symbol from stockprices')
    for row in c.fetchall():
        stockTuple = (row[0],row[2],row[3],row[4],row[5],row[6])
        stockList.append(stockTuple)
    return stockList
```

```

# initial list
stockList = []
stockList = getStocks()
print len(stockList)

#write pickle
# r = read; w = write; rb = read binary; wb = write binary
my_file = open("stockPick.dat","wb")
#protocol version 0 (ascii),1 (binary), or 2 (binary)

cPickle.dump(stockList,my_file,2)
my_file.close()

```

READING PICKLE FILES

```

import cPickle

#r = read; w = write; rb = read binary; wb = write binary
# r/w is for ascii (protocol 0 ); rb/wb is for binary (protocol 2)

my_file = open("stockPickle.dat", "rb")
myPickleList = cPickle.load(my_file)
for x in myPickleList:
    print x[0],x[1],x[6]

```

USING A CLASS IN PICKLE AS OBJECT

```

import sqlite3
import cPickle

class Stock: pass
#create list of object
def getStocks():
    stockList = []
    conn = sqlite3.connect("/home/refuge/Desktop/python/pluralsight/stoks.db")
    c = conn.cursor()
    c.execute('Select date, close, high, low, open, volume, symbol from stockprices')
    for row in c.fetchall():
        stockTuple = (row[0],row[2],row[3],row[4],row[5],row[6])
        stock = Stock()
        stock.date = row[0]
        stock.close = row[1]

```

```
    stock.high = row[2]
    stock.low  = row[3]
    stock.open = row[4]
    stock.volume = row[5]
    stock.symbol = row[6]
    stockList.append(stock)
return stockList
```

```
# initial list
stockList = []
stockList = getStocks()
print len(stockList)
```

```
#write pickle
# r = read; w = write; rb = read binary; wb = write binary
my_file = open("stockPick.dat","wb")
#protocol version 0 (ascii),1 (binary), or 2 (binary)
```

```
cPickle.dump(stockList,my_file,2)
my_file.close()
```

```
#saving data in the file
```

```
my_file = open ("objectPickle.dat", "wb")
cPickle.dump(stockList, my_file, 2)
my_file.close()
```

READING FILE OBJECT

```
import cPickle
```

```
class Stock: pass
#r = read; w = write; rb = read binary; wb = write binary
# r/w is for ascii (protocol 0 ); rb/wb is for binary (protocol 2)
```

```
my_file = open("stockObjects.dat", "rb")
myPickleList = cPickle.load(my_file)
for x in myPickleList:
    stk = Stock()
    stk = x
    print stk.close, x.volume, x.date, x.symbol
```