Pandas

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

1.Create any Series and print the output

In [3]:
```
aa=pd.Series([10,20,30,40,50])
aa
```

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

2. Create any dataframe of 10x5 with few nan values and print the output

In [4]:
```
cc=pd.DataFrame(np.random.randn(6,11))
cc
```

Out[4]:

	0	1	2	3	4	5	6	7	8
0	0.087055	0.040409	-1.202122	-0.459730	-0.257264	0.627703	-0.693306	-1.099147	0.827331
1	0.094815	1.064883	0.923173	0.277194	-0.896353	-0.811128	-1.110553	0.397514	-1.002204
2	-1.238957	-0.870804	0.985004	0.146141	0.689239	0.919842	1.308149	-0.847420	1.150464
3	-0.710920	-0.983236	0.231675	-1.276713	0.455958	-0.646559	1.027367	-0.040753	-0.521059
4	1.315673	0.241626	-0.925382	0.472495	-1.357830	-0.233035	-0.685131	0.100734	-0.118617
5	0.218941	0.606138	-0.969310	1.077064	-1.945778	1.483430	0.853154	-0.188808	-0.406128

In [30]:

```
dd=pd.DataFrame(
{
    "a":1.0,
    "b":pd.Timestamp("20230721"),
    "g":pd.Series(np.nan,index=list(range(11))),
    "f":100,
    "h":2003,
    "o":562213,
    "j":pd.Series(np.nan,index=list(range(11))),
    "y":6338,
    "z":8212344,
    "c":pd.Series(np.nan,index=list(range(11)))
}
)
dd
```

Out[30]:

	a	b	g	f	h	o	j	y	z	c
0	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
1	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
2	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
3	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
4	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
5	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
6	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
7	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
8	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
9	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
10	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN

3. Display top 7 and last 6 rows and print the output

In [31]:

```
dd.head(7)
```

Out[31]:

	a	b	g	f	h	o	j	y	z	c
0	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
1	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
2	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
3	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
4	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
5	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
6	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN

In [32]:

```
dd.tail(6)
```

Out[32]:

	a	b	g	f	h	o	j	y	z	c
5	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
6	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
7	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
8	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
9	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
10	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN

4. Fill with a constant value and print the output

In [33]:

```
dd.fillna(value=10)
```

Out[33]:

	a	b	g	f	h	o	j	y	z	c
0	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
1	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
2	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
3	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
4	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
5	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
6	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
7	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
8	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
9	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0
10	1.0	2023-07-21	10.0	100	2003	562213	10.0	6338	8212344	10.0

5. Drop the column with missing values and print the output

In [34]:

```
dd.isna()
```

Out[34]:

	a	b	g	f	h	o	j	y	z	c
0	False	False	True	False	False	False	True	False	False	True
1	False	False	True	False	False	False	True	False	False	True
2	False	False	True	False	False	False	True	False	False	True
3	False	False	True	False	False	False	True	False	False	True
4	False	False	True	False	False	False	True	False	False	True
5	False	False	True	False	False	False	True	False	False	True
6	False	False	True	False	False	False	True	False	False	True
7	False	False	True	False	False	False	True	False	False	True
8	False	False	True	False	False	False	True	False	False	True
9	False	False	True	False	False	False	True	False	False	True
10	False	False	True	False	False	False	True	False	False	True

6. Drop the row with missing values and print the output

In [36]:

```
dd.dropna()
```

Out[36]:

a	b	g	f	h	o	j	y	z	c

7. To check the presence of missing values in your dataframe

In [37]:

```
dd.isnull()
```

Out[37]:

	a	b	g	f	h	o	j	y	z	c
0	False	False	True	False	False	False	True	False	False	True
1	False	False	True	False	False	False	True	False	False	True
2	False	False	True	False	False	False	True	False	False	True
3	False	False	True	False	False	False	True	False	False	True
4	False	False	True	False	False	False	True	False	False	True
5	False	False	True	False	False	False	True	False	False	True
6	False	False	True	False	False	False	True	False	False	True
7	False	False	True	False	False	False	True	False	False	True
8	False	False	True	False	False	False	True	False	False	True
9	False	False	True	False	False	False	True	False	False	True
10	False	False	True	False	False	False	True	False	False	True

8.Use operators and check the condition and print the output

In [46]:

```
f = dd[dd['y'] < 80]
f
```

Out[46]:

a	b	g	f	h	o	j	y	z	c

In [47]:

```
f = dd[dd['y'] > 80]
f
```

Out[47]:

	a	b	g	f	h	o	j	y	z	c
0	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
1	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
2	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
3	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
4	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
5	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
6	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
7	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
8	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
9	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
10	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN

9. Display your output using loc and iloc, row and column heading

In [40]:

```
dd.loc[0:2]
```

Out[40]:

	a	b	g	f	h	o	j	y	z	c
0	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
1	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN
2	1.0	2023-07-21	NaN	100	2003	562213	NaN	6338	8212344	NaN

In [43]:

```
dd.iloc[2]
```

Out[43]:

```
a                    1.0
b    2023-07-21 00:00:00
g                    NaN
f                    100
h                   2003
o                 562213
j                    NaN
y                   6338
z                8212344
c                    NaN
Name: 2, dtype: object
```

```
In [44]: dd.iloc[2:5]
```

```
Out[44]:
     a           b    g    f     h       o    j     y        z    c
2  1.0  2023-07-21  NaN  100  2003  562213  NaN  6338  8212344  NaN
3  1.0  2023-07-21  NaN  100  2003  562213  NaN  6338  8212344  NaN
4  1.0  2023-07-21  NaN  100  2003  562213  NaN  6338  8212344  NaN
```

10. Display the statistical summary of data

```
In [42]: dd.describe()
```

```
Out[42]:
          a    g      f       h         o    j       y          z    c
count  11.0  0.0   11.0    11.0      11.0  0.0    11.0       11.0  0.0
mean    1.0  NaN  100.0  2003.0  562213.0  NaN  6338.0  8212344.0  NaN
std     0.0  NaN    0.0     0.0       0.0  NaN     0.0        0.0  NaN
min     1.0  NaN  100.0  2003.0  562213.0  NaN  6338.0  8212344.0  NaN
25%     1.0  NaN  100.0  2003.0  562213.0  NaN  6338.0  8212344.0  NaN
50%     1.0  NaN  100.0  2003.0  562213.0  NaN  6338.0  8212344.0  NaN
75%     1.0  NaN  100.0  2003.0  562213.0  NaN  6338.0  8212344.0  NaN
max     1.0  NaN  100.0  2003.0  562213.0  NaN  6338.0  8212344.0  NaN
```

```
In [ ]:
```