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

```
2-D Data
columns and rows

1. DataFrame  >> 2-D   >>  rows & columns
2. Series     >> 1-D   >>  rows

DataFrame >> It is a data structure that is used to store the data in form of rows and columns (Spreadsheet).
DataFrame is Mutable.
Add/update/delete : rows and columns
Pandas DataFrame can be created using Dict,csv,excel,DataBase(MongoDB,sqlLite),List,Array
string in dataframe is called object.

DataBase uses structured query language for writing and quering.
2. Series     >> 1-D   >>  rows
Data
Base uses the table to store data.
```

In [2]:
```
data={'A':[10,20,30,40,50],'B':[100,200,300,400,500]}
type(data)
```

Out[2]:
```
dict
```

In [3]:
```
df=pd.DataFrame(data)
df
```

Out[3]:

	A	B
0	10	100
1	20	200
2	30	300
3	40	400
4	50	500

In [5]:
```
series=pd.Series(data)
series
```

Out[5]:
```
A         [10, 20, 30, 40, 50]
B    [100, 200, 300, 400, 500]
dtype: object
```

In [13]:
```
df
```

Out[13]:
```
A         [10, 20, 30, 40, 50]
B    [100, 200, 300, 400, 500]
C                [1, 2, 3, 4, 5]
dtype: object
```

In [14]:
```
df=pd.DataFrame(data)
df
```

Out[14]:

	A	B
0	10	100
1	20	200
2	30	300
3	40	400
4	50	500

In [17]:
```
df['A']  >> Series >> 1-D
```

Out[17]:
```
0    10
1    20
2    30
3    40
4    50
Name: A, dtype: int64
```

In [20]:
```
df[['A']]  #>> 2-D

```

Out[20]:

	A
0	10
1	20
2	30
3	40
4	50

In [21]:
```
df=pd.DataFrame(data)
df

```

Out[21]:

	A	B
0	10	100
1	20	200
2	30	300
3	40	400
4	50	500

In [22]:
```
df['C']=[89,56,23,78,45]     # Adding columns in data frame
```

In [23]:
```
1 df
```

Out[23]:

	A	B	C
0	10	100	89
1	20	200	56
2	30	300	23
3	40	400	78
4	50	500	45

In [24]:
```
1 df[['A','C']]      # Accessing dataframe
```

Out[24]:

	A	C
0	10	89
1	20	56
2	30	23
3	40	78
4	50	45

Read csv / excel file :-

```
1 df=pd.read_csv('filename.csv')     # for reading csv file
2 df=pd.read_excel('filename.xlsx')  # for reading excel file
```

In [29]:
```
1 df=pd.read_csv('Iris.csv')
2 df
```

Out[29]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [30]:
```
df=pd.read_excel('Emp_Records.xlsx')
df
```

Out[30]:

	Emp ID	First Name	Age in Yrs	Weight in Kgs	Age in Company	Unnamed: 5	City	Salary
0	677509	Lois	36.36	60	13.68	NaN	Denver	168251
1	940761	Brenda	47.02	60	9.01	NaN	Stonewall	51063
2	428945	Joe	54.15	68	0.98	NaN	Michigantown	50155
3	408351	Diane	39.67	51	18.30	NaN	Hydetown	180294
4	193819	Benjamin	40.31	58	4.01	NaN	Fremont	117642
...	...	...	...	...	...	...	...	...
95	639892	Jose	22.82	89	1.05	NaN	Biloxi	129774
96	704709	Harold	32.61	77	5.93	NaN	Carol Stream	156194
97	461593	Nicole	52.66	60	28.53	NaN	Detroit	95673
98	392491	Theresa	29.60	57	6.99	NaN	Mc Grath	51015
99	495141	Tammy	38.38	55	2.26	NaN	Alma	93650

100 rows × 8 columns

In [31]:
```
arr1=np.random.randint(10,100,size=(20,6))
arr1
```

Out[31]:
```
array([[26, 44, 84, 29, 45, 61],
       [36, 99, 77, 20, 95, 49],
       [12, 66, 61, 98, 61, 80],
       [28, 82, 81, 61, 50, 61],
       [84, 66, 96, 33, 55, 20],
       [80, 36, 31, 17, 49, 77],
       [29, 31, 90, 48, 97, 40],
       [57, 25, 50, 61, 75, 73],
       [52, 87, 81, 86, 66, 24],
       [76, 59, 66, 53, 17, 22],
       [33, 46, 73, 17, 29, 60],
       [61, 41, 58, 31, 48, 79],
       [26, 92, 46, 99, 85, 83],
       [82, 37, 45, 21, 74, 45],
       [56, 95, 43, 54, 62, 16],
       [68, 43, 50, 60, 33, 66],
       [38, 21, 16, 30, 29, 35],
       [82, 22, 87, 24, 30, 93],
       [49, 42, 13, 69, 64, 12],
       [74, 48, 61, 18, 86, 33]])
```

In [32]:
```
df=pd.DataFrame(arr1)
df
```

Out[32]:

	0	1	2	3	4	5
0	26	44	84	29	45	61
1	36	99	77	20	95	49
2	12	66	61	98	61	80
3	28	82	81	61	50	61
4	84	66	96	33	55	20
5	80	36	31	17	49	77
6	29	31	90	48	97	40
7	57	25	50	61	75	73
8	52	87	81	86	66	24
9	76	59	66	53	17	22
10	33	46	73	17	29	60
11	61	41	58	31	48	79
12	26	92	46	99	85	83
13	82	37	45	21	74	45
14	56	95	43	54	62	16
15	68	43	50	60	33	66
16	38	21	16	30	29	35
17	82	22	87	24	30	93
18	49	42	13	69	64	12
19	74	48	61	18	86	33

In [ ]:
```
```