In [1]:

```
import numpy as np
import pandas as pd
```

1. Create any Series and print the output

In [3]:

```
b = pd.Series([1,2,3,4,5])
b
```

Out[3]:

```
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

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

In [24]:

```
data=pd.DataFrame({
   "a": [1,2,3,4,5,6,7,8,9,10],
   "b": [6,7,8,9,0, np.nan,np.nan,1,2,3],
   "c": [1,4,6,8,4,2, np.nan,np.nan,np.nan,21],
   "d": [3,5,6, np.nan,4,1,7,8, np.nan, np.nan],
   "e": [5, np.nan,3,4,5, np.nan, np.nan,np.nan,2,1],
})
data
```

Out[24]:

	a	b	c	d	e
0	1	6.0	1.0	3.0	5.0
1	2	7.0	4.0	5.0	NaN
2	3	8.0	6.0	6.0	3.0
3	4	9.0	8.0	NaN	4.0
4	5	0.0	4.0	4.0	5.0
5	6	NaN	2.0	1.0	NaN
6	7	NaN	NaN	7.0	NaN
7	8	1.0	NaN	8.0	NaN
8	9	2.0	NaN	NaN	2.0
9	10	3.0	21.0	NaN	1.0

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

In [25]:
```
data.head(7)
```

Out[25]:

	a	b	c	d	e
0	1	6.0	1.0	3.0	5.0
1	2	7.0	4.0	5.0	NaN
2	3	8.0	6.0	6.0	3.0
3	4	9.0	8.0	NaN	4.0
4	5	0.0	4.0	4.0	5.0
5	6	NaN	2.0	1.0	NaN
6	7	NaN	NaN	7.0	NaN

In [9]:
```
df.tail(6)
```

Out[9]:

	A	B	C	D	E
4	0.367545	0.451091	0.651559	0.770249	0.943636
5	0.439093	0.945271	0.851614	NaN	0.531055
6	NaN	0.460776	0.778962	0.901202	NaN
7	0.694129	0.708941	NaN	0.684833	0.947279
8	0.437531	0.850078	0.398429	0.473639	NaN
9	NaN	0.433774	0.768066	0.522360	NaN

4. Fill with a constant value and print the output

In [28]:
```
np.isnan(data)
```

Out[28]:

	a	b	c	d	e
0	False	False	False	False	False
1	False	False	False	False	True
2	False	False	False	False	False
3	False	False	False	True	False
4	False	False	False	False	False
5	False	True	False	False	True
6	False	True	True	False	True
7	False	False	True	False	True
8	False	False	True	True	False
9	False	False	False	True	False

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

In [29]:

```
data.dropna(axis=1,how="any")
```

Out[29]:

	a
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10

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

In [30]:

```
data.dropna()
```

Out[30]:

	a	b	c	d	e
0	1	6.0	1.0	3.0	5.0
2	3	8.0	6.0	6.0	3.0
4	5	0.0	4.0	4.0	5.0

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

```
In [31]: data.isna()
```

Out[31]:

	a	b	c	d	e
0	False	False	False	False	False
1	False	False	False	False	True
2	False	False	False	False	False
3	False	False	False	True	False
4	False	False	False	False	False
5	False	True	False	False	True
6	False	True	True	False	True
7	False	False	True	False	True
8	False	False	True	True	False
9	False	False	False	True	False

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

```
In [33]: data[data["a"]>7]
```

Out[33]:

	a	b	c	d	e
7	8	1.0	NaN	8.0	NaN
8	9	2.0	NaN	NaN	2.0
9	10	3.0	21.0	NaN	1.0

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

```
In [34]: data.loc[2:5]
```

Out[34]:

	a	b	c	d	e
2	3	8.0	6.0	6.0	3.0
3	4	9.0	8.0	NaN	4.0
4	5	0.0	4.0	4.0	5.0
5	6	NaN	2.0	1.0	NaN

10. Display the statistical summary of data

In [35]:
```
data.describe()
```

Out[35]:
```
               a         b          c         d         e
count   10.00000  8.000000   7.000000  7.000000  6.000000
mean     5.50000  4.500000   6.571429  4.857143  3.333333
std      3.02765  3.422614   6.778819  2.410295  1.632993
min      1.00000  0.000000   1.000000  1.000000  1.000000
25%      3.25000  1.750000   3.000000  3.500000  2.250000
50%      5.50000  4.500000   4.000000  5.000000  3.500000
75%      7.75000  7.250000   7.000000  6.500000  4.750000
max     10.00000  9.000000  21.000000  8.000000  5.000000
```

In [ ]: