```
import pandas as pd
```

```
#series pd.Series(column,index)
a=['Shankar','Binnu','Dileep','Subash','Shiva','Sakulji']
index=[1,2,3,4,5,6]
b=pd.Series(a,index)
print(b)
```

```
1    Shankar
2      Binnu
3     Dileep
4     Subash
5      Shiva
6    Sakulji
dtype: object
```

```
#Loading csv
dia=pd.read_csv("/content/diabetcsv.csv")
dia
```

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	tested_negative
764	2	122	70	27	0	36.8	0.340	27	tested_negative
765	5	121	72	23	112	26.2	0.245	30	tested_negative
766	1	126	60	0	0	30.1	0.349	47	tested_positive
767	1	93	70	31	0	30.4	0.315	23	tested_negative

768 rows × 9 columns

```
#Reading a text file
grad=pd.read_csv("/content/demodt.txt")
grad
```

	State	Literacy	Cleanliness	Crime_Rate	Good
0	A	92	90	54	0
1	B	56	67	50	1
2	C	78	85	62	0
3	D	63	72	48	1
4	E	85	79	55	0
5	F	71	68	58	0
6	G	80	83	51	0
7	H	67	74	47	1
8	I	89	88	53	0
9	J	58	65	49	1
10	K	82	81	60	0
11	L	75	78	57	0
12	M	69	70	46	1
13	N	87	86	52	0
14	O	61	63	45	1
15	P	93	91	56	0
16	Q	55	66	61	0
17	R	76	77	59	0
18	S	84	82	44	1
19	T	70	69	50	1
20	U	94	92	57	0
21	V	59	64	52	0
22	W	83	80	43	1
23	X	74	76	63	0
24	Y	68	73	41	1
25	Z	88	84	47	1

```
#Reading a xlsx file
diaxl=pd.read_excel("/content/diabetes.xlsx")
diaxl
```

	preg	plas	pres	skin	insu	mass	pedi	age	class
0	6	148	72	35	0	33.6	0.627	50	tested_positive
1	1	85	66	29	0	26.6	0.351	31	tested_negative
2	8	183	64	0	0	23.3	0.672	32	tested_positive
3	1	89	66	23	94	28.1	0.167	21	tested_negative
4	0	137	40	35	168	43.1	2.288	33	tested_positive
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	tested_negative
764	2	122	70	27	0	36.8	0.340	27	tested_negative
765	5	121	72	23	112	26.2	0.245	30	tested_negative
766	1	126	60	0	0	30.1	0.349	47	tested_positive
767	1	93	70	31	0	30.4	0.315	23	tested_negative

768 rows × 9 columns

```
#Reading a particular sheet by giving sheet name(sheet_name="sheetname")
diaxl1=pd.read_excel("/content/diabetes.xlsx",sheet_name="dora")
diaxl1
```

	Dead	Alive
0	yes	no
1	yes	no
2	yes	no
3	yes	no
4	yes	no

```
#describe a data frame
diaxl.describe()
```

	preg	plas	pres	skin	insu	mass	pedi	age
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000

```
#Shape
print(dia.shape)#returns rows and columns
print(dia.shape[0])#only rows
print(dia.shape[1])#only columns
```

```
(768, 9)
768
9
```

```
dia1=pd.read_csv("/content/grades_withnulls.csv")
dia1
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	NaN	9.1	9.9	A+	1
9	Rajini	M	NaN	9.1	9.3	A	0
10	Kiran	V	NaN	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
#finding null values,false means there is no null value and true means there is a null value
print(dia1.isnull())
```

```
    Names  Initials   SEM1   SEM2   SEM3  Grade  Placed
0   False     False  False  False  False  False   False
1   False     False  False  False  False  False   False
2   False     False  False  False  False  False   False
3   False     False  False  False   True  False   False
4   False     False  False  False  False  False   False
5   False     False  False  False  False  False   False
6   False     False  False  False  False  False   False
7   False     False  False  False  False  False   False
8   False     False   True  False  False  False   False
9   False     False   True  False  False  False   False
10  False     False   True  False  False  False   False
11  False     False  False  False  False  False   False
12  False     False  False  False  False  False   False
13  False     False  False  False  False  False   False
14  False     False  False  False  False  False   False
15  False     False  False  False  False  False   False
16  False     False  False  False  False  False   False
```

```
#display the sum of null values ina a df
print(dia1.isnull().sum())
```

```
Names       0
Initials    0
SEM1        3
SEM2        0
SEM3        1
Grade       0
Placed      0
dtype: int64
```

```
dia1.dropna()
#it just remove the null values and shows but original data frame will not be reflected remains same
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
#using another data frame to save changes by dropna
dia1=pd.read_csv("/content/grades_withnulls.csv")
df=dia1.dropna()
df
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
#using inplace=True ,remove the null values
dia1=pd.read_csv("/content/grades_withnulls.csv")
dia1.dropna(inplace=True)
dia1
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
df1=pd.read_csv("/content/grades_withnulls.csv")
df1.fillna(555,inplace=True)#Changes are saved
df1
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	555.0	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	555.0	9.1	9.9	A+	1
9	Rajini	M	555.0	9.1	9.3	A	0
10	Kiran	V	555.0	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
df1=pd.read_csv("/content/grades_withnulls.csv")
mv=df1['SEM1'].mean()
print(mv)
df1.fillna(mv,inplace=True)
df1
```

```
8.992857142857144
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.800000	10.0	9.900000	A+	1
1	Rajesh	M	8.900000	9.1	9.300000	A	1
2	Kissan	V	9.900000	9.8	10.000000	A	0
3	Mary	N	7.700000	8.0	8.992857	B	0
4	Jeen	K	9.800000	9.1	9.900000	A+	1
5	Raj	M	8.900000	9.1	9.300000	A	1
6	Hassan	V	9.900000	9.0	9.200000	A	1
7	Mari	N	7.700000	8.0	7.100000	B	1
8	Jess	K	8.992857	9.1	9.900000	A+	1
9	Rajini	M	8.992857	9.1	9.300000	A	0
10	Kiran	V	8.992857	9.3	9.200000	A	0
11	Maya	N	7.700000	8.0	7.100000	B	0
12	Jolin	K	9.800000	9.1	9.900000	A+	1
13	Rajesh	M	8.900000	9.1	9.300000	A	1
14	Riya	M	9.300000	9.9	10.000000	A	1
15	Sana	V	9.900000	9.3	9.200000	A	0
16	Mark	N	7.700000	8.0	7.000000	B	0

```
#Access the data
#iloc-integer location,index
#loc-fields names,index
#dfname.loc[index]-->rows
#dfname.loc[st:stop]-->range of rows
#dfname.loc[row_index,col_index]-->rows and columns
mydf=pd.read_csv("/content/grades_withnulls.csv")
```

```
mydf.loc[0]#first record  df.loc[n]
```

```
Names        Joe
Initials       K
SEM1         9.8
SEM2        10.0
SEM3         9.9
Grade         A+
Placed         1
Name: 0, dtype: object
```

```
#range of records
#df.loc[i,j] range of records
mydf.loc[0:5]#first five record
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1

```
mydf.iloc[5:7,0:3]
```

	Names	Initials	SEM1
5	Raj	M	8.9
6	Hassan	V	9.9

```
mydf=pd.read_csv("/content/grades_withnulls.csv")
mydf
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	NaN	9.1	9.9	A+	1
9	Rajini	M	NaN	9.1	9.3	A	0
10	Kiran	V	NaN	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
#print the records who scored 9.3 in sem3
mydf[mydf.SEM3==9.3]
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
1	Rajesh	M	8.9	9.1	9.3	A	1
5	Raj	M	8.9	9.1	9.3	A	1
9	Rajini	M	NaN	9.1	9.3	A	0
13	Rajesh	M	8.9	9.1	9.3	A	1

```
#display the data based uop the condition
mydf.loc[mydf.SEM1>9.5,'Names']
```

```
0        Joe
2     Kissan
4       Jeen
6     Hassan
12     Jolin
15      Sana
Name: Names, dtype: object
```

```
#print the grades of the students who scored more than 9 in sem3
mydf.loc[mydf.SEM3>9,"Grade"]
#accessing the records with sem3>9.7 but printing only grade column
```

```
0     A+
1      A
2      A
4     A+
5      A
```

```
6      A
8     A+
9      A
10     A
12    A+
13     A
14     A
15     A
Name: Grade, dtype: object
```

```
mydf.drop_duplicates
mydf
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1
5	Raj	M	8.9	9.1	9.3	A	1
6	Hassan	V	9.9	9.0	9.2	A	1
7	Mari	N	7.7	8.0	7.1	B	1
8	Jess	K	NaN	9.1	9.9	A+	1
9	Rajini	M	NaN	9.1	9.3	A	0
10	Kiran	V	NaN	9.3	9.2	A	0
11	Maya	N	7.7	8.0	7.1	B	0
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
mydf.head()#give top five record by deafult 5 records are displayed ,we can give specific number in the parenthesis
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.8	10.0	9.9	A+	1
1	Rajesh	M	8.9	9.1	9.3	A	1
2	Kissan	V	9.9	9.8	10.0	A	0
3	Mary	N	7.7	8.0	NaN	B	0
4	Jeen	K	9.8	9.1	9.9	A+	1

```
mydf.tail()#Last top five records
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
12	Jolin	K	9.8	9.1	9.9	A+	1
13	Rajesh	M	8.9	9.1	9.3	A	1
14	Riya	M	9.3	9.9	10.0	A	1
15	Sana	V	9.9	9.3	9.2	A	0
16	Mark	N	7.7	8.0	7.0	B	0

```
#printing one row by index
mydf[mydf.index==5]
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
5	Raj	M	8.9	9.1	9.3	A	1

```
mydf.columns#columns
```

```
Index(['Names', 'Initials', 'SEM1', 'SEM2', 'SEM3', 'Grade', 'Placed'], dtype='object')
```

```
#renaming column names
mydf.rename(columns={'Grade':'GPA'},inplace=True)
mydf.columns
```

```
Index(['Names', 'Initials', 'SEM1', 'SEM2', 'SEM3', 'GPA', 'Placed'], dtype='object')
```

```
#Creating a new column
#dfname["new column name"]=values
mydf["conduct"]="Good"
mydf
```

	Names	Initials	SEM1	SEM2	SEM3	GPA	Placed	conduct
0	Joe	K	9.8	10.0	9.9	A+	1	Good
1	Rajesh	M	8.9	9.1	9.3	A	1	Good
2	Kissan	V	9.9	9.8	10.0	A	0	Good
3	Mary	N	7.7	8.0	NaN	B	0	Good
4	Jeen	K	9.8	9.1	9.9	A+	1	Good
5	Raj	M	8.9	9.1	9.3	A	1	Good
6	Hassan	V	9.9	9.0	9.2	A	1	Good
7	Mari	N	7.7	8.0	7.1	B	1	Good
8	Jess	K	NaN	9.1	9.9	A+	1	Good
9	Rajini	M	NaN	9.1	9.3	A	0	Good
10	Kiran	V	NaN	9.3	9.2	A	0	Good
11	Maya	N	7.7	8.0	7.1	B	0	Good
12	Jolin	K	9.8	9.1	9.9	A+	1	Good
13	Rajesh	M	8.9	9.1	9.3	A	1	Good
14	Riya	M	9.3	9.9	10.0	A	1	Good
15	Sana	V	9.9	9.3	9.2	A	0	Good
16	Mark	N	7.7	8.0	7.0	B	0	Good

```
import pandas as pd
```

```
mydf=pd.read_csv("/content/grades_withnulls.csv")
mv=mydf['SEM1'].mean()
print(mv)
mydf.fillna(mv,inplace=True)
mydf
```

```
8.992857142857144
```

	Names	Initials	SEM1	SEM2	SEM3	Grade	Placed
0	Joe	K	9.800000	10.0	9.900000	A+	1
1	Rajesh	M	8.900000	9.1	9.300000	A	1
2	Kissan	V	9.900000	9.8	10.000000	A	0
3	Mary	N	7.700000	8.0	8.992857	B	0
4	Jeen	K	9.800000	9.1	9.900000	A+	1
5	Raj	M	8.900000	9.1	9.300000	A	1
6	Hassan	V	9.900000	9.0	9.200000	A	1
7	Mari	N	7.700000	8.0	7.100000	B	1
8	Jess	K	8.992857	9.1	9.900000	A+	1
9	Rajini	M	8.992857	9.1	9.300000	A	0
10	Kiran	V	8.992857	9.3	9.200000	A	0
11	Maya	N	7.700000	8.0	7.100000	B	0
12	Jolin	K	9.800000	9.1	9.900000	A+	1
13	Rajesh	M	8.900000	9.1	9.300000	A	1
14	Riya	M	9.300000	9.9	10.000000	A	1
15	Sana	V	9.900000	9.3	9.200000	A	0
16	Mark	N	7.700000	8.0	7.000000	B	0

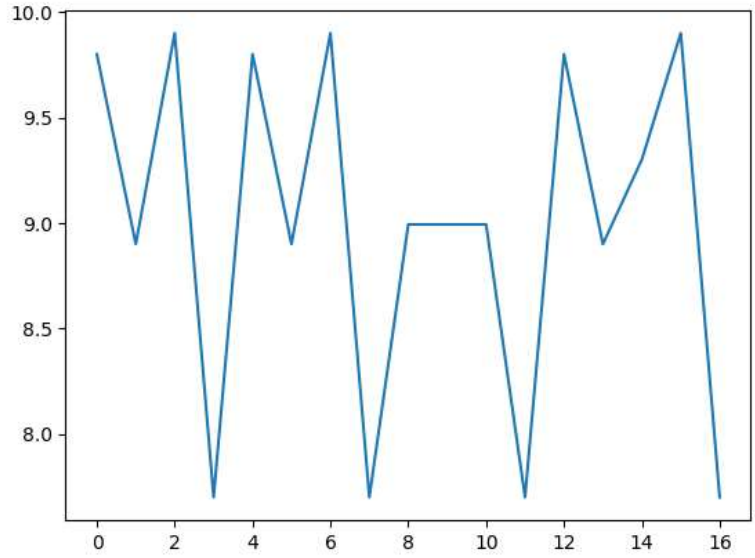
```
#create new column  with avg with avg values of three sum
mydf["Average"]=(mydf['SEM1']+mydf['SEM2']+mydf['SEM3'])/3
mydf.to_csv("mydf.csv")
```

```
from google.colab import drive
drive.mount('/content/drive')
```

plotting with pandas

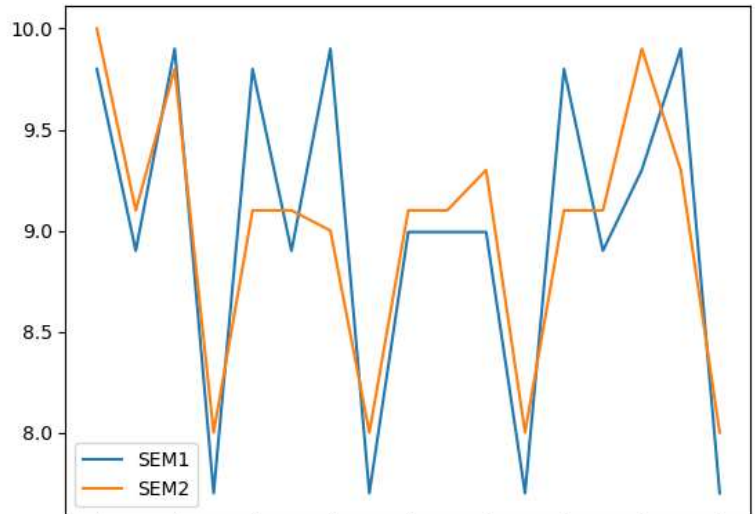
```
mydf['SEM1'].plot.line()#dfname['col'].plot.line()
```

```
<Axes: >
```




```
mydf[["SEM1","SEM2"]].plot.line()#more coulumns
```

```
<Axes: >
```




```
mydf[["SEM1","SEM2"]].plot.line(subplots=True)#Subplots
```