PYTHON TRAINING WEEK 2 EXERCISE

Attempt all Questions

1. Import the numpy library

In [1]:
```
import numpy as np
```

2a. Create a 1D numpy array with the values [1, 2, 3, 4, 5] and assign it to a variable named "a"

2b. Find the shape of the array "a"

In [2]:
```
a=np.array([1,2,3,4,5])
a1= a.shape
print("The shape of the array is", str(a1))
```

```
The shape of the array is (5,)
```

3. Find the minimum and maximum values in the array "a"

In [3]:
```
print("The minimum value is ",str(min(a)))
print("The minimum value is " , str(max(a)))
```

```
The minimum value is  1
The minimum value is  5
```

4a. Create a 2D array with [6, 7, 8, 9 10], [11, 12, 13, 14, 15] and assign it to a variable called "b"

In [4]:
```
b=np.array([[6,7,8,9,10], [11,12,13,14,15]])
```

4b Find the sum and product of "a" & "b"

In [5]:
```
c= a + b
c
```

Out[5]:
```
array([[ 7,  9, 11, 13, 15],
       [12, 14, 16, 18, 20]])
```

In [6]:
```
d= a*b
print(d)
```

```
[[ 6 14 24 36 50]
 [11 24 39 56 75]]
```

5. Transpose "b"

hint: use transpose method

In [7]:
```
b_transpose=b.transpose()
print(b_transpose)
```

```
[[ 6 11]
 [ 7 12]
 [ 8 13]
 [ 9 14]
 [10 15]]
```

In [8]:
```
arr = np.array([45, 75, 55, 34, 21, 6])
```

6a. Sort 'arr'

6b. Reshape 'arr' to 3 by 2

In [9]:
```
arr.sort()
print(arr)
```

```
[ 6 21 34 45 55 75]
```

In [10]:
```
arr_reshape=arr.reshape(3,2)
print(arr_reshape)
```

```
[[ 6 21]
 [34 45]
 [55 75]]
```

7. Create a numpy array with shape (5, 4) filled with zeros

In [11]:
```
np_zero= np.zeros((5,4))
print(np_zero)
```

```
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

Pandas

1. Import pandas library

In [12]:
```
import pandas as pd
```

2. Convert "a" arrays to pandas series and assign it to a variable called series_a

In [13]:
```
series_a= pd.Series(a)
series_a
```

Out[13]:
```
0    1
1    2
2    3
3    4
4    5
dtype: int32
```

3. Convert "b" array to pandas dataframe and assign it to a variable called dataframe_b

In [14]:
```
dataframe_b = pd.DataFrame(b)
dataframe_b
```

Out[14]:

	0	1	2	3	4
0	6	7	8	9	10
1	11	12	13	14	15

4. What is the difference between a pandas series and dataFrame

write text here

Section C- Data Exploration with Pandas

In [15]:
```
values = {'points': ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j'],
            'X': [78, 85, 96, 80, 86, 100, 55, 80, 45, 30],
            'Y': [84, 94, 89, 83, 86, 0, 66, 78, 65, 25],
            'Z': [86, 97, 96, 72, 83,  0, 99, 43, 67, 2]  }
```

Using the values dictionary answer the following question

1. Create a new dataframe called df

In [16]:
```
df= pd.DataFrame(values)
df
```

Out[16]:

	points	X	Y	Z
0	a	78	84	86
1	b	85	94	97
2	c	96	89	96
3	d	80	83	72
4	e	86	86	83
5	f	100	0	0
6	g	55	66	99
7	h	80	78	43
8	i	45	65	67
9	j	30	25	2

2a. How many row and columns are in df?

2b. Print the first 3 rows in df

In [17]:
```
rows= len(df.axes[0])
cols= len(df.axes[1])
print("Number of rows: " , rows)
print("Number of columns: " , cols)
```

```
Number of rows:  10
Number of columns:  4
```

In [18]:
```
df.head(3)
```

Out[18]:

	points	X	Y	Z
0	a	78	84	86
1	b	85	94	97
2	c	96	89	96

3. Check the number of columns and rows in the dataframe

hint: use shape method

In [19]:
```
rows_1=df.shape[0]
col_2=df.shape[1]
print("Number of rows: " , rows_1)
print("Number of columns: " , col_2)
```

```
Number of rows:  10
Number of columns:  4
```

4. Check dataframe info

In [20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   points  10 non-null     object
 1   X       10 non-null     int64 
 2   Y       10 non-null     int64 
 3   Z       10 non-null     int64 
dtypes: int64(3), object(1)
memory usage: 448.0+ bytes
```

5. Describe the dataframe

Hint: use .describe

In [21]:
```
df.describe()
```

Out[21]:

	X	Y	Z
count	10.000000	10.000000	10.000000
mean	73.500000	67.000000	64.500000
std	22.726636	30.728199	37.467764
min	30.000000	0.000000	0.000000
25%	60.750000	65.250000	49.000000
50%	80.000000	80.500000	77.500000
75%	85.750000	85.500000	93.500000
max	100.000000	94.000000	99.000000

6. Print the out the columns available

In [22]:
```
df.columns
```

Out[22]:
```
Index(['points', 'X', 'Y', 'Z'], dtype='object')
```

7. Rename the columns from uppercase to lower case

In [23]:
```
df.columns=df.columns.str.lower()
df
```

Out[23]:

	points	x	y	z
0	a	78	84	86
1	b	85	94	97
2	c	96	89	96
3	d	80	83	72
4	e	86	86	83
5	f	100	0	0
6	g	55	66	99
7	h	80	78	43
8	i	45	65	67
9	j	30	25	2

8. What is the data type in the different columns

In [24]:
```
df.dtypes
```

Out[24]:
```
points    object
x          int64
y          int64
z          int64
dtype: object
```

9. What is the min, max and median in the x, y and z columns respectively

In [25]:
```
print("The mininmum value is : " ,df['x'].min())
print("The maxinmum value is : " ,df['x'].max())
print("The median value is : " ,df['x'].median())
```

```
The mininmum value is :  30
The maxinmum value is :  100
The median value is :  80.0
```

10. Make the 'point' column the index of the dataframe

In [26]:
```
df = df.set_index('points')
```

In [27]:
```
df
```

Out[27]:

points	x	y	z
a	78	84	86
b	85	94	97
c	96	89	96
d	80	83	72
e	86	86	83
f	100	0	0
g	55	66	99
h	80	78	43
i	45	65	67
j	30	25	2

11. Create a new column called 'sum_xyz' which is a sum of x, y and z columns

In [28]:
```
df['sum_xyz']= df['x'] + df['y'] + df['z']
df
```

Out[28]:

points	x	y	z	sum_xyz
a	78	84	86	248
b	85	94	97	276
c	96	89	96	281
d	80	83	72	235
e	86	86	83	255
f	100	0	0	100
g	55	66	99	220
h	80	78	43	201
i	45	65	67	177
j	30	25	2	57

12. Delete column "z"

In [29]:
```
df.pop('z')
```

Out[29]:
```
points
a    86
b    97
c    96
d    72
e    83
f     0
g    99
h    43
i    67
j     2
Name: z, dtype: int64
```

In [30]:
```
df
```

Out[30]:

points	x	y	sum_xyz
a	78	84	248
b	85	94	276
c	96	89	281
d	80	83	235
e	86	86	255
f	100	0	100
g	55	66	220
h	80	78	201
i	45	65	177
j	30	25	57

13. Create a new column called 'x_over_y' which is a which is a division of x and y

In [31]:
```
df['x_over_y'] = df['x'] /df['y']
```

In [32]:
```
df
```

Out[32]:

points	x	y	sum_xyz	x_over_y
a	78	84	248	0.928571
b	85	94	276	0.904255
c	96	89	281	1.078652
d	80	83	235	0.963855
e	86	86	255	1.000000
f	100	0	100	inf
g	55	66	220	0.833333
h	80	78	201	1.025641
i	45	65	177	0.692308
j	30	25	57	1.200000

In [ ]:
```

```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js