In [1]:
```
import pandas as pd
import numpy as np
Class_performance = {'names':['joe', 'mark', 'faith', 'ace', 'john', 'lucky'],
                     'Gpa':[88, 78, 73, 73, 72, 71],
                     'rank':[1, 2, 3, 3, 5, 6]}
performance_data = pd.DataFrame(Class_performance)
performance_data
```

Out[1]:

	names	Gpa	rank
0	joe	88	1
1	mark	78	2
2	faith	73	3
3	ace	73	3
4	john	72	5
5	lucky	71	6

In [2]:
```
performance_data.names
```

Out[2]:
```
0      joe
1     mark
2    faith
3      ace
4     john
5    lucky
Name: names, dtype: object
```

In [3]:
```
performance_data.Gpa
```

Out[3]:
```
0    88
1    78
2    73
3    73
4    72
5    71
Name: Gpa, dtype: int64
```

In [4]:
```
print(pd.__version__)
```

```
1.2.4
```

series

In [5]:
```
# creating a series from a list
marks = [76,77,78,82,73,79]
c1 = pd.Series(marks)
c1
```

Out[5]:
```
0    76
```

```
1    77
2    78
3    82
4    73
5    79
dtype: int64
```

In [6]:
```
rank = [1,2,3,4,5,6]
c2 = pd.Series(marks, index = rank)
c2
```

Out[6]:
```
1    76
2    77
3    78
4    82
5    73
6    79
dtype: int64
```

In [ ]:
```

```

In [7]:
```
import numpy as np
numbers = np.random.randn(7)
numbers
```

Out[7]:
```
array([ 0.77470227, -1.49012574, -0.9970397 ,  0.43703786, -1.04659871,
        0.62950665,  1.19047892])
```

In [8]:
```
# creating series from dictionary
import pandas as pd
capital_city = {'Kenya':'Nairobi', 'USA':'Washington Dc', 'France': 'Paris'}
d = pd.Series(capital_city)
d
```

Out[8]:
```
Kenya           Nairobi
USA       Washington Dc
France            Paris
dtype: object
```

In [9]:
```
#slicing
d[:-2]
```

Out[9]:
```
Kenya    Nairobi
dtype: object
```

In [10]:
```
d.drop('USA')
d
```

Out[10]:
```
Kenya           Nairobi
USA       Washington Dc
France            Paris
dtype: object
```

In [11]:
```
arr1 = [1,2,3,4,5,6,7]
arr2 = ['a','b','c','d']
```

```
s7 = pd.Series(arr2)
s7
```

Out[11]:
```
0    a
1    b
2    c
3    d
dtype: object
```

In [12]:
```
s8 = pd.Series(arr1)
s8
```

Out[12]:
```
0    1
1    2
2    3
3    4
4    5
5    6
6    7
dtype: int64
```

In [13]:
```
c1.add(s8)
```

Out[13]:
```
0    77.0
1    79.0
2    81.0
3    86.0
4    78.0
5    85.0
6     NaN
dtype: float64
```

In [14]:
```
print('median:', c1.median())
```

```
median: 77.5
```

In [15]:
```
print("maximum:",c1.max())
```

```
maximum: 82
```

DataFrames

In [16]:
```
#creating dataframe with dictionary array
data = {'Day':['monday','tuesday','wednesday','thursday', 'friday', 'saturday', 'sunday
       'No Of Eggs' : [1300, 1327, 1298,1287,1356,1309,1358],
       'prices':[15.0,15.5,15.3,16,15.9,16.5,16.0]}
labels = ['a','b','c','d','e','f','g']
df1 = pd.DataFrame(data, index = labels)
df1
```

Out[16]:

	Day	No Of Eggs	prices
a	monday	1300	15.0
b	tuesday	1327	15.5

	Day	No Of Eggs	prices
c	wednesday	1298	15.3
d	thursday	1287	16.0
e	friday	1356	15.9
f	saturday	1309	16.5
g	sunday	1358	16.0

In [ ]:

```
```

In [17]:

```
df1.dtypes
df1.head(3)
df1.tail()
```

Out[17]:

	Day	No Of Eggs	prices
c	wednesday	1298	15.3
d	thursday	1287	16.0
e	friday	1356	15.9
f	saturday	1309	16.5
g	sunday	1358	16.0

In [18]:

```
df1.describe()
```

Out[18]:

	No Of Eggs	prices
count	7.000000	7.000000
mean	1319.285714	15.742857
std	28.517330	0.506153
min	1287.000000	15.000000
25%	1299.000000	15.400000
50%	1309.000000	15.900000
75%	1341.500000	16.000000
max	1358.000000	16.500000

In [19]:

```
df1.columns
```

Out[19]:

```
Index(['Day', 'No Of Eggs', 'prices'], dtype='object')
```

In [20]:

```
units = ['sco 202', 'sco 204', 'sco 107', 'sma 204', 'sma 203', 'sst 205']
c3 = pd.Series(marks,index = index)
```

```
data ={'UNITS':units,'MARKS' : marks,'GRADE': ['A','A','A','A','A','A']}
labels  =[1,2,3,4,5,6]
df7 = pd.DataFrame(data, index = labels)
df7
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-20-1b6b226e088a> in <module>
      1 units = ['sco 202', 'sco 204', 'sco 107', 'sma 204', 'sma 203', 'sst 205']
----> 2 c3 = pd.Series(marks,index = index)
      3 data ={'UNITS':units,'MARKS' : marks,'GRADE': ['A','A','A','A','A','A']}
      4 labels  =[1,2,3,4,5,6]
      5 df7 = pd.DataFrame(data, index = labels)

NameError: name 'index' is not defined
```

In [ ]:
```
df7.mean()
```

In [ ]:
```
dates = pd.date_range('today', periods=6)
num_arr = np.random.randn(6,5)
columns = ['A','B','C','D','E']

df2 = pd.DataFrame(num_arr, index=dates, columns = columns)
df2
```

In [ ]:
```
df2.values
```

In [ ]:
```
df2.describe()
```

In [ ]:
```
df2.T
```

In [ ]:
```
df1.T
```

In [ ]:
```
df1.sort_values(by = 'prices')
```

In [ ]:
```
# slicing of dataframes
df2[1:4]
```

In [ ]:
```
# query dataframe by tag
df1[['Day', 'No Of Eggs']]
```

In [ ]:
```
# query rows 2,3
df2.iloc[1:3]
```

In [ ]:
```
df1.iloc[0:4]
```

In [ ]:
```
df1.head(4)
```

In [ ]:
```
df3 = df2.copy()
df3
```

In [ ]:
```
df1.isnull()
```

In [ ]:
```
df1
```

In [ ]:
```
df1.mean()
```

In [ ]:
```
df1.sum()
```

operations for DataFrame missing values

In [ ]:
```
data = {'Day':['monday','tuesday','wednesday','thursday', 'friday', 'saturday', 'sunday
       'No Of Eggs' : [1300, 1327,np.nan,1287,np.nan,1309,1358],
       'prices':[15.0,15.5,16,np.nan,15.9,16.5,16.0]}
labels = ['a',' b','c','d','e','f','g']
df5 = pd.DataFrame(data, index = labels)
df5
```

In [ ]:
```
df5
```

In [ ]:
```
df5.dropna(how = 'any')
```

DataFrame file operations

In [ ]:
```
df5.to_csv('Eggs_production.csv')
df_production = pd.read_csv('Eggs_production.csv')
df_production.head()
```

visualization in pandas

In [ ]:
```
import numpy as np
import pandas as pd
%matplotlib inline

ts = pd.Series(np.random.randn(50), index = pd.date_range('today', periods = 50))
ts =ts.cumsum()
ts.plot()
```

In [ ]:
```
df = pd.DataFrame(np.random.randn(50,4), index=ts.index,
                  columns = ['A','B','C','D'])
df = df.cumsum()
df.plot()
```

removing repeated data in pandas

In [ ]:
```
df = pd.DataFrame({'A':[1,2,2,2,4,7,6,6,4,1,9,7,5,3,1,3,3]})
df.loc[df['A'].shift() != df['A']]
```

In [26]:
```
import pandas
pd.read_csv('wines reviews.csv')
reviews
wine reviews_csv.to_csv('wines reviews.csv')
```

```
  File "<ipython-input-26-7fbd4bdc68a3>", line 4
    wines reviews_csv.to_csv('wines reviews.csv')
          ^
SyntaxError: invalid syntax
```

In [ ]:
```

```