Pandas ===> It is an open-source library that is used for handle data manipulation.

In [1]:
```
# Data Structure :
#     (1) . Series  (2). DataFrame
```

In [2]:
```
import pandas as pd
```

In [3]:
```
a = pd.Series([1,23,67,90])
a
```

Out[3]:
```
0     1
1    23
2    67
3    90
dtype: int64
```

In [4]:
```
type(a)
```

Out[4]:
```
pandas.core.series.Series
```

(2). DataFrame

In [5]:
```
a = {
    "Emp_ID" : [1,2,3,4,5,6,7,8],
    "Name" : ['Sam' , 'Mohit' , 'Gorav' , 'Aniket' , 'Raj' , 'Rahul' , 'Deepak'
    "Department" : ['IT','HR','HR','IT','Opeartions','IT','Opeartions','HR'],
    "Working_Hours" : [8,9,9,9,9,9,6,7]
}
```

In [6]:
```
type(a)
```

Out[6]:
```
dict
```

In [7]:
```
df = pd.DataFrame(a)
df
```

Out[7]:

	Emp_ID	Name	Department	Working_Hours
0	1	Sam	IT	8
1	2	Mohit	HR	9
2	3	Gorav	HR	9
3	4	Aniket	IT	9
4	5	Raj	Opeartions	9
5	6	Rahul	IT	9
6	7	Deepak	Opeartions	6
7	8	Kunal	HR	7

In [8]:
```
df.columns
```

Out[8]:
```
Index(['Emp_ID', 'Name', 'Department', 'Working_Hours'], dtype='object')
```

In [9]:
```
df.head()        # it returns top 5 row
```

Out[9]:

	Emp_ID	Name	Department	Working_Hours
0	1	Sam	IT	8
1	2	Mohit	HR	9
2	3	Gorav	HR	9
3	4	Aniket	IT	9
4	5	Raj	Opeartions	9

In [10]:
```
df.tail()    # it returns bottom 5 row
```

Out[10]:

	Emp_ID	Name	Department	Working_Hours
3	4	Aniket	IT	9
4	5	Raj	Opeartions	9
5	6	Rahul	IT	9
6	7	Deepak	Opeartions	6
7	8	Kunal	HR	7

In [11]:
```
df.tail(2)
```

Out[11]:

	Emp_ID	Name	Department	Working_Hours
6	7	Deepak	Opeartions	6
7	8	Kunal	HR	7

In [12]:
```
df.head(3)
```

Out[12]:

	Emp_ID	Name	Department	Working_Hours
0	1	Sam	IT	8
1	2	Mohit	HR	9
2	3	Gorav	HR	9

In [13]:
```
df.sample(4)   # it returns random-indexed rows.
```

Out[13]:

	Emp_ID	Name	Department	Working_Hours
0	1	Sam	IT	8
2	3	Gorav	HR	9
5	6	Rahul	IT	9
6	7	Deepak	Opeartions	6

In [14]:
```
df.describe()      # it returns statically view of data.
```

Out[14]:

	Emp_ID	Working_Hours
count	8.00000	8.000000
mean	4.50000	8.250000
std	2.44949	1.164965
min	1.00000	6.000000
25%	2.75000	7.750000
50%	4.50000	9.000000
75%	6.25000	9.000000
max	8.00000	9.000000

In [15]:
```
df.info()    # Complete overview of data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Emp_ID         8 non-null      int64 
 1   Name           8 non-null      object
 2   Department     8 non-null      object
 3   Working_Hours  8 non-null      int64 
dtypes: int64(2), object(2)
memory usage: 388.0+ bytes
```

In [ ]:
```
# If you want to export your data in csv format
```

In [18]:
```
df.to_csv("C:\\Users\\yashs\\OneDrive\\Desktop\\ty\\yash.csv")
```

In [19]:
```
df.to_csv("C:\\Users\\yashs\\OneDrive\\Desktop\\ty\\yash1.csv" , index = False)
```

In [20]:
```
df.to_excel("C:\\Users\\yashs\\OneDrive\\Desktop\\ty\\Emp_Info.xlsx")
```

How to read csv file into jupyter notebook

In [25]:
```
df=pd.read_csv("D:\\Summer Training Video\\ML\\covid_toy.csv")
```

In [26]:
```
df
```

Out[26]:

	age	gender	fever	cough	city	has_covid
0	60	Male	103.0	Mild	Kolkata	No
1	27	Male	100.0	Mild	Delhi	Yes
2	42	Male	101.0	Mild	Delhi	No
3	31	Female	98.0	Mild	Kolkata	No
4	65	Female	101.0	Mild	Mumbai	No
...	...	...	...	...	...	...
95	12	Female	104.0	Mild	Bangalore	No
96	51	Female	101.0	Strong	Kolkata	Yes
97	20	Female	101.0	Mild	Bangalore	No
98	5	Female	98.0	Strong	Mumbai	No
99	10	Female	98.0	Strong	Kolkata	Yes

100 rows × 6 columns

In [27]:
```
df.head()
```

Out[27]:

	age	gender	fever	cough	city	has_covid
0	60	Male	103.0	Mild	Kolkata	No
1	27	Male	100.0	Mild	Delhi	Yes
2	42	Male	101.0	Mild	Delhi	No
3	31	Female	98.0	Mild	Kolkata	No
4	65	Female	101.0	Mild	Mumbai	No

In [28]:
```
df['gender'].value_counts()
```

Out[28]:
```
gender
Female    59
Male      41
Name: count, dtype: int64
```

In [29]:
```
a = {
    "Emp_ID" : [1,2,3,4,5,6,7,8],
    "Name" : ['Sam' , 'Mohit' , 'Gorav' , 'Aniket' , 'Raj' , 'Rahul' , 'Deepak'
    "Department" : ['IT','HR','HR','IT','Opeartions','IT','Opeartions','HR'],
    "Working_Hours" : [8,9,9,9,9,9,6,7]
}
```

In [30]:
```
df = pd.DataFrame(a)
```

In [31]:
```
df
```

Out[31]:

	Emp_ID	Name	Department	Working_Hours
0	1	Sam	IT	8
1	2	Mohit	HR	9
2	3	Gorav	HR	9
3	4	Aniket	IT	9
4	5	Raj	Opeartions	9
5	6	Rahul	IT	9
6	7	Deepak	Opeartions	6
7	8	Kunal	HR	7

In [32]:
```
df['Name'][0] = 'Kriti'
```

```
C:\Users\yashs\AppData\Local\Temp\ipykernel_20360\1991129534.py:1: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df['Name'][0] = 'Kriti'
```

In [33]:
```
df.head()
```

Out[33]:

	Emp_ID	Name	Department	Working_Hours
0	1	Kriti	IT	8
1	2	Mohit	HR	9
2	3	Gorav	HR	9
3	4	Aniket	IT	9
4	5	Raj	Opeartions	9

In [34]:
```
# df.head()
```

In [35]:
```
# Raj ===> working_hours = 7
```

In [36]:
```
# loc and iloc

# df.loc["row_range" , "column_name"]
# df.iloc["row_range" , "column_range"]
```

In [37]:
```
df.loc[2]
```

Out[37]:
```
Emp_ID               3
Name             Gorav
Department          HR
Working_Hours        9
Name: 2, dtype: object
```

In [38]:
```
df.loc[2 , "Name"]
```

Out[38]:
```
'Gorav'
```

In [40]:
```
df.loc[2:5 , "Name"]
```

Out[40]:
```
2     Gorav
3    Aniket
4       Raj
5     Rahul
Name: Name, dtype: object
```

In [41]:
```
df.loc[2:5 , ["Name" , "Department"]]
```

Out[41]:

	Name	Department
2	Gorav	HR
3	Aniket	IT
4	Raj	Opeartions
5	Rahul	IT

In [42]:
```
df.iloc[2 , 1]
```

Out[42]:
```
'Gorav'
```

In [43]:
```
df.iloc[2 , 2]
```

Out[43]:
```
'HR'
```

In [44]:
```
df.iloc[2:5 , 2]
```

Out[44]:
```
2            HR
3            IT
4    Opeartions
Name: Department, dtype: object
```

In [45]:
```
df.iloc[2:5 , [1,2]]
```

Out[45]:

	Name	Department
2	Gorav	HR
3	Aniket	IT
4	Raj	Opeartions

In [46]:
```
df.iloc[2:5 , 1:3]
```

Out[46]:

	Name	Department
2	Gorav	HR
3	Aniket	IT
4	Raj	Opeartions

In [47]:
```
df['Name'][2] = None
```

```
C:\Users\yashs\AppData\Local\Temp\ipykernel_20360\1090881245.py:1: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df['Name'][2] = None
```

In [51]:
```
df['Department'][4] = None
```

```
C:\Users\yashs\AppData\Local\Temp\ipykernel_20360\2611955995.py:1: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df['Department'][4] = None
```

In [52]:
```
df
```

Out[52]:

	Emp_ID	Name	Department	Working_Hours
0	1	Kriti	IT	8
1	2	Mohit	HR	9
2	3	None	HR	9
3	4	Aniket	IT	9
4	5	Raj	None	9
5	6	Rahul	IT	9
6	7	Deepak	Opeartions	6
7	8	Kunal	HR	7

In [53]:
```
df['Department'][3] = None
df['Department'][4] = None
```

```
C:\Users\yashs\AppData\Local\Temp\ipykernel_20360\59155726.py:1: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df['Department'][3] = None
C:\Users\yashs\AppData\Local\Temp\ipykernel_20360\59155726.py:2: SettingWithC
opyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/s
table/user_guide/indexing.html#returning-a-view-versus-a-copy (https://panda
s.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ver
sus-a-copy)
  df['Department'][4] = None
```

In [55]:
```
df.isnull()          # it will returb true if you have missing values otherwis
```

Out[55]:

	Emp_ID	Name	Department	Working_Hours
0	False	False	False	False
1	False	False	False	False
2	False	True	False	False
3	False	False	True	False
4	False	False	True	False
5	False	False	False	False
6	False	False	False	False
7	False	False	False	False

In [57]:
```
df.isnull().sum()    # We can check total missing value using this command fro
```

Out[57]:
```
Emp_ID          0
Name            1
Department      2
Working_Hours   0
dtype: int64
```

In [58]:
```
df = df.dropna()    # using this method ,  we can drop our missing values
```

In [59]:
```
df
```

Out[59]:

	Emp_ID	Name	Department	Working_Hours
0	1	Kriti	IT	8
1	2	Mohit	HR	9
5	6	Rahul	IT	9
6	7	Deepak	Opeartions	6
7	8	Kunal	HR	7

In [ ]:
```

```

In [ ]:
```

```