S.NO	TOPIC NAME	DATE	SIGN
1a	**Python code to print profile**	**10/7/25**	
1b	**Python code to print adition of two numbers**	**10/7/25**	
1c	**Python code to print square root of number**	**10/7/25**	
1d	**Python code to calculate area of triangle**	**10/7/25**	
1e	**Python code to swap two variables**	**17/7/25**	
2a	**Python code to create nested tuples**	**17/7/25**	
2b	**Python code to sort nested tuples**	**17/7/25**	
2c	**Python code to copy or clone list**	**17/7/25**	
2d	**Python code to check immutability property of python tuples**	**24/7/25**	
3a	**Python code to create variable and sort text**	**24/7/25**	
3b	**Python code to retrieve data from html file**	**24/7/25**	
3c	**Python code to print current date in different format**	**24/7/25**	
3d	**Python code to convert time stamp to date stamp**	**31/7/25**	
3e	**Python code to develop calendar module**	**31/7/25**	
3f	**Python code to compare two dates**	**31/7/25**	
4a	**Python code to create numpy array**	**31/7/25**	
4b	**Python code todemonstrate operations on single array**	**7/8/25**	
4c	python code to Create array with 10 elements and slice 1st to 5th element	**7/8/25**	
4d	**Python code to sort array alphabatically**	**7/8/25**	
4e	**Python code to create a filter array**	**7/8/25**	
5a	**Python code to create data frame object**	**14/8/25**	
5b	**Python code to show statistical information on data set**	**14/8/25**	
5c	**Python code to create panda series from dictionaries**	**21/8/25**	
5d	**Python code to filter panda series**	**21/8/25**	

PRACTICAL NO 1A

Aim:-Write a python code to print your profile

```
print("1a. Output Start")
profile = {
"name": "Your Name",
"age": 20,
"location": "India",
"languages": ["Python", "Java"],
}
print("Profile:", profile)
print("1a. Output End\n")
```

OUTPUT:-

```
Output Start
Profile: {'name': 'Your Name', 'age': 20, 'location': 'India', 'languages': ['Python', 'Java']}
1a. Output End
```

PRACTICAL NO 1B

Aim:-Write a python code to print addition of two numbers.

```
print("1b. Output Start")

a, b = 5, 7

result = a + b

print("Addition of", a, "and", b, "is", result)

print("1b. Output End\n")
```

```
1b. Output Start

Addition of 5 and 7 is 12

1b. Output End
```

PRACTICAL NO 1C

Aim:-Write a python code to print square root of two numbers.

```
print("1c. Output Start")

import math

num = 16

sqrt_num = math.sqrt(num)

print("Square root of", num, "is", sqrt_num)

print("1c. Output End\n")
```

```
1c. Output Start

Square root of 16 is 4.0

1c. Output End
```

PRACTICAL NO 1D

Aim:-Write a python code to print area of triangle.

```
print("1d. Output Start")

base = 10

height = 5

area = 0.5 * base * height

print("Area of triangle with base", base, "and height", height, "is", area)

print("1d. Output End\n")
```

```
1d. Output Start

Area of triangle with base 10 and height 5 is 25.0

1d. Output End
```

PRACTICAL NO 1E

Aim:-Write a python code to swap two variables.

```
print("1d. Output Start")

x, y = 3, 8

print("Before swap: x =", x, ", y =", y)

x, y = y, x

print("After swap: x =", x, ", y =", y)

print("1d. Output End\n")

# 2a. Create nested tuples

print("2a. Output Start")

nested_tuple = (1, (2, 3), (4, (5, 6)))

print("Nested tuple:", nested_tuple)

print("2a. Output End\n")
```

```
1e. Output Start

Before swap: x = 3 , y = 8

After swap: x = 8 , y = 3

1e. Output End
```

PRACTICAL NO 2A

Aim:-Write a python code to create nested tuples.

```
print("2a. Output Start")

nested_tuple = (1, (2, 3), (4, (5, 6)))

print("Nested tuple:", nested_tuple)

print("2a. Output End\n")
```

```
2a. Output Start

Nested tuple: (1, (2, 3), (4, (5, 6)))

2a. Output End
```

PRACTICAL NO 2B

Aim:-Write a python code to SORT nested tuples.

```
print("2b. Output Start")

nested = ((3, 4), (1, 2), (5, 0))

sorted_nested = tuple(sorted(nested))

print("Sorted nested tuple:", sorted_nested)

print("2b. Output End\n")
```

```
2b. Output Start

Sorted nested tuple: ((1, 2), (3, 4), (5, 0))

2b. Output End
```

PRACTICAL NO 2C

Aim:-Write a python code to copy or clone list.

```
print("2c. Output Start")

original_list = [1, 2, 3]

cloned_list = original_list[:]

print("Original list:", original_list)

print("Cloned list:", cloned_list)

print("2c. Output End\n")
```

```
2c. Output Start

Original list: [1, 2, 3]

Cloned list: [1, 2, 3]

2c. Output End
```

PRACTICAL NO 2D

Aim:-Write python code to Check immutability property of python tuples

```
print("2d. Output Start")

t = (1, 2, 3)

try:

   t[0] = 10

except TypeError as e:

   print("Tuples are immutable:", e)

print("2d. Output End\n")
```

```
2d. Output Start

Tuples are immutable: 'tuple' object does not support item assignment

2d. Output End
```

PRACTICAL NO 3 A

Aim:-Write a python program to Create a variable and store text to search

```
print("3a. Output Start")

search_text = "Python is awesome"

print("Text to search:", search_text)

print("3a. Output End\n")
```

```
3a. Output Start

Text to search: Python is awesome

3a. Output End
```

PRACTICAL NO 3 B

Aim:-Write a python program to Retrieve data from HTML file

```
print("3b. Output Start")

from bs4 import BeautifulSoup

html = "<html><body><h1>Hello</h1><p>World</p></body></html>"

soup = BeautifulSoup(html, "html.parser")

print("HTML Title:", soup.h1.text)

print("HTML Paragraph:", soup.p.text)

print("3b. Output End\n")
```

```
3b. Output Start

HTML Title: Hello

HTML Paragraph: World

3b. Output End
```

PRACTICAL NO 3 C

Aim:- Write a python program to Print current date in different format

```
print("3c. Output Start")
from datetime import datetime
now = datetime.now()
print("YYYY-MM-DD:", now.strftime("%Y-%m-%d"))
print("DD/MM/YYYY:", now.strftime("%d/%m/%Y"))
print("Month Day, Year:", now.strftime("%B %d, %Y"))
print("3c. Output End\n")
```

```
3c. Output Start
YYYY-MM-DD: 2025-09-24
DD/MM/YYYY: 24/09/2025
Month Day, Year: September 24, 2025
```

```
3c. Output End
```

PRACTICAL NO 3 D

Aim:-Write a python program to Convert timestamp to datestamp

```
print("3d. Output Start")

timestamp = 1700000000

date_from_timestamp = datetime.fromtimestamp(timestamp)

print("Date from timestamp:", date_from_timestamp.strftime("%Y-%m-%d %H:%M:%S"))

print("3d. Output End\n")
```

```
3d. Output Start

Date from timestamp: 2023-11-15 03:43:20

3d. Output End
```

PRACTICAL NO 3 E

Aim:- Write a python program to Develop calendar module

```
print("3e. Output Start")

import calendar

year = 2025

month = 9

print("Calendar for", month, year)

print(calendar.month(year, month))

print("3e. Output End\n")
```

```
3e. Output Start

Calendar for 9 2025

   September 2025

Mo Tu We Th Fr Sa Su

 1  2  3  4  5  6  7

 8  9 10 11 12 13 14

15 16 17 18 19 20 21

22 23 24 25 26 27 28

29 30
```

PRACTICAL NO 3 F

Aim:-Write a python program to Compare two dates

```
print("3f. Output Start")
from datetime import datetime

date1 = datetime(2025, 9, 24)
date2 = datetime(2024, 12, 31)

if date1 > date2:
    print(f"{date1.strftime('%Y-%m-%d')} is after {date2.strftime('%Y-%m-%d')}")
elif date1 < date2:
    print(f"{date1.strftime('%Y-%m-%d')} is before {date2.strftime('%Y-%m-%d')}")
else:
    print(f"{date1.strftime('%Y-%m-%d')} is the same as {date2.strftime('%Y-%m-%d')}")
print("3f. Output End\n")
```

```
3f. Output Start
2025-09-24 is after 2024-12-31
```

3f. Output End

PRACTICAL NO 4A

Aim:- Write a python code to print Create Numpy Array

```
print("4a. Output Start")

import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print("Numpy Array:", arr)

print("4a. Output End\n")
```

PRACTICAL NO 4B

Aim:-Write python code to demonstrate Basic operations on single array

```
print("4b. Output Start")

arr = np.array([10, 20, 30])

print("Original Array:", arr)

print("Array + 5:", arr + 5)

print("Array * 2:", arr * 2)

print("Sum of Array:", arr.sum())

print("Mean of Array:", arr.mean())

print("4b. Output End\n")
```

PRACTICAL NO 4C

Aim:-Write python code to Create array with 10 elements and slice 1st to 5th element

```
print("4c. Output Start")

arr = np.arange(1, 11)

print("Array:", arr)

print("Slice 1st to 5th element:", arr[0:5])

print("4c. Output End\n")
```

PRACTICAL NO 4D

Aim :-Write python code to Sort an array alphabetically

```
print("4d. Output Start")

arr = np.array(['banana', 'apple', 'cherry', 'date'])

sorted_arr = np.sort(arr)

print("Original Array:", arr)

print("Sorted Array:", sorted_arr)

print("4d. Output End\n")
```

PRACTICAL NO 4E

Aim:-Write python code to Create a filter array that will return maximum values from an array

```
print("4e. Output Start")

arr = np.array([10, 50, 30, 50, 20])

max_val = arr.max()

filter_arr = arr == max_val

print("Original Array:", arr)

print("Maximum Value:", max_val)

print("Filter Array:", filter_arr)

print("Maximum Values:", arr[filter_arr])

print("4e. Output End\n")
```

PRACTICAL NO 5A

Aim :-Write python code to Import pandas and create DataFrame object

```
print("5a. Output Start")

import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Age': [25, 30, 22]}

df = pd.DataFrame(data)

print("DataFrame:\n", df)

print("5a. Output End\n")
```

PRACTICAL NO 5B

Aim:-Write python code to Show statistical information on given data set

```
print("5b. Output Start")

print("Statistical Info:\n", df.describe())

print("5b. Output End\n")
```

PRACTICAL NO 5C

Aim:-Write python code to Create pandas series from dictionaries

```
print("5c. Output Start")

dict_data = {'a': 100, 'b': 200, 'c': 300}

series = pd.Series(dict_data)

print("Pandas Series:\n", series)

print("5c. Output End\n")
```

PRACTICAL NO 5D

Aim:-Write python code to Filter pandas series with Boolean arrays

```
print("5d. Output Start")

bool_filter = series > 150

print("Boolean Filter:", bool_filter.values)

filtered_series = series[bool_filter]

print("Filtered Series:\n", filtered_series)

print("5d. Output End\n")
```