

DAY 03

Create any series and print the output?

```
In [1]: def arithmetic_series(a1, d, n):  
        series = [a1 + (i - 1) * d for i in range(1, n + 1)]  
        return series  
first_term = 5  
common_difference = 3  
number_of_terms = 10  
  
c = arithmetic_series(first_term, common_difference, number_of_terms)  
print("Arithmetic Series:")  
print(c)
```

Arithmetic Series:
[5, 8, 11, 14, 17, 20, 23, 26, 29, 32]

Create a dataframe of 10*5 with few nan values and print the output?

```
In [3]: import pandas as pd  
import numpy as np  
data = {  
    'A': np.random.randint(1, 100, 10),  
    'B': np.random.randint(1, 100, 10),  
    'C': np.random.randint(1, 100, 10),  
    'D': np.random.randint(1, 100, 10),  
    'E': [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.n  
}  
  
df = pd.DataFrame(data)  
print(df)
```

	A	B	C	D	E
0	65	4	73	99	NaN
1	14	25	91	39	NaN
2	16	62	36	97	NaN
3	16	69	3	90	NaN
4	60	5	87	4	NaN
5	77	40	95	98	NaN
6	22	81	67	74	NaN
7	38	30	61	29	NaN
8	53	79	15	8	NaN
9	56	60	37	98	NaN

Display top 7 and last 6 rows and print the output

```
In [4]: import pandas as pd
import numpy as np
data = {
    'A': np.random.randint(1, 100, 10),
    'B': np.random.randint(1, 100, 10),
    'C': np.random.randint(1, 100, 10),
    'D': np.random.randint(1, 100, 10),
    'E': [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]
}

df = pd.DataFrame(data)
print("Top 7 Rows:")
print(df.head(7))
print("\nLast 6 Rows:")
print(df.tail(6))
```

Top 7 Rows:

	A	B	C	D	E
0	41	99	29	89	NaN
1	5	39	63	15	NaN
2	72	49	28	27	NaN
3	1	80	22	12	NaN
4	44	90	62	86	NaN
5	26	74	8	69	NaN
6	33	95	53	24	NaN

Last 6 Rows:

	A	B	C	D	E
4	44	90	62	86	NaN
5	26	74	8	69	NaN
6	33	95	53	24	NaN
7	76	77	93	88	NaN
8	33	30	8	20	NaN
9	53	68	54	30	NaN

Drop the column with missing values and print the output?

```
In [5]: import pandas as pd
import numpy as np
data = {
    'A': np.random.randint(1, 100, 10),
    'B': np.random.randint(1, 100, 10),
    'C': np.random.randint(1, 100, 10),
    'D': np.random.randint(1, 100, 10),
    'E': [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]
}

df = pd.DataFrame(data)
df_dropped = df.dropna(axis=1)
print(df_dropped)
```

	A	B	C	D
0	12	68	16	3
1	8	21	2	96
2	8	91	72	19
3	29	38	66	97
4	56	87	56	98
5	85	11	71	51
6	22	39	28	30
7	97	16	33	68
8	49	19	59	61
9	89	75	45	6

Drop the row with missing values and print the output using python?

```
In [6]: import pandas as pd
import numpy as np
data = {
    'A': np.random.randint(1, 100, 10),
    'B': np.random.randint(1, 100, 10),
    'C': np.random.randint(1, 100, 10),
    'D': np.random.randint(1, 100, 10),
    'E': [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]
}

df = pd.DataFrame(data)
df_dropped = df.dropna()
print(df_dropped)
```

```
Empty DataFrame
Columns: [A, B, C, D, E]
Index: []
```

To check the presence of the missing values in your dataframe?

```
In [7]: import pandas as pd
import numpy as np
data = {
    'A': np.random.randint(1, 100, 10),
    'B': np.random.randint(1, 100, 10),
    'C': np.random.randint(1, 100, 10),
    'D': np.random.randint(1, 100, 10),
    'E': [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]
}

df = pd.DataFrame(data)
missing_values = df.isnull()
print(missing_values)
```

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

Use operators and check the condition and print the output?

```
In [8]: import pandas as pd
import numpy as np
data = {
    'A': np.random.randint(1, 10, 3),
    'B': np.random.randint(1, 10, 3),
    'C': np.random.randint(1, 10, 3),
}

df = pd.DataFrame(data)
threshold = 5
condition_result = df > threshold
print("Original DataFrame:")
print(df)
print("\nCondition Result (Greater than {}):".format(threshold))
print(condition_result)
```

Original DataFrame:

	A	B	C
0	1	1	5
1	1	8	3
2	2	5	7

Condition Result (Greater than 5):

	A	B	C
0	False	False	False
1	False	True	False
2	False	False	True

Display the output using loc and iloc,row and column heading?

```
In [11]: import pandas as pd
import numpy as np
data = {
    'A': np.random.randint(1, 10, 3),
    'B': np.random.randint(1, 10, 3),
    'C': np.random.randint(1, 10, 3),
}

df = pd.DataFrame(data)
print("Original DataFrame:")
print(df)
print("\nUsing loc:")
print("Value at row 1 and column 'B':", df.loc[1, 'B'])
print("Values in row 2:", df.loc[2])
print("Values in column 'C':", df.loc[:, 'C'])
print("\nUsing iloc:")
print("Value at row 1 and column 1:", df.iloc[1, 1])
print("Values in row 2:", df.iloc[2])
print("Values in column 3:", df.iloc[:, 2])
```

Original DataFrame:

	A	B	C
0	6	6	5
1	5	7	6
2	8	2	1

Using loc:

Value at row 1 and column 'B': 7

Values in row 2: A 8

B 2

C 1

Name: 2, dtype: int32

Values in column 'C': 0 5

1 6

2 1

Name: C, dtype: int32

Using iloc:

Value at row 1 and column 1: 7

Values in row 2: A 8

B 2

C 1

Name: 2, dtype: int32

Values in column 3: 0 5

1 6

2 1

Name: C, dtype: int32

In []:

