

In [1]:

```
pip install pandas
```

Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.3)  
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)  
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)  
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

## Creating Pandas Series

In [2]:

```
import pandas as pd
import numpy as np
ser = pd.Series()
data = np.array(['g', 'p', 'r', 'e', 'c'])
ser = pd.Series(data)
print("Pandas Series:\n", ser)
```

Pandas Series:

```
0    g
1    p
2    r
3    e
4    c
dtype: object
```

In [5]:

```
import numpy as np
import pandas as pd
info= np.array(['p','a','n','d','a','s'])
print(info)
ser=pd.Series(info)
print(ser)
```

```
['p' 'a' 'n' 'd' 'a' 's']
0    p
1    a
2    n
3    d
4    a
5    s
dtype: object
```

## Accessing data from Series

In [6]:

```
import pandas as pd
import numpy as np
data = np.array(['g','p','r','e','c','k', 'u','r','n','o','o','l'])
ser = pd.Series(data)
print(ser[:5])
```

```
0    g
1    p
2    r
```

```
2
3     e
4     c
dtype: object
```

# Creating DataFrame

## Empty dataframe

In [7]:

```
# import pandas as pd
import pandas as pd

df = pd.DataFrame()

print(df)
```

Empty DataFrame  
Columns: []  
Index: []

In [8]:

```
import pandas as pd
lst = ['Assam', 'Andhra Pradesh', 'Bhopal', 'Delhi', 'Maharastra', 'Tamilnadu', 'Karnata
ka']
df = pd.DataFrame(lst)
print(df)
```

```
0
0      Assam
1  Andhra Pradesh
2      Bhopal
3      Delhi
4  Maharastra
5  Tamilnadu
6  Karnataka
```

In [9]:

```
import pandas as pd
data = {'Name':['Tom', 'nick', 'krish', 'jack'], 'Age':[20, 21, 19, 18]}
df = pd.DataFrame(data)
print(df)
```

```
   Name  Age
0   Tom   20
1  nick   21
2 krish   19
3  jack   18
```

In [10]:

```
data['Age']
```

Out[10]:

```
[20, 21, 19, 18]
```

In [11]:

```
import pandas as pd
dict = {'name':['aparna', 'pankaj', 'sudhir', 'Geeku'],
        'degree': ["MBA", "BCA", "M.Tech", "MBA"],
        'score':[90, 40, 80, 98]}
df = pd.DataFrame(dict)
print(df)
```

```
   name  degree  score
0  aparna    MBA     90
```

```
1  pankaj      BCA      40
2  sudhir    M.Tech    80
3   Geeku      MBA     98
```

In [12]:

```
import pandas as pd
data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}
df = pd.DataFrame(data)
print(df)
```

```
   calories  duration
0        420         50
1        380         40
2        390         45
```

## Giving names to index

In [13]:

```
import pandas as pd
data = {
    "calories": [420, 380, 390],
    "duration": [50, 40, 45]
}
df = pd.DataFrame(data, index = ["day1", "day2", "day3"])
print(df)
```

```
   calories  duration
day1        420         50
day2        380         40
day3        390         45
```

In [17]:

```
import pandas as pd
data=pd.read_csv("cocoa.csv")
data
```

Out[17]:

	Company \n(Maker- if known)	Specific Bean Origin\nor Bar Name	REF	Review\nDate	Cocoa\nPercent	Company\nLocation	Rating	Bean\nType	Bean\nName
0	A. Morin	Agua Grande	1876	2016	63%	France	3.75		Sao
1	A. Morin	Kpime	1676	2015	70%	France	2.75		
2	A. Morin	Atsane	1676	2015	70%	France	3.00		
3	A. Morin	Akata	1680	2015	70%	France	3.50		
4	A. Morin	Quilla	1704	2015	70%	France	3.50		
...	...	...	...	...	...	...	...	...	
1790	Zotter	Peru	647	2011	70%	Austria	3.75		
1791	Zotter	Congo	749	2011	65%	Austria	3.00	Forastero	C
1792	Zotter	Kerala State	749	2011	65%	Austria	3.50	Forastero	
1793	Zotter	Kerala State	781	2011	62%	Austria	3.25		
1794	Zotter	Brazil, Mitzi Blue	486	2010	65%	Austria	3.00		

In [18]:

```
data.tail()
```

Out[18]:

	Company \n(Maker- if known)	Specific Bean Origin\nor Bar Name	REF	Review\nDate	Cocoa\nPercent	Company\nLocation	Rating	Bean\nType	Bean\nC
1790	Zotter	Peru	647	2011	70%	Austria	3.75		
1791	Zotter	Congo	749	2011	65%	Austria	3.00	Forastero	C
1792	Zotter	Kerala State	749	2011	65%	Austria	3.50	Forastero	
1793	Zotter	Kerala State	781	2011	62%	Austria	3.25		
1794	Zotter	Brazil, Mitzi Blue	486	2010	65%	Austria	3.00		

In [19]:

```
data.head(20)
```

Out[19]:

	Company \n(Maker- if known)	Specific Bean Origin\nor Bar Name	REF	Review\nDate	Cocoa\nPercent	Company\nLocation	Rating	Bean\nType	Bean\n
0	A. Morin	Agua Grande	1876	2016	63%	France	3.75		Sa
1	A. Morin	Kpime	1676	2015	70%	France	2.75		
2	A. Morin	Atsane	1676	2015	70%	France	3.00		
3	A. Morin	Akata	1680	2015	70%	France	3.50		
4	A. Morin	Quilla	1704	2015	70%	France	3.50		
5	A. Morin	Carenero	1315	2014	70%	France	2.75	Criollo	Ver
6	A. Morin	Cuba	1315	2014	70%	France	3.50		
7	A. Morin	Sur del Lago	1315	2014	70%	France	3.50	Criollo	Ver
8	A. Morin	Puerto Cabello	1319	2014	70%	France	3.75	Criollo	Ver
9	A. Morin	Pablino	1319	2014	70%	France	4.00		
10	A. Morin	Panama	1011	2013	70%	France	2.75		F
11	A. Morin	Madagascar	1011	2013	70%	France	3.00	Criollo	Mada
12	A. Morin	Brazil	1011	2013	70%	France	3.25		
13	A. Morin	Equateur	1011	2013	70%	France	3.75		E
14	A. Morin	Colombie	1015	2013	70%	France	2.75		Co
15	A. Morin	Birmanie	1015	2013	70%	France	3.00		
16	A. Morin	Papua New Guinea	1015	2013	70%	France	3.25		Pap
17	A. Morin	Chuao	1015	2013	70%	France	4.00	Trinitario	Ver
18	A. Morin	Piura	1019	2013	70%	France	3.25		
19	A. Morin	Chanchamayo Province	1019	2013	70%	France	3.50		

In [20]:

```
data.describe()
```

Out[20]:

	REF	Review\nDate	Rating
<b>count</b>	1795.000000	1795.000000	1795.000000
<b>mean</b>	1035.904735	2012.325348	3.185933
<b>std</b>	552.886365	2.927210	0.478062
<b>min</b>	5.000000	2006.000000	1.000000
<b>25%</b>	576.000000	2010.000000	2.875000
<b>50%</b>	1069.000000	2013.000000	3.250000
<b>75%</b>	1502.000000	2015.000000	3.500000
<b>max</b>	1952.000000	2017.000000	5.000000

In [21]:

```
import pandas as pd
```

In [26]:

```
df=pd.read_csv("iris.csv")
```

In [27]:

```
df
```

Out[27]:

	sepal_length	sepal_width	petal_length	petal_width	target
<b>0</b>	5.1	3.5	1.4	0.2	0
<b>1</b>	4.9	3.0	1.4	0.2	0
<b>2</b>	4.7	3.2	1.3	0.2	0
<b>3</b>	4.6	3.1	1.5	0.2	0
<b>4</b>	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
<b>145</b>	6.7	3.0	5.2	2.3	2
<b>146</b>	6.3	2.5	5.0	1.9	2
<b>147</b>	6.5	3.0	5.2	2.0	2
<b>148</b>	6.2	3.4	5.4	2.3	2
<b>149</b>	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

In [28]:

```
df.head()
```

Out[28]:

	sepal_length	sepal_width	petal_length	petal_width	target
<b>0</b>	5.1	3.5	1.4	0.2	0
<b>1</b>	4.9	3.0	1.4	0.2	0

2	sepal_length	sepal_width	petal_length	petal_width	target
	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [29]:

```
df.tail()
```

Out[29]:

	sepal_length	sepal_width	petal_length	petal_width	target
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

In [30]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
#   Column          Non-Null Count  Dtype
---  -
0   sepal_length    150 non-null   float64
1   sepal_width     150 non-null   float64
2   petal_length    150 non-null   float64
3   petal_width     150 non-null   float64
4   target          150 non-null   int64
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
```

In [31]:

```
df.describe()
```

Out[31]:

	sepal_length	sepal_width	petal_length	petal_width	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667	1.000000
std	0.828066	0.433594	1.764420	0.763161	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

In [32]:

```
df.isnull().sum()
```

Out[32]:

```
sepal_length    0
sepal_width     0
petal_length    0
petal width     0
```

```
target      0  
dtype: int64
```

```
In [33]:
```

```
df.shape
```

```
Out[33]:
```

```
(150, 5)
```

```
In [34]:
```

```
df.columns
```

```
Out[34]:
```

```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'target'], dtype='object')
```

```
In [ ]:
```