In [18]:

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

1. Create any Series and print the output

In [7]:

```
a=pd.Series([10,54,32,59,43])
print(a)
```

```
0    10
1    54
2    32
3    59
4    43
dtype: int64
```

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

In [27]:

```
b=pd.DataFrame(
{
    "A":86,
    "B":34,
    "C":pd.Series(np.nan,index=list(range(10))),
    "D":12,
    "E":100,
})
b
```

Out[27]:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **0** | 86 | 34 | NaN | 12 | 100 |
| **1** | 86 | 34 | NaN | 12 | 100 |
| **2** | 86 | 34 | NaN | 12 | 100 |
| **3** | 86 | 34 | NaN | 12 | 100 |
| **4** | 86 | 34 | NaN | 12 | 100 |
| **5** | 86 | 34 | NaN | 12 | 100 |
| **6** | 86 | 34 | NaN | 12 | 100 |
| **7** | 86 | 34 | NaN | 12 | 100 |
| **8** | 86 | 34 | NaN | 12 | 100 |
| **9** | 86 | 34 | NaN | 12 | 100 |

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

In [41]:

```
print(b.head(7))
print(b.tail(6))
```

```
    A   B   C   D    E
0  86  34 NaN  12  100
1  86  34 NaN  12  100
2  86  34 NaN  12  100
3  86  34 NaN  12  100
4  86  34 NaN  12  100
5  86  34 NaN  12  100
6  86  34 NaN  12  100
    A   B   C   D    E
4  86  34 NaN  12  100
5  86  34 NaN  12  100
6  86  34 NaN  12  100
7  86  34 NaN  12  100
8  86  34 NaN  12  100
9  86  34 NaN  12  100
```

4. Fill with a constant value and print the output

In [37]:

```
b.fillna(value=8)
```

Out[37]:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **0** | 86 | 34 | 8.0 | 12 | 100 |
| **1** | 86 | 34 | 8.0 | 12 | 100 |
| **2** | 86 | 34 | 8.0 | 12 | 100 |
| **3** | 86 | 34 | 8.0 | 12 | 100 |
| **4** | 86 | 34 | 8.0 | 12 | 100 |
| **5** | 86 | 34 | 8.0 | 12 | 100 |
| **6** | 86 | 34 | 8.0 | 12 | 100 |
| **7** | 86 | 34 | 8.0 | 12 | 100 |
| **8** | 86 | 34 | 8.0 | 12 | 100 |
| **9** | 86 | 34 | 8.0 | 12 | 100 |

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

In [43]:

```
b.dropna(axis=1)
```

Out[43]:

| | A | B | D | E |
|---|---|---|---|---|
| **0** | 86 | 34 | 12 | 100 |
| **1** | 86 | 34 | 12 | 100 |
| **2** | 86 | 34 | 12 | 100 |
| **3** | 86 | 34 | 12 | 100 |
| **4** | 86 | 34 | 12 | 100 |
| **5** | 86 | 34 | 12 | 100 |
| **6** | 86 | 34 | 12 | 100 |
| **7** | 86 | 34 | 12 | 100 |
| **8** | 86 | 34 | 12 | 100 |
| **9** | 86 | 34 | 12 | 100 |

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

In [44]:

```
b.dropna()
```

Out[44]:

| | A | B | C | D | E |
|---|---|---|---|---|---|

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

In [45]:

```
b.isna()
```

Out[45]:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **0** | False | False | True | False | False |
| **1** | False | False | True | False | False |
| **2** | False | False | True | False | False |
| **3** | False | False | True | False | False |
| **4** | False | False | True | False | False |
| **5** | False | False | True | False | False |
| **6** | False | False | True | False | False |
| **7** | False | False | True | False | False |
| **8** | False | False | True | False | False |
| **9** | False | False | True | False | False |

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

In [69]:

```
d=np.arange(1,16)
e=d[(d>6)&(d<13)]
print(e)
```

```
[ 7  8  9 10 11 12]
```

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

In [65]:

```
print(b.loc["A":"D"])
print(b.iloc[0:5])
```

```
Empty DataFrame
Columns: [A, B, C, D, E]
Index: []
    A   B   C   D    E
0  86  34 NaN  12  100
1  86  34 NaN  12  100
2  86  34 NaN  12  100
3  86  34 NaN  12  100
4  86  34 NaN  12  100
```

10. Display the statistical summary of data

In [57]:

```
b.describe()
```

Out[57]:

| | A | B | C | D | E |
|---|---|---|---|---|---|
| **count** | 10.0 | 10.0 | 0.0 | 10.0 | 10.0 |
| **mean** | 86.0 | 34.0 | NaN | 12.0 | 100.0 |
| **std** | 0.0 | 0.0 | NaN | 0.0 | 0.0 |
| **min** | 86.0 | 34.0 | NaN | 12.0 | 100.0 |
| **25%** | 86.0 | 34.0 | NaN | 12.0 | 100.0 |
| **50%** | 86.0 | 34.0 | NaN | 12.0 | 100.0 |
| **75%** | 86.0 | 34.0 | NaN | 12.0 | 100.0 |
| **max** | 86.0 | 34.0 | NaN | 12.0 | 100.0 |

In [ ]:

```

```