

Aim:- Write a Python code to print your profile using a dictionary.

Program:-

~~# Program to Print user Profile using a dictionary.~~

~~Profile = {~~

~~"Name": "Khan Yasmeen Warsi",~~

~~"Age": 19,~~

~~"Qualification": "Second Year BSC. Computer Science",~~

~~"College": "ABC college of science"~~

~~}~~

~~Print ("---- Profile ----")~~

~~for key, value in Profile.items ():~~

~~Print (f"\{key\}: \{value\}")~~

```
✓ [1] profile = {  
06     "Name": "KHAN YASMEEN WARSI ",  
      "Age": 19,  
      "Qualification": "Second Year B.Sc. Computer Sci  
      "College": "ABC College of Science"  
    }  
    print("----- Profile -----")  
    for key, value in profile.items():  
        print(f"{key}: {value}")  
  
→ ----- Profile -----  
Name: KHAN YASMEEN WARSI  
Age: 19  
Qualification: Second Year B.Sc. Computer Sci  
College: ABC College of Science
```

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

~~Program:-~~

~~# Program to add two numbers~~

~~num1 = 15~~

~~num2 = 25~~

~~sum = num1 + num2~~

~~print("The sum of ", num1, " and ", num2, " is
sum)~~

```
✓  ⏎ # Program to add two numbers
  0s      num1 = 15
          num2 = 25
          sum = num1 + num2
          print("KHAN YASMEEN WARSI :")
          print("The sum of", num1, "and", num2, "is", sum)

→  KHAN YASMEEN WARSI :
    The sum of 15 and 25 is 40
```

Aim :- write a Python code to Print square root of a number.

Program:-

~~# Program to find square root of a number~~

~~import math~~

~~number = 36~~

~~Squrt = math.sqrt(number)~~

~~Print ("The square root of ", number, "is", Squrt)~~

```
✓ 0s  # Program to find square root of a number
      import math
      number = 36
      sqrt = math.sqrt(number)
      print("KHAN YASMEEN WARSI :")
      print("The square root of", number, "is", sqrt)

→ KHAN YASMEEN WARSI :
The square root of 36 is 6.0
```

Aim:- Write a Python code to calculate area of a triangle.

Programme :-

~~# Program to calculate area of triangle.~~

$$\text{Base} = 12$$

$$\text{height} = 6$$

$$\text{area} = 0.5 * \text{base} * \text{height}$$

Print ("The area of the triangle is", area)

```
✓ [3] # Program to calculate area of a triangle  
os   base = 12  
      height = 6  
      area = 0.5 * base * height  
      print("KHAN YASMEEN WARSI :")  
      print("The area of the triangle is", area)
```

```
→ KHAN YASMEEN WARSI :  
    The area of the triangle is 36.0
```

Aim :- write a Python code to swap two variable

Program :-

Program to swap two variables.

~~a=8~~

~~b=10~~

Print ("Before swapping : a = ", a, ", b = ", b)

swapping

a, b = b, a

Print ("After swapping : a = ", a, ", b = ", b)

```
✓ 08 # Program to swap two variables
    a = 5
    b = 10
    print("KHAN YASMEEN WARSI :")
    print("Before swapping: a =", a, ", b =", b)
    # Swapping
    a, b = b, a
    print("After swapping: a =", a, ", b =", b)
```

```
→ KHAN YASMEEN WARSI :
Before swapping: a = 5 , b = 10
After swapping: a = 10 , b = 5
```

A Aim:- Write a Python code to create nested tuples.

Program

```
nested_tuple = (( "apple", "banana"), (1, 2, 3), ("x", "y", "z",  
"NUM1"))
```

```
Print ("Nested Tuple : ", nested_tuple)
```

By write a Python code to sort the nested tuple using sorted() function.

Program

```
nested_tuple = ((3, 4), (1, 2), (0, 5))
```

```
sorted_tuple = tuple(sorted(nested_tuple))
```

```
Print ("Original Tuple : ", nested_tuple)
```

```
Print ("Sorted Tuple : ", sorted_tuple)
```

C) Program to copy or clone a list.

Program

```
original_list = [1, 2, 3, 4]
```

```
cloned_list = original_list.copy()
```

```
Print ("Original List : ", original_list)
```

```
Print ("Cloned List : ", cloned_list)
```

```
[2] # Program to create nested tuples  
nested_tuple = ("apple", "banana"), (1, 2, 3), ("x", "y", "z", "NUM1")  
print("KHAN YASMEEN WARSI :")  
print("Nested Tuple:", nested_tuple)  
→ KHAN YASMEEN WARSI :  
Nested Tuple: (('apple', 'banana'), (1, 2, 3), ('x', 'y', 'z', 'NUM1'))
```

Output → 2.A

```
[3] # Program to sort nested tuple using sorted()  
nested_tuple = ((3, 4), (1, 2), (0, 5))  
sorted_tuple = tuple(sorted(nested_tuple))  
print("KHAN YASMEEN WARSI :")  
print("Original Tuple:", nested_tuple)  
print("Sorted Tuple:", sorted_tuple)  
→ KHAN YASMEEN WARSI :  
Original Tuple: ((3, 4), (1, 2), (0, 5))  
Sorted Tuple: ((0, 5), (1, 2), (3, 4))
```

Output → 2.B

```
[4] # Program to copy or clone a list  
original_list = [1, 2, 3, 4]  
cloned_list = original_list.copy()  
print("KHAN YASMEEN WARSI :")  
print("Original List:", original_list)  
print("Cloned List:", cloned_list)  
→ KHAN YASMEEN WARSI :  
Original List: [1, 2, 3, 4]  
Cloned List: [1, 2, 3, 4]
```

Output → 2.C

Q7 write a Python code to check immutability property of Python tuples.

Program.

```
Sample-tuple = (10, 20, 30)
print ("Original tuple : ", Sample-tuple)

try:
    print (sample-tuple [2])
    sample-tuple [0] = 100
except TypeError as e:
    print ("Error : ", e)
    print ("Tuples are immutable")
```



```
✓ [11] # Program to check immutability of tuples
$ sample_tuple = (10, 20, 30)
print("Original Tuple:", sample_tuple)
try:
    print(sample_tuple[2])
    sample_tuple[0] = 100
except TypeError as e:
    # These lines should be indented under the except block
    print("Error:", e)
    print("Tuples are immutable.")

→ Original Tuple: (10, 20, 30)
30
Error: 'tuple' object does not support item assignment
Tuples are immutable.
```

Output 2.1

A) Aim :- write a Python code for creating a variable & storing the text that we want to search

Practical 3a

```
# Storing a sample sentence in a variable
Search_text = "Data Science involves statistics, Programming and machine learning."
Print("Text to search from", Search_text)
```

By Aim :- write a Python code to retrieve data from HTML file.

Program:-

```
from bs4 import BeautifulSoup
# Sample HTML element content
html_data = """
<html>
<head> <title> welcome Page </title> </head>
<body>
<h1> Hello Students ! </h1>
<p> This is a Data Science Practical session </p>
</body>
</html>
"""
```

creating a BeautifulSoup object
 soup = BeautifulSoup(html_data, 'html.parser')

```
[3] # Storing a sample sentence in a variable
search_text = "Data Science involves statistics, programming, and machine learning."
print("KHAN YASMEEN WARI SYIT 21:")
print("Text to search from:", search_text)
```

☞ KHAN YASMEEN WARI SYIT 21:

Text to search from: Data Science involves statistics, programming, and machine learn

Output - 3A.

```
[8] from bs4 import BeautifulSoup
# Sample HTML content
html_data = """
<html>
<head><title>Welcome Page</title></head>
<body>
<h1>Hello Students!</h1>
<p>This is a Data Science practical session.</p>
</body>
</html>
"""

# Creating a BeautifulSoup object
soup = BeautifulSoup(html_data, 'html.parser')
# Extracting title and paragraph
print("KHAN YASMEEN WARI SYIT21:")
print("Title:", soup.title.string)
print("Paragraph:", soup.p.text)
```

☞ KHAN YASMEEN WARI SYIT21:

Title: Welcome Page

Paragraph: This is a Data Science practical session.

Output - 3B.

Extracting title and Paragraph

Print ("Title : ", soup.title.string)

Print ("Paragraph : ", soup.p.text)

C) Aim :- write a Python code to Print current date in different format.

Program:-

```
from datetime import datetime
```

Get current datetime

current_date = datetime.now()

Display in different formats

Print ("Default : ", current_date)

Print ("DD-MM-YYYY : ", current_date.strftime ("%d-%m-%Y"))

Print ("Full Format : ", current_date.strftime ("%A, %B %d, %Y"))

Print ("YYYY/MM/DD : ", current_date.strftime ("%Y/%m/%d"))

D) write a Python code to convert timestamp to date stamp?

Program:-

```
from datetime import datetime
```

Example UNIX timestamp

converted_date = datetime.fromtimestamp (timestamp)

Print ("Timestamp : ", timestamp)

Print ("Converted Date : ", converted_date.strftime ("%d-%m-%Y
%H:%M:%S"))

16 - 9/9/2024

```
✓ ① from datetime import datetime
# Get current datetime
current_date = datetime.now()
# Display in different formats
print("KHAN YASMEEN WARSI SYIT21:")
print("Default:", current_date)
print("DD-MM-YYYY:", current_date.strftime("%d-%m-%Y"))
print("Full Format:", current_date.strftime("%A, %B %d, %Y"))
print("YYYY/MM/DD:", current_date.strftime("%Y/%m/%d"))

② KHAN YASMEEN WARSI SYIT21:
Default: 2025-07-04 12:48:36.759199
DD-MM-YYYY: 04-07-2025
Full Format: Friday, July 04, 2025
YYYY/MM/DD: 2025/07/04
```

Output - 3C

```
① from datetime import datetime
# Example UNIX timestamp
timestamp = 1700000000
converted_date = datetime.fromtimestamp(timestamp)
print("KHAN YASMEEN WARSI SYIT21:")
print("Timestamp:", timestamp)
print("Converted Date:", converted_date.strftime("%Y-%m-%d %H:%M:%S"))

② KHAN YASMEEN WARSI SYIT21:
Timestamp: 1700000000
Converted Date: 2023-11-14 22:13:20
```

Output - 3D

E) Aim:- write a Python code to develop calendar module.

Program:-

```
import calendar  
year = 2025  
month = 11
```

```
Print ("calendar for.", calendar.month_name[month], year)  
Print (calendar.month (year, month))
```

F) Aim:- write a Python code to compare two dates

Program:-

```
from datetime import date  
first_date = date(2025, 5, 10)  
Second_date = date(2025, 12, 25)
```

if first_date < second_date:

```
    Print ("First Date is earlier.")
```

elif first_date > second_date:

```
    Print ("First Date is later.")
```

else:

```
    Print ("Both dates are the same.")
```

```

import calendar
year = 2025
month = 11
print("KHAN YASMEEN WARSI SYIT21:")
print("Calendar for", calendar.month_name[month], year)
print(calendar.month(year, month))

KHAN YASMEEN WARSI SYIT21:
Calendar for November 2025
November 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

```

Output - 3e

```

from datetime import date
first_date = date(2025, 5, 10)
second_date = date(2025, 12, 25)
print("KHAN YASMEEN WARSI SYIT21:")
if first_date < second_date:
print("First Date is earlier.")
elif first_date > second_date:
print("First Date is later.")
else:
print("Both dates are the same.")


```

KHAN YASMEEN WARSI SYIT21:
First Date is earlier.

output - 3f

A) Aim :- Python code to create a Numpy array.

```
import numpy as np
# creating a numpy array
array = np.array([1, 2, 3, 4, 5])
print("Numpy Array : ", array)
print("Type of arr : ", type(array))
```

By Python code to demonstrate basic operations on a single array.

```
import numpy as np
arr = np.array([10, 20, 30])

print("Original Array : ", arr)
print("Addition (arr + 5) : ", arr + 5)
print("Multiplication (arr * 2) : ", arr * 2)
print("Mean of array : ", np.mean(arr))
print("Sum of array : ", np.sum(arr))
print("Square of array : ", np.square(arr))
print("Sqrt of array : ", np.sqrt(arr))
```

4a

```
import numpy as np

array = np.array([1,2,3,4,5])
print ("yasmeen khan 21")
print ("Numpy Array:", array)
print ("Type of arr:", type(array))
```

yasmeen khan 21
 Numpy Array: [1 2 3 4 5]
 Type of arr: <class 'numpy.ndarray'>

4b

yasmeen khan 21
 Original array: [10 20 30]
 Addition (arr +5): [15 25 35]
 Multiplication (arr*2): [20 40 60]
 Sum of array: 60
 Square of array: [100 400 900]
 squareroot of array: [3.16227766 4.47213595 5.47722558]

c) Aim :- Python code to create an array with 10 elements & slice from 1st to 5th element.

Import numpy as np

arr = np.array([10, 1, 2, 3, 4, 5, 6, 7, 8, 9])

sliced_arr = arr[0:5]

Print("Sliced Elements (1st to 5th):", sliced_arr)

d) Aim :- Python code to sort an array alphabetically

Import numpy as np

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

sorted_arr = np.sort(arr)

Print("Alphabetically sorted Array:", sorted_arr)

e) Aim :- Python code to create a filter array that returns the maximum value(s)

Import numpy as np

arr = np.array([10, 25, 35, 35, 20])

max_value = np.max(arr)

filtered_arr = arr[arr == max_value]

Print("Maximum value(s):", filtered_arr)

4c

```
import numpy as np  
arr = np.array([0,1,2,3,4,5,6,7,8,9,])  
sliced_arr = arr[0:5]  
print("yasmeen khan 47")  
print("sliced Elements (1st to 5th):",sliced_arr)  
  
yasmeen khan 47  
sliced Elements (1st to 5th): [0 1 2 3 4]
```

4d

```
import numpy as np  
arr = np.array(['banana','apple','cherry','date'])  
sorted_arr = np.sort(arr)  
print("yasmeen khan 47")  
print("Alphabetically sorted Array:", sorted_arr)  
  
yasmeen khan 47  
Alphabetically sorted Array: ['apple' 'banana' 'cherry' 'date']
```

4e

```
import numpy as np  
  
arr = np.array ([10,2,35,35,28])  
max_value =np.max (arr)  
filtered_arr =arr[arr == max_value]  
print("yasmeen khan 21")  
print("maximum value(s):",filtered_arr)  
  
yasmeen khan 21  
maximum value(s): [35 35]
```

A) Aim :- Python code to import Pandas and create a dataframe object.

import Pandas as pd

```
data = {  
    'Name': ['Alice', 'Bob', 'Charlie'],  
    'Age': [24, 30, 22]  
}
```

```
df = pd.DataFrame(data)  
print("Dataframe :\n", df)
```

By B) Aim :- Python code to import Pandas and create a dataframe objects.

Import Pandas as pd

```
data = {  
    'Math': [85, 90, 95],  
    'Science': [88, 92, 84]  
}
```

```
df = pd.DataFrame(data)  
print("statistical summary :\n", df.describe())
```

5a

```
import pandas as pd  
data = {  
    'Name': ['Alice', 'Bob', 'Charlie'],  
    'Age': [24, 30, 22]  
}  
df = pd.DataFrame(data)  
print("yasmeen khan 21")  
print("DataFrame:\n", df)
```

yasmeen khan 21

DataFrame:
Name Age
0 Alice 24
1 Bob 30
2 Charlie 22

kedodlo yonius no foos ot eloos nakkhi - 3 mila 20

Gece yonius kroqil

pandas, (kroqil) yonius (7) yonius (7) = yonius
(yono) kroqil fr = yono - kroqil

5b

yasmeen khan 21

Statistical Summary :

	Maths	Science
count	3.0	3.000000
mean	90.8	89.666667
std	5.0	2.881666
min	85.0	88.000000
25%	87.5	88.500000
50%	90.0	89.000000
75%	92.5	90.500000
max	95.0	92.000000

(105, 87, 92, 85, 95) yonius ja 3 mila
(yono) kroqil fr = yono - kroqil

kedodlo yonius = yono - kroqil fr = yono - kroqil
(yono - kroqil, "a(2) yono min yono") kroqil

c) Aim:- Python code to create a Panda series from a dictionary.

```
import pandas Pandas as pd  
data = {'a': 100, 'b': 200, 'c': 300}  
series = pd.Series(data)  
print ("Panda series:\n", series)
```

d) Aim : Python code to filter a Panda series using Boolean array.

```
import Pandas as pd
```

```
series = pd.Series([10, 20, 30, 40, 50])  
filtered_series = series [series > 25]
```

```
print ("Filtered series (values > 25):\n", filtered_
```

5c

```
import pandas as pd
data = {'a':100,'b':200,'c':300}
series = pd.Series(data)
print("yasmeen khan 21")
print("pandas serices:\n",series)

yasmeen khan 21
pandas serices:
a    100
b    200
c    300
dtype: int64
```

Ess, 28, 1850 10/10

(okolo) emeklukola 1850 10
(lo, "a": 100, "b": 200, "c": 300) type

o ba sebeni teknik ola naiye - mba
atistolo emeklolo

ba sebeni teknik ola naiye

5d

```
import pandas as pd
series = pd.Series ([10,20,30,40,50])
filtered_series = series[series > 25]
print("yasmeen khan 21")
print("Filtered Series (values > 25):\n",filtered_series)

yasmeen khan 21
Filtered Series (values > 25):
30
40
50
dtype: int64
```