# Python Lab Notebook

Blank notebook to be used for class exercises.

Name: Brenda Parnin

abc123: svw192

# Exercise 1

Write code that reads the csv file "housing_prices.csv" and calculate/print the following:

- Calculate and print the sum of all house prices. Do not use the sum() method
- Calculate and print the average price. Do not use any external packages.
- Calculate and print the max price (all prices are > 0). Do not use the max() method.
- Print the name of the street that contains the house with the most expensive house.

The path for the file is "./house_prices.csv".

**Important Python concepts:** for, lists, open ('r'), if (>)

Run the cell below to view the "house_prices.csv" file.

In [1]:

```
with open("house_prices.csv") as iFile:
    print(iFile.read())
```

```
"street name","square feet","price"
"Sreet 1",400,10000
"Street 2",650,15000
"Street 3",1000,20000
```

In [2]:

```
import csv

price_sum = 0
num_house = 0
max_price = 0
max_price_name = ""

myfile = open("house_prices.csv")
myCSV = csv.reader(myfile, delimiter= ',')
next(myCSV)

for row in myCSV:
    num_house += 1

    cur_price = float(row[2])
    price_sum += cur_price
    #print(row)
    if cur_price > max_price:
        max_price = cur_price
        max_price_name = row[0]

myfile.close()
```

```
avg_price = price_sum/num_house
print("Num Houses: {}".format(num_house))
print("Avg Price: {}".format(avg_price))
print("Max Price: {}".format(max_price))
print("Street with Max Price: {}".format(max_price_name))
```

```
Num Houses: 3
Avg Price: 15000.0
Max Price: 20000.0
Street with Max Price: Street 3
```

# Exercise 2

Given the following list of lists

myData = [['name','department','birthday month'], ['JohnDoe','Marketing','November'],['Jane Smith', 'IT', 'March']]

create a csv file that is delimited with the tab ('\t') character using the csv.writer() method. Name the file "employee_birthday.csv".

In [3]:

```
myData = [['name','department','birthday month'], ['John Doe','Marketing','November'],[
```

In [4]:

```
import csv
# write code to write the list of lists above to the CSV file "employee_birthday.csv"
out_file = open("employee_birthday.csv", 'w', newline='')

oCSV = csv.writer(out_file, delimiter= '\t')

for row in myData:
    oCSV.writerow(row)

out_file.close()
```

Run the line below to check your work:

In [5]:

```
with open("employee_birthday.csv", "r") as inFile:
    print(inFile.read())
```

```
name    department      birthday month
John Doe        Marketing       November
Jane Smith      IT      March
```

# Exercise 3

A garden center has an XML (plant_catalog.xml) file that stores information, including price, for all plants they sell. The store is having a sale where everything is 20% off. Write a program that that prints the plant "COMMON" name, the current price, and the new sale price. An example of what the output should look like is shown below:

```
Bloodroot $2.44 to $1.95
Columbine $9.37 to $7.50
```

```
Marsh Marigold $6.81 to $5.45
...
```

File absolute path: "./plant_catalog.xml" **Hint:** You will need to use "string indexing".

In [6]:

```
# Use this code to look at the structure of plant_catalog.xml
# n = number of lines to show
n = 26
with open("plant_catalog.xml") as myfile:
    head = [next(myfile) for x in range(n)]
print(''.join(head))
```

```
<?xml version="1.0" encoding="UTF-8"?>
<CATALOG>
  <PLANT>
    <COMMON>Bloodroot</COMMON>
    <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$2.44</PRICE>
    <AVAILABILITY>031599</AVAILABILITY>
  </PLANT>
  <PLANT>
    <COMMON>Columbine</COMMON>
    <BOTANICAL>Aquilegia canadensis</BOTANICAL>
    <ZONE>3</ZONE>
    <LIGHT>Mostly Shady</LIGHT>
    <PRICE>$9.37</PRICE>
    <AVAILABILITY>030699</AVAILABILITY>
  </PLANT>
  <PLANT>
    <COMMON>Marsh Marigold</COMMON>
    <BOTANICAL>Caltha palustris</BOTANICAL>
    <ZONE>4</ZONE>
    <LIGHT>Mostly Sunny</LIGHT>
    <PRICE>$6.81</PRICE>
    <AVAILABILITY>051799</AVAILABILITY>
  </PLANT>
```

In [7]:

```
# Write code here
import xml.etree.ElementTree as ET

myFile = open("plant_catalog.xml")

xml_string = myFile.read()

root = ET.fromstring(xml_string)

plants = root.findall('PLANT')
for p in plants:
    name = p.find("COMMON").text
    price = p.find("PRICE").text
    new_price = 0.8*float(price[1:])

    print("{} {} to ${:.2f}".format(name,price,new_price))
myFile.close()
```

```
Bloodroot $2.44 to $1.95
Columbine $9.37 to $7.50
Marsh Marigold $6.81 to $5.45
Cowslip $9.90 to $7.92
```

```
Dutchman's-Breeches $6.44 to $5.15
Ginger, Wild $9.03 to $7.22
Hepatica $4.45 to $3.56
Liverleaf $3.99 to $3.19
Jack-In-The-Pulpit $3.23 to $2.58
Mayapple $2.98 to $2.38
Phlox, Woodland $2.80 to $2.24
Phlox, Blue $5.59 to $4.47
Spring-Beauty $6.59 to $5.27
Trillium $3.90 to $3.12
Wake Robin $3.20 to $2.56
Violet, Dog-Tooth $9.04 to $7.23
Trout Lily $6.94 to $5.55
Adder's-Tongue $9.58 to $7.66
Anemone $8.86 to $7.09
Grecian Windflower $9.16 to $7.33
Bee Balm $4.59 to $3.67
Bergamot $7.16 to $5.73
Black-Eyed Susan $9.80 to $7.84
Buttercup $2.57 to $2.06
Crowfoot $9.34 to $7.47
Butterfly Weed $2.78 to $2.22
Cinquefoil $7.06 to $5.65
Primrose $6.56 to $5.25
Gentian $7.81 to $6.25
Blue Gentian $8.56 to $6.85
Jacob's Ladder $9.26 to $7.41
Greek Valerian $4.36 to $3.49
California Poppy $7.89 to $6.31
Shooting Star $8.60 to $6.88
Snakeroot $5.63 to $4.50
Cardinal Flower $3.02 to $2.42
```

# Exercise 4

Using the "exampleJSON.json" file, complete the following tasks:

- Load the file into a python dictionary.
- Change the email of item with the name "Anthony" to "anthony.rios@utsa.edu"
- Add a new person to the list with the name "David" and email "david@fakeemail.edu"
- Save the new dictionary to a JSON file "exampleJSON2.json"

File path: ./exampleJSON.json

In [11]:

```
with open("exampleJSON.json") as iFile:
    print(iFile.read())
```

In [15]:

```
with open("exampleJSON (3).json") as iFile:
    print(iFile.read())
```

```
[{"name": "Anthony", "email": "a@utsa.edu", "age": 102}, {"name": "John", "email": "john
@fake.edu"}, {"name": "Jane", "email": "jane@fake.edu"}]
```

In [17]:

```
# Write code here
import json

inFile = open('exampleJSON (3).json')
```

```
JSON_to_dict = json.load(inFile)

inFile.close()
```

In [18]:

```
JSON_to_dict[0]['email'] = "anthony.rios@utsa.edu"
newObj = {'name': 'David', 'email': 'david@fakeemail.edu'}
#JSON_to_dict.append(newObj)
JSON_to_dict.append(newObj)
```

In [19]:

```
JSON_to_dict
```

Out[19]:

```
[{'name': 'Anthony', 'email': 'anthony.rios@utsa.edu', 'age': 102},
 {'name': 'John', 'email': 'john@fake.edu'},
 {'name': 'Jane', 'email': 'jane@fake.edu'},
 {'name': 'David', 'email': 'david@fakeemail.edu'}]
```

In [20]:

```
outFile = open('exampleJSON2.json','w')

json.dump(JSON_to_dict, outFile)

outFile.close()
```

In [21]:

```
nums = [1,2,True,4]

json.dumps(nums)
```

Out[21]:

```
'[1, 2, true, 4]'
```

Run the following cell to check your work

In [22]:

```
# Run this cell to check your code
with open("exampleJSON2.json") as iFile:
    print(iFile.read())
```

```
[{"name": "Anthony", "email": "anthony.rios@utsa.edu", "age": 102}, {"name": "John", "em
ail": "john@fake.edu"}, {"name": "Jane", "email": "jane@fake.edu"}, {"name": "David", "e
mail": "david@fakeemail.edu"}]
```

# Exercise 5

Write code to loop over the Twitter JSONL file "twitter.jsonl" and compute the following:

- Count and print the total number of tweets.
- Count and print the total number of users are in the dataset. Hint: row['user']['screen_name']
- Print the screen name of the user who has the most tweets.

**Tip:** Don't process the entire file right away, start by processing 1 to 2 lines.

Run the next cell to view the first row line of the file

In [23]:

```
import pprint
import json
with open('./twitter.jsonl') as iFile:
    for row in iFile:
        pprint.pprint(json.loads(row.strip()))
        break
```

```
{'contributors': None,
 'coordinates': None,
 'created_at': 'Thu Aug 18 17:17:12 +0000 2016',
 'display_text_range': [0, 95],
 'entities': {'hashtags': [],
              'symbols': [],
              'urls': [{'display_url': 'dlvr.it/M3sHSw',
                        'expanded_url': 'http://dlvr.it/M3sHSw',
                        'indices': [72, 95],
                        'url': 'https://t.co/uIV7TKHs9K'}],
              'user_mentions': []},
 'favorite_count': 1,
 'favorited': False,
 'full_text': 'Adam Cole Praises Kevin Owens + A Preview For Next Week’s ROH '
              'Broadcast https://t.co/uIV7TKHs9K',
 'geo': None,
 'id': 766323071976247296,
 'id_str': '766323071976247296',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="https://dlvrit.com/" rel="nofollow">dlvr.it</a>',
 'truncated': False,
 'user': {'contributors_enabled': False,
          'created_at': 'Thu Dec 05 09:48:45 +0000 2013',
          'default_profile': False,
          'default_profile_image': False,
          'description': 'i sing my own rhythm.',
          'entities': {'description': {'urls': []}},
          'favourites_count': 0,
          'follow_request_sent': False,
          'followers_count': 76,
          'following': False,
          'friends_count': 15,
          'geo_enabled': False,
          'has_extended_profile': False,
          'id': 2231233110,
          'id_str': '2231233110',
          'is_translation_enabled': False,
          'is_translator': False,
          'lang': 'en',
          'listed_count': 34,
          'location': 'main; @Kan1shk3',
          'name': '',
          'notifications': False,
          'profile_background_color': 'FFFFFF',
          'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.
png',
          'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/the
me1/bg.png',
          'profile_background_tile': False,
          'profile_banner_url': 'https://pbs.twimg.com/profile_banners/2231233110/138762
2004',
          'profile_image_extensions_alt_text': None,
          'profile_image_url': 'http://pbs.twimg.com/profile_images/414342229096808449/f
YvzqXN7_normal.png',
          'profile_image_url_https': 'https://pbs.twimg.com/profile_images/4143422290968
08449/fYvzqXN7_normal.png',
          'profile_link_color': '08C2C2',
          'profile_sidebar_border_color': 'FFFFFF',
          'profile_sidebar_fill_color': 'DDEEF6',
          'profile_text_color': '333333',
          'profile_use_background_image': True,
          'protected': False,
          'screen_name': 'sheezy0',
          'statuses_count': 151093,
          'time_zone': None,
          'translator_type': 'none',
          'url': None,
          'utc_offset': None,
          'verified': False}}
```

In [24]:

```
# Write Code here
import json

numTweets = 0
userTweetCnts = {}

tweetFile = open('twitter.jsonl')

for line in tweetFile:
    numTweets += 1

    tweetObj = json.loads(line.strip())

    sName = tweetObj['user']['screen_name']
    userTweetCnts[sName] = userTweetCnts.get(sName, 0) + 1

tweetFile.close()

print(numTweets)
print(len(userTweetCnts))

maxUser = ''
maxTweets = 0
for k,v in userTweetCnts.items():
   #print(k,v)
    if v > maxTweets:
        maxTweets = v
        maxUser = k
print(maxUser)
```

```
10000
4
sheezy0
```