Python Lab Notebook

Blank notebook to be used for class exercises.

Name: Rudy Martinez

abc123: Lpe538

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 [71]:

```
import csv

house_count = 0

```

```python
total_price = 0
max_price = 0
house_street = ''

my_file = open('house_prices.csv')
my_csv = csv.reader(my_file, delimiter = ',')
next(my_csv)

for row in my_csv:
    house_count += float(1)
    current_price = float(row[2])
    total_price += current_price

    if current_price > max_price:
        max_price = current_price
        house_street = row[0]

my_file.close()

average_price = total_price / house_count


print(f"The sum of all house prices is: ${total_price}\n")
print(f"The average price is ${average_price}\n")
print(f"The max price is: ${max_price}\n")
print(f"The street name is: {house_street}")
```

```
The sum of all house prices is: $45000.0

The average price is $15000.0

The max price is: $20000.0
The street name is: 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 [73]:

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

In [82]:

```
import csv

my_file = open('employee_birthday.csv', 'w', newline = '')
csvWriter = csv.writer(my_file, delimiter = '\t')

for row in myData:
    csvWriter.writerow(row)

my_file.close()
```

Run the line below to check your work:

In [83]:

```
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 [84]:

```python
# 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 [134…

```
import xml.etree.ElementTree as ET
to_open = open('plant_catalog.xml')
input = to_open.read()
lst = ET.fromstring(input)

names = tree.findall('PLANT')

for item in lst:
    name = item.find('COMMON').text
    price = item.find('PRICE').text
    new_price = round(float(item.find('PRICE').text[1:]) * .80, 2)
    print(f"{name} {price} to ${new_price}")
to_open.close()
```

```
Bloodroot $2.44 to $1.95
Columbine $9.37 to $7.5
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 $4.5
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 [149…

```python
with open("exampleJSON.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 [179…

```python
import json


myFile = open('exampleJSON.json')
data = json.load(myFile)
myFile.close()
```

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

with open('exampleJSON2.json', 'w') as iFile:
    json.dump(data, iFile)
```

Run the following cell to check your work

In [177…

```
# 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", "email": "john@fake.edu"},
{"name": "Jane", "email": "jane@fake.edu"}, {"name": "David", "email": "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 [180…

```
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/theme1/bg.png',
                 'profile_background_tile': False,
                 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/2231233110/1387622004',
                 'profile_image_extensions_alt_text': None,
                 'profile_image_url': 'http://pbs.twimg.com/profile_images/414342229096808449/fYvzqXN7_normal.png',
                 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/414342229096808449/fYvzqXN7_normal.p
ng',
                 '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 [200…

```
import json

tweet_count = 0
user_tweet_counts = {}
max_tweets = 0
max_user = ''

myFile = open('twitter.jsonl')
for line in myFile:
    tweet_count += 1

    lineData = json.loads(line.strip())
    sName = lineData['user']['screen_name']
    user_tweet_counts[sName] = user_tweet_counts.get(sName, 0) + 1

for k,v in user_tweet_counts.items():
    if v > max_tweets:
        max_tweets = v
        max_user = k

myFile.close()

print(f'Number of Tweets: {tweet_count}')
print(f'Number of Users in the Dataset: {len(user_tweet_counts)}')
print(f'Screen Name: {max_user}')
```

```
Number of Tweets: 10000
Number of Users in the Dataset: 4
Screen Name: sheezy0
```