HW1

August 27, 2025

1 HW1 - Python Exercises

1.1 Exercises

Answer the questions or complete the tasks outlined in bold below, use the specific method described if applicable.

Maximum Number of Points: 30. Questions 1-10 (2 points each). Questions 11 (10 points).

1. What is 7 to the power of 4?

[3]:

```
```

[3]:

```
2401
```

2. Split this string:

```
s = "Hi there Sam!"
```

into a list.

[5]:

```
```

[6]:

```
```

[6]:

```
['Hi', 'there', 'Sam!']
```

3. Given the variables:

```
planet = "Earth"
diameter = 12742
```

Use .format() to print the following string:

```
The diameter of Earth is 12742 kilometers.
```

[8]:

```
planet = "Earth"
diameter = 12742
```

[11]:

```
```

```
The diameter of Earth is 12742 kilometers.
```

4. Given this nested list, use indexing to grab the word "hello"

```
[12]: lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
```

```
[13]:
```

```
[13]: ['hello']
```

5. Given this nested dictionary grab the word "hello". Be prepared, this will be annoying/tricky

```
[16]: d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':
        ↪[1,2,3,'hello']}]}]}
```

```
[22]:
```

```
[22]: 'hello'
```

6. What is the main difference between a tuple and a list?

```
[ ]:
```

7. Create a function that grabs the email website domain from a string in the form:

`user@domain.com`

So for example, passing "user@domain.com" would return: domain.com

```
[26]:
```

```
[27]: domainGet('user@domain.com')
```

```
[27]: 'domain.com'
```

8. Create a basic function that returns True if the word 'dog' is contained in the input string. Importantly, the function should not distinguish between capitalization. This means that if the input string contains 'doG', the function should still yield True.

```
[36]:
```

```
[37]: findDog('Is there a dog here?')
```

```
[37]: True
```

9. Create a function that counts the number of times the word "dog" occurs in a string.

```
[38]:
```

```
[39]: countDog('This dog runs faster than the other dog dude!')
```

```
[39]: 2
```

10. Use lambda expressions and the filter() function to filter out words from a list that don't start with the letter 's'. For example:

```
seq = ['soup','dog','salad','cat','great']
```

should be filtered down to:

```
['soup','salad']
```

[41]:

```
seq = ['soup','dog','salad','cat','great']
```

[43]:

```

```

[43]:

```
['soup', 'salad']
```

11. Final Problem

You are driving a little too fast, and a police officer stops you. Write a function to return one of 3 possible results: "No ticket", "Small ticket", or "Big Ticket". If your speed is 60 or less, the result is "No Ticket". If speed is between 61 and 80 inclusive, the result is "Small Ticket". If speed is 81 or more, the result is "Big Ticket". Unless it is your birthday (encoded as a boolean value in the parameters of the function) – on your birthday, your speed can be 5 higher in all cases.

[44]:

```
def caught_speeding(speed, is_birthday):
    pass
```

[45]:

```
caught_speeding(81,True)
```

[45]:

```
'Small Ticket'
```

[46]:

```
caught_speeding(81,False)
```

[46]:

```
'Big Ticket'
```

2 HW1 - NumPy Exercise

2.1 Exercises

Answer the questions or complete the tasks outlined in bold below, use the specific method described if applicable.

Maximum Number of Points: 40. Questions 1-20 (2 points each).

1. Import NumPy as np

[1]:

```

```

2. Create an array of 10 zeros

[2]:

```

```

[2]:

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

3. Create an array of 10 ones

```
[4]:
```

```
[4]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

4. Create an array of 10 fives

```
[5]:
```

```
[5]: array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

5. Create an array of the integers from 10 to 50

```
[7]:
```

```
[7]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
            27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
            44, 45, 46, 47, 48, 49, 50])
```

6. Create an array of all the even integers from 10 to 50

```
[10]:
```

```
[10]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
             44, 46, 48, 50])
```

7. Create a 3x3 matrix with values ranging from 0 to 8

```
[12]:
```

```
[12]: array([[0, 1, 2],
             [3, 4, 5],
             [6, 7, 8]])
```

8. Create a 3x3 identity matrix

```
[16]:
```

```
[16]: array([[1., 0., 0.],
             [0., 1., 0.],
             [0., 0., 1.]])
```

9. Use NumPy to generate a random number between 0 and 1

```
[18]:
```

```
[18]: array([0.72061738])
```

10. Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

```
[19]:
```

```
[19]: array([-0.29233072,  0.6149879 ,  0.62826371, -0.41211443,  0.07533853,
       -1.05007604,  0.48632808,  0.32808645,  0.29371673, -0.07529016,
        0.38813823,  0.61931288, -0.24723198, -0.60339607,  0.21665797,
       -0.67345814,  0.445508  , -1.97456597,  1.96788751,  1.18614563,
        1.45326745, -0.59075696, -1.58267156, -2.78914624,  0.69546833])
```

11. Create the following matrix:

```
[22]:
```

```
[22]: array([[0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 ],
       [0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 ],
       [0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 ],
       [0.31, 0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 ],
       [0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.5 ],
       [0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6 ],
       [0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7 ],
       [0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8 ],
       [0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9 ],
       [0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1.  ]])
```

12. Create an array of 20 linearly spaced points between 0 and 1:

```
[23]:
```

```
[23]: array([0.        , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
       0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
       0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
       0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.        ])
```

2.2 Numpy Indexing and Selection

Now you will be asked to replicate the resulting matrix outputs:

```
[ ]: # 13. Denote the matrix below as mat (you can create such a matrix using
      ↪arange() and reshape()).
```

```
[25]:
```

```
[25]: array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

```
[3]: # 14. WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
     # BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
     # BE ABLE TO SEE THE OUTPUT ANY MORE
```

[26]:

```
```

[26]:

```
array([[12, 13, 14, 15],
       [17, 18, 19, 20],
       [22, 23, 24, 25]])
```

[29]:

```
# 15. WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
```

[28]:

```
```

[28]:

```
20
```

[30]:

```
# 16. WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
```

[32]:

```
```

[32]:

```
array([[ 2],
       [ 7],
       [12]])
```

[31]:

```
# 17. WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
# BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
# BE ABLE TO SEE THE OUTPUT ANY MORE
```

[33]:

```
```

[33]:

```
array([21, 22, 23, 24, 25])
```

2.2.1 Now do the following

18. Get the sum of all the values in mat, which is the matrix that question 13 asked you to create

[36]:

```
```

[36]:

```
325
```

19. Get the standard deviation of the values in mat

[39]:

```
```

[39]:

```
7.211102550927978
```

20. Get the sum of all the columns in mat

[42]:

```
```

[42]:
```
array([55, 60, 65, 70, 75])
```

3 HW1 - Pandas Exercises

3.1 Exercises

Answer the questions or complete the tasks outlined in bold below, use the specific method described if applicable.

Maximum Number of Points: 30. Questions 1-15 (2 points each).

1. Import pandas as pd.

[2]:
```

```

2. Read Salaries.csv as a dataframe called sal.

[3]:
```

```

[3]:
```
             Id       EmployeeName  \
0             1     NATHANIEL FORD
1             2       GARY JIMENEZ
2             3     ALBERT PARDINI
3             4  CHRISTOPHER CHONG
4             5    PATRICK GARDNER
…        …                …
148649  148650      Roy I Tillery
148650  148651       Not provided
148651  148652       Not provided
148652  148653       Not provided
148653  148654          Joe Lopez

                                              JobTitle    BasePay  \
0       GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY  167411.18
1                      CAPTAIN III (POLICE DEPARTMENT)  155966.02
2                      CAPTAIN III (POLICE DEPARTMENT)  212739.13
3                 WIRE ROPE CABLE MAINTENANCE MECHANIC   77916.00
4         DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)  134401.60
…                                                …        …
148649                                       Custodian       0.00
148650                                    Not provided        NaN
148651                                    Not provided        NaN
148652                                    Not provided        NaN
148653                      Counselor, Log Cabin Ranch       0.00

        OvertimePay   OtherPay  Benefits   TotalPay  TotalPayBenefits  Year  \
0              0.00  400184.25       NaN  567595.43         567595.43  2011
1         245131.88  137811.38       NaN  538909.28         538909.28  2011
2         106088.18   16452.60       NaN  335279.91         335279.91  2011
```

```
3           56120.71  198306.90       NaN  332343.61         332343.61  2011
4            9737.00  182234.59       NaN  326373.19         326373.19  2011
…             …        …       …        …               …   …
148649         0.00       0.00       0.0       0.00              0.00  2014
148650          NaN        NaN       NaN       0.00              0.00  2014
148651          NaN        NaN       NaN       0.00              0.00  2014
148652          NaN        NaN       NaN       0.00              0.00  2014
148653         0.00    -618.13       0.0    -618.13           -618.13  2014

        Notes         Agency  Status
0         NaN  San Francisco     NaN
1         NaN  San Francisco     NaN
2         NaN  San Francisco     NaN
3         NaN  San Francisco     NaN
4         NaN  San Francisco     NaN
…       …            …     …
148649    NaN  San Francisco     NaN
148650    NaN  San Francisco     NaN
148651    NaN  San Francisco     NaN
148652    NaN  San Francisco     NaN
148653    NaN  San Francisco     NaN

[148654 rows x 13 columns]
```

3. Check the head of the DataFrame.

[4]:

[4]:

```
   Id       EmployeeName                                          JobTitle  \
0   1     NATHANIEL FORD  GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY
1   2       GARY JIMENEZ                  CAPTAIN III (POLICE DEPARTMENT)
2   3     ALBERT PARDINI                  CAPTAIN III (POLICE DEPARTMENT)
3   4  CHRISTOPHER CHONG             WIRE ROPE CABLE MAINTENANCE MECHANIC
4   5    PATRICK GARDNER    DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)

     BasePay  OvertimePay   OtherPay  Benefits   TotalPay  TotalPayBenefits  \
0  167411.18         0.00  400184.25       NaN  567595.43         567595.43
1  155966.02    245131.88  137811.38       NaN  538909.28         538909.28
2  212739.13    106088.18   16452.60       NaN  335279.91         335279.91
3   77916.00     56120.71  198306.90       NaN  332343.61         332343.61
4  134401.60      9737.00  182234.59       NaN  326373.19         326373.19

   Year  Notes         Agency  Status
0  2011    NaN  San Francisco     NaN
1  2011    NaN  San Francisco     NaN
2  2011    NaN  San Francisco     NaN
3  2011    NaN  San Francisco     NaN
4  2011    NaN  San Francisco     NaN
```

4. Use the .info() method to find out how many entries there are.

[5]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148654 entries, 0 to 148653
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Id                148654 non-null  int64  
 1   EmployeeName      148654 non-null  object 
 2   JobTitle          148654 non-null  object 
 3   BasePay           148045 non-null  float64
 4   OvertimePay       148650 non-null  float64
 5   OtherPay          148650 non-null  float64
 6   Benefits          112491 non-null  float64
 7   TotalPay          148654 non-null  float64
 8   TotalPayBenefits  148654 non-null  float64
 9   Year              148654 non-null  int64  
 10  Notes             0 non-null       float64
 11  Agency            148654 non-null  object 
 12  Status            0 non-null       float64
dtypes: float64(8), int64(2), object(3)
memory usage: 14.7+ MB
```

5. What is the average BasePay ?

[6]:

```
[6]: 66325.44884050643
```

6. What is the highest amount of OvertimePay in the dataset?

[7]:

```
[7]: 245131.88
```

7. What is the job title of JOSEPH DRISCOLL? Note: Use all caps, otherwise you may get an answer that doesn't match up (there is also a lowercase Joseph Driscoll).

[9]:

```
[9]: 24    CAPTAIN, FIRE SUPPRESSION
     Name: JobTitle, dtype: object
```

8. How much does JOSEPH DRISCOLL make (including benefits)?

[10]:

```
[10]: 24    270324.91
      Name: TotalPayBenefits, dtype: float64
```

9. What is the name of highest paid person (including benefits)?

[11]:

[11]:
```
   Id    EmployeeName                                        JobTitle  \
0   1  NATHANIEL FORD  GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY

     BasePay  OvertimePay   OtherPay  Benefits   TotalPay  TotalPayBenefits  \
0  167411.18          0.0  400184.25       NaN  567595.43         567595.43

   Year  Notes         Agency  Status
0  2011    NaN  San Francisco     NaN
```

10. What is the name of lowest paid person (including benefits)? Do you notice something strange about how much he or she is paid?

[12]:

[12]:
```
            Id EmployeeName                    JobTitle  BasePay  OvertimePay  \
148653  148654    Joe Lopez  Counselor, Log Cabin Ranch      0.0          0.0

        OtherPay  Benefits  TotalPay  TotalPayBenefits  Year  Notes  \
148653   -618.13       0.0   -618.13           -618.13  2014    NaN

               Agency  Status
148653  San Francisco     NaN
```

11. What was the average (mean) BasePay of all employees per year? (2011-2014)?

[15]:

[15]:
```
Year
2011    63595.956517
2012    65436.406857
2013    69630.030216
2014    66564.421924
Name: BasePay, dtype: float64
```

12. How many unique job titles are there?

[16]:

[16]:
```
2159
```

13. What are the top 5 most common jobs?

[29]:

[29]:
```
Transit Operator                7036
Special Nurse                   4389
```

```
Registered Nurse                3736
Public Svc Aide-Public Works    2518
Police Officer 3                2421
Name: JobTitle, dtype: int64
```

14. How many Job Titles were represented by only one person in 2013? (e.g. Job Titles with only one occurence in 2013?)

[27]:

```
[27]: 202
```

15. How many people have the word Chief in their job title? (This can be tricky)

[31]:

```
[31]: 627
```