In [1]:
```
import numpy as np
```

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

1.Sries and print

In [3]:
```
a=pd.Series([7,8,9,10])
a
```

Out[3]:
```
0     7
1     8
2     9
3    10
dtype: int64
```

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

In [6]:
```
a=pd.DataFrame(np.random.randn(10,5))
a
```

Out[6]:

	0	1	2	3	4
0	-0.745132	1.167885	-0.061078	0.834838	-2.943751
1	1.263154	-0.237109	-1.113000	1.390301	0.192975
2	0.332518	-0.630479	-0.057223	0.121989	-0.644276
3	0.736260	0.541631	-0.320803	-0.275841	-0.905967
4	-0.430364	0.180501	-0.421489	0.487178	0.668544
5	-0.499357	-0.159479	-0.813900	-0.311887	2.296102
6	1.554240	0.867740	0.095968	1.311170	0.304391
7	0.436374	-1.158938	-0.292872	0.019722	-0.728985
8	-0.970953	0.829894	0.993591	-1.205190	2.468500
9	-0.225483	-0.647058	0.149899	-0.447804	-0.199471

3.Display top 7 and last 6 rows and print

In [19]:
```
a.head(7)
```

Out[19]:

	0	1	2	3	4
0	-0.745132	1.167885	-0.061078	0.834838	-2.943751
1	1.263154	-0.237109	-1.113000	1.390301	0.192975
2	0.332518	-0.630479	-0.057223	0.121989	-0.644276
3	0.736260	0.541631	-0.320803	-0.275841	-0.905967
4	-0.430364	0.180501	-0.421489	0.487178	0.668544
5	-0.499357	-0.159479	-0.813900	-0.311887	2.296102
6	1.554240	0.867740	0.095968	1.311170	0.304391

In [18]:
```
a.tail(6)
```

Out[18]:

	0	1	2	3	4
4	-0.430364	0.180501	-0.421489	0.487178	0.668544
5	-0.499357	-0.159479	-0.813900	-0.311887	2.296102
6	1.554240	0.867740	0.095968	1.311170	0.304391
7	0.436374	-1.158938	-0.292872	0.019722	-0.728985
8	-0.970953	0.829894	0.993591	-1.205190	2.468500
9	-0.225483	-0.647058	0.149899	-0.447804	-0.199471

```
4.Fill with a constant value and print
```

In [52]:
```
df=pd.DataFrame(
{'a':[100, 90, np.nan, 95],
 'b': [30, 45, 56, np.nan],
 'c':[np.nan, 40, 80, 98]
})
df.fillna(value=0)
```

Out[52]:

	a	b	c
0	100.0	30.0	0.0
1	90.0	45.0	40.0
2	0.0	56.0	80.0
3	95.0	0.0	98.0

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

In [62]:
```
df=pd.DataFrame(
{'a':[100, 90, np.nan, 95],
 'b': [30, 45, 56, np.nan],
})
df.dropna()
```

Out[62]:

	a	b
0	100.0	30.0
1	90.0	45.0

6.Drop the row with missing values and print

In [54]:
```
df=pd.DataFrame(
{'a':[100, 90, np.nan, 95],
 'b': [30, 45, 56, np.nan],
 'c':[np.nan, 40, 80, 98]
})
df.dropna()
```

Out[54]:

	a	b	c
1	90.0	45.0	40.0

```
7.Check the presence of missing values in your dataframe
```

In [42]:
```
i = {'First Score':[100, 90, np.nan, 95],
        'Second Score': [30, 45, 56, np.nan],
        'Third Score':[np.nan, 40, 80, 98]}
df = pd.DataFrame(i)
df.isnull()
df
```

Out[42]:

	First Score	Second Score	Third Score
0	100.0	30.0	NaN
1	90.0	45.0	40.0
2	NaN	56.0	80.0
3	95.0	NaN	98.0

```
8.Display loc and iloc
```

In [64]:
```
df=pd.DataFrame(
{'a':[100, 90, np.nan, 95],
 'b': [30, 45, 56, np.nan],
 'c':[np.nan, 40, 80, 98]
})
df.loc[0:2]
```

Out[64]:

	a	b	c
0	100.0	30.0	NaN
1	90.0	45.0	40.0
2	NaN	56.0	80.0

In [65]:
```
df.iloc[0:2]
```

Out[65]:

	a	b	c
0	100.0	30.0	NaN
1	90.0	45.0	40.0

```
9.Use operators and check loc and iloc,row and column heading
```

In [67]:
```
df=pd.DataFrame(
{'a':[100, 90, np.nan, 95],
 'b': [30, 45, 56, np.nan],
 'c':[np.nan, 40, 80, 98]
})
df[df["a"]>2]
```

Out[67]:

	a	b	c
0	100.0	30.0	NaN
1	90.0	45.0	40.0
3	95.0	NaN	98.0

In [68]:
```
df=pd.DataFrame(
{'a':[100, 90, np.nan, 95],
 'b': [30, 45, 56, np.nan],
 'c':[np.nan, 40, 80, 98]
})
df.describe()
```

Out[68]:

	a	b	c
count	3.0	3.000000	3.000000
mean	95.0	43.666667	72.666667
std	5.0	13.051181	29.687259
min	90.0	30.000000	40.000000
25%	92.5	37.500000	60.000000
50%	95.0	45.000000	80.000000
75%	97.5	50.500000	89.000000
max	100.0	56.000000	98.000000

In [ ]:
```

```