21/07/2023

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

Create any Series and print the output

In [2]:
```
s = pd.Series([1, 2, 3, np.nan, 4, 5])
print(s)
```

```
0    1.0
1    2.0
2    3.0
3    NaN
4    4.0
5    5.0
dtype: float64
```

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

In [3]:
```
df = pd.DataFrame({
    "col1": [1, 2, 3, np.nan, 4, 5],
    "col2": [6, 7, 8, 9, 10, 11],
    "col3": [12, 13, 14, 15, 16, 17],
    "col4": [18, 19, 20, 21, 22, 23],
    "col5": [24, 25, 26, 27, 28, 29]
})
print(df)
```

```
   col1  col2  col3  col4  col5
0   1.0     6    12    18    24
1   2.0     7    13    19    25
2   3.0     8    14    20    26
3   NaN     9    15    21    27
4   4.0    10    16    22    28
5   5.0    11    17    23    29
```

Display top 7 and last 6 rows and print the output

In [4]:
```
print(df.head(7))
print(df.tail(6))
```

```
   col1  col2  col3  col4  col5
0   1.0     6    12    18    24
1   2.0     7    13    19    25
2   3.0     8    14    20    26
3   NaN     9    15    21    27
4   4.0    10    16    22    28
5   5.0    11    17    23    29
   col1  col2  col3  col4  col5
0   1.0     6    12    18    24
1   2.0     7    13    19    25
2   3.0     8    14    20    26
3   NaN     9    15    21    27
4   4.0    10    16    22    28
5   5.0    11    17    23    29
```

Fill with a constant value and print the output

In [5]:
```
df.fillna(0, inplace=True)
print(df)
```

```
   col1  col2  col3  col4  col5
0   1.0     6    12    18    24
1   2.0     7    13    19    25
2   3.0     8    14    20    26
3   0.0     9    15    21    27
4   4.0    10    16    22    28
5   5.0    11    17    23    29
```

Drop the column with missing values and print the output

In [6]:
```
df = df.dropna(axis=1)
print(df)
```

```
   col1  col2  col3  col4  col5
0   1.0     6    12    18    24
1   2.0     7    13    19    25
2   3.0     8    14    20    26
3   0.0     9    15    21    27
4   4.0    10    16    22    28
5   5.0    11    17    23    29
```

To check the presence of missing values in your dataframe

In [7]:
```
print(df.isnull().any())
```

```
col1    False
col2    False
col3    False
col4    False
col5    False
dtype: bool
```

Use operators and check the condition and print the output

In [8]:
```
print(df[df["col1"] > 10])
```

```
Empty DataFrame
Columns: [col1, col2, col3, col4, col5]
Index: []
```

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

In [9]:
```
print(df.loc[0, "col1"])
print(df.iloc[0, 0])
```

```
1.0
1.0
```

Display the statistical summary of data

In [10]:
```
print(df.describe())
```

```
           col1       col2       col3       col4       col5
count  6.000000   6.000000   6.000000   6.000000   6.000000
mean   2.500000   8.500000  14.500000  20.500000  26.500000
std    1.870829   1.870829   1.870829   1.870829   1.870829
min    0.000000   6.000000  12.000000  18.000000  24.000000
25%    1.250000   7.250000  13.250000  19.250000  25.250000
50%    2.500000   8.500000  14.500000  20.500000  26.500000
75%    3.750000   9.750000  15.750000  21.750000  27.750000
max    5.000000  11.000000  17.000000  23.000000  29.000000
```

In [ ]: