DataFrame:

DataFrame is a Two-Dimensional Array with homogenious and hetrogenious data.

In DataFrame Rows are homogenious/hetrogenious but columns must be homogenious.

ex1: create a dataframe by using list objects?

```
import pandas as pd
x=[[1,2,3],[4,5,6]]
y=pd.DataFrame(x)
print(y)
print(type(y))
```

output:

```
   0  1  2
0  1  2  3
1  4  5  6
<class 'pandas.core.frame.DataFrame'>
```

ex2: create a dataframe object by using list comprihension

```
import pandas as pd
x=[[i,i*2] for i in range(3)]
y=pd.DataFrame(x)
print(y)
print(type(y))
```

output:

```
   0  1
0  0  0
1  1  2
2  2  4
<class 'pandas.core.frame.DataFrame'>
```

ex3: create a dataframe by using list with columns?

```
import pandas as pd
x=[['siva',29],['rama',30],['krishna',28]]
y=pd.DataFrame(data=x,columns=['Name','Age'])
print(y)
print(type(y))
```

output:

```
      Name  Age
0     siva   29
```

```
1     rama   30
2  krishna   28
<class 'pandas.core.frame.DataFrame'>
```

ex4: create a dataframe by using dict object

```
import pandas as pd
x={'Name':['siva','rama','krishna'],
   'Age':[29,30,28]}
df=pd.DataFrame(data=x)
print(df)
print(type(df))
```

output:

```
      Name  Age
0     siva   29
1     rama   30
2  krishna   28
<class 'pandas.core.frame.DataFrame'>
```

ex5:

```
import pandas as pd
x=[{'name':'siva','age':29},
   {'name':'rama','age':30},
   {'name':'krishna','age':28}]
df=pd.DataFrame(x,index=['FirstEmploye','SecondEmploye',
                         'ThirdEmploye'])
print(df)
print(type(df))
```

output:

```
                  name  age
FirstEmploye      siva   29
SecondEmploye     rama   30
ThirdEmploye   krishna   28
<class 'pandas.core.frame.DataFrame'>
```

ex6:

---

create a dataframe by using zip()?

```
import pandas as pd
names=['siva','rama','krishna']
ages=[29,30,28]
sal=[2000,4000,3000]
data=zip(names,ages,sal)
```

```
df=pd.DataFrame(data,columns=['Names','Ages','Salary'])
print(df)
print('*'*20)
print(df.loc[0])
print('*'*20)
print(df.to_dict())
print('*'*20)
df.to_json('emp.json')
df.to_csv('emp.csv')
df.to_csv('emp1.csv',header=False,index=False)
df=pd.read_csv('emp1.csv')
print(df)
print('*'*20)
df=pd.read_csv('emp1.csv',names=["Names","Ages","Salary"])
print(df)
print('*'*20)
print(df['Salary'].max())
print(df['Salary'].min())
print(df['Salary'].sum())
print(df['Salary'].mean())
print(df['Salary'].head(1))
print(df['Salary'].tail(1))
print('*'*20)
df=pd.read_json('emp.json')
print(df)
```

output:

```
------
     Names   Ages    Salary
0     siva    29    2000
1     rama    30    4000
2  krishna    28    3000
********************

Names     siva
Ages        29
Salary    2000
Name: 0, dtype: object
********************

{'Names': {0: 'siva', 1: 'rama', 2: 'krishna'}, 'Ages': {0: 29, 1: 30, 2: 28},
'Salary': {0: 2000, 1: 4000, 2: 3000}}
********************
```

emp.json file is created

emp.csv file is created

emp1.csv file is cretaed without headings and indexes

```
      siva  29  2000
0     rama  30  4000
1  krishna  28  3000
********************

      Names  Ages  Salary
0     siva    29    2000
1     rama    30    4000
2  krishna    28    3000
********************

4000
2000
9000
3000.0
0    2000
Name: Salary, dtype: int64
2    3000
Name: Salary, dtype: int64
********************

     Names  Ages  Salary
0     siva    29    2000
1     rama    30    4000
2  krishna    28    3000
```