

# Python Lab Notebook

Blank notebook to be used for class exercises.

Name: Dan Schumacher

abc123: hdd249

## 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 [6]: import csv
```

```
In [28]: 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 [44]: myFile = open('house_prices.csv')
          csvRead = csv.reader(myFile, delimiter = ',')

          sum_house_prices = 0
          number_of_houses = 0
          max_house_price = 0
          max_price_house_name = ''

          isHeader = True
          for row in csvRead:
              if isHeader:
                  isHeader = False
              else:
                  house_cost = float(row[2])
                  sum_house_prices += house_cost
                  number_of_houses += 1
```

```

        if house_cost > max_house_price:
            max_house_price = house_cost
            max_price_house_name = row[0]

average_house_price = sum_house_prices/number_of_houses

print('Sum of house prices:', '${:,}'.format(sum_house_prices))
print('Average house price:', '${:,}'.format(average_house_price))
print('Max house price', '${:,}'.format(max_house_price))
print('The name of the max-house is:',max_price_house_name)

```

```

Sum of house prices: $45,000.0
Average house price: $15,000.0
Max house price $20,000.0
The name of the max-house 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 [47]: myData = [['name','department','birthday month'], ['John Doe','Marketing','November'],
```

```
In [50]: myFile = open('employee_birthday.csv','w', newline='')
csvWriter = csv.writer(myFile, delimiter = '\t')
for row in myData:
    csvWriter.writerow(row)
myFile.close()
```

Run the line below to check your work:

```
In [51]: 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 [83]: import xml.etree.ElementTree as ET
to_open = open('./plant_catalog.xml')
input = to_open.read()
stuff = ET.fromstring(input)

lst = stuff.findall('PLANT')

for item in lst:
    name = item.find('COMMON').text
    price = item.find('PRICE').text
    discounted_price = float(price.split('$')[1]) * 0.8
    discounted_price = '${:,.2f}'.format(round(discounted_price,2))
    print(name, price, 'to', discounted_price)
```

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

```
In [14]: # 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 [ ]: # Write code here
```

## 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 [82]: 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 [83]: import json
inFile = open('exampleJSON.json')
jsonData = json.load(inFile)
inFile.close()

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

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

Run the following cell to check your work

```
In [84]: # 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 [5]: 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://dlvr.it/" 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/2231233110/138

```

```

7622004',
    'profile_image_extensions_alt_text': None,
    'profile_image_url': 'http://pbs.twimg.com/profile_images/41434222909680844
9/fYvzqXN7_normal.png',
    'profile_image_url_https': 'https://pbs.twimg.com/profile_images/4143422290
96808449/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 [74]: myFile = open('twitter.jsonl')
total_tweet_count = 0
number_of_users = 0
set_of_screen_names = set()
num_tweets_tracker = {}

for line in myFile:
    row = json.loads(line.strip())
    total_tweet_count += 1
    screen_name = row['user']['screen_name']
    set_of_screen_names.add(screen_name)
#dictionary to keep track of counts

myFile.close()

for name in set_of_screen_names:
    num_tweets_tracker[name] = 0
    number_of_users += 1
    myFile = open('twitter.jsonl')
    for line in myFile:
        row = json.loads(line.strip())
        screen_name = row['user']['screen_name']
        if name == screen_name:
            num_tweets_tracker[name] += 1

myFile.close()

print('Total Tweet Count:', total_tweet_count)
print('\n')
print('Number of Users:', number_of_users)
print('\n')
print('Number of Tweets by User: ', num_tweets_tracker)

```

Total Tweet Count: 10000

Number of Users: 4

Number of Tweets by User: {'DavdBurnett': 3192, 'sheezy0': 3243, 'douggarland': 1832, 'WilfordGemma': 1733}