

Objective :

WAP to Print Current date & Time
In Python.

Code :

```
from Datetime Import Datetime  
Current_datetime.now()  
print ("Current Date & Time:",  
      Currentdatetime)  
# Printing Date & Time
```

⇒ When we run this code, it will
print Current Date & time in
following

format :-

YYYY -MM- DD

HH:~~MM~~:SS

If Sir

Output :

Current Date & Time : 2024-03-24 14:56:32

Experiment - 13.

Objective :

Demonstrate Working with Tuples

Code :

Making Tuple

Tuple = ('Aryan', 'Rohit', 'Mohit')

Printing Tuple

Print (Tuple)

In this code a Tuple is created with values 'aryan', 'rohit' & 'mohit' inside quotations & the result is simply printed in the compiler

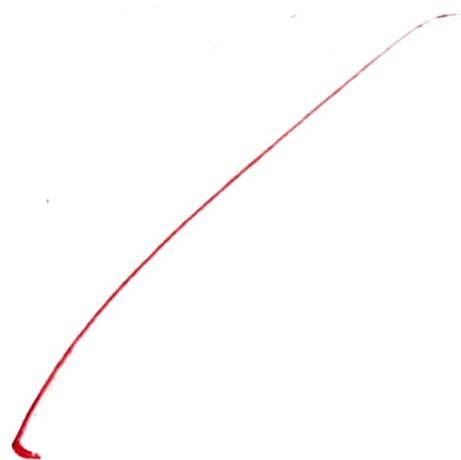
8
60

Output :

Aryan

Rohit

Mohit



Experiment - 14

objective: WAP to demonstrate working with dictionaries

Code:

Making Dict

dict = { "Fruit": "apple", "Car": "Bugatti" }

Print Dictionary
Print (dict)

In this code a Dictionary is created using curly brackets with values fruit- apple & car = Bugatti & the result is simply printed on compiler.

(Q)
Qo

SJM

output

Fruit

car

Apple

Bugatti

Experiment-15

Objective :- WAP to demonstrate any ?math built in function.

Code :

i) import math

base = 2

power = 4

result = math.pow(base, power) // Finding power
print(result)

ii) import math

result = math.sqrt(num) // Finding square root
print(result)

Note :

→ In the (i) program math module is used and pow() function is used.

→ In (ii) program sqrt() function is used.

Output :

i) 16.0

ii) 4.0

Experiment-16

Objective : WAP to demonstrate the use of user defined module

Code :-

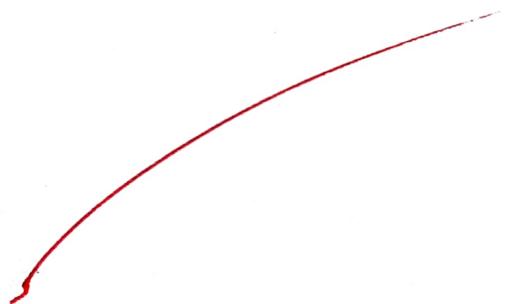
Making a module GFG. }
Defining a function }
def greet():
 print ("Hello Buddy") }
Defining a variable }
~~location = "Jaipur"~~ }

Importing Module
import GFG
Using the function greet
GFG.greet()
Print the variable declared
print (GFG.location)

Output

Hello Buddy

Jaipur



Experiment - 17

Objectives :- Perform following operations
on dictionaries.

- Create
- Access
- update
- Delete

Code :-

```
this dict = {  
    "brand": "ford",  
    "model": "mustang",  
    "year": 1969}
```

```
x = this dict.get("model")
```

```
this dict.update({"color": "red"})
```

```
y = this dict.get("color")
```

```
del this dict["model"]
```

```
print(this dict)
```

```
print(x)
```

```
print(y)
```

Experiment-18

Objective :- Perform following operations
on Tuples

- Create
- Delete
- Update
- Access

Code

Create
This Tuple = ("apple", "Banana", "cherry")
print (This Tuple)

Access
print (This Tuple[1])

update
y = list (This Tuple)
y = append ("orange")
This Tuple = Tuple (y)
print (This Tuple)

delete
del This Tuple
print (This Tuple)

Output

("Apple", "Banana", "cherry")

("Banana")

("Apple", "Banana", "cherry", "orange")

Experiment-19

Objective :- Demonstrate different numbers data Types in Python

Code

a = 200

b = 200.2

~~print (Type (a))
print (Type (b))~~

~~The~~ In this we give the value, the output will give the data Type of value

Output :

{class 'int'
{class 'float'}

output

{class 'int'}

{class 'float'}

Experiment-20

Objective : WAP To perform various arithmetic operators on numbers in Python.

Code

Assign value To a Variable 'A'

a = 10

Assign value To a variable 'B'

b = 20

~~print (a+b)~~

~~print (a*b)~~

~~print (a/b)~~

~~print (a-b)~~

Output

30
200
0.5
-10



Experiment-21

Objective : Define a module to find fibonacci numbers and import module to another program.

Code

```
# create 2 Files # create module
1) Fibonacci
    If n<0 :
        print ("Incorrect input")
    elif n == 0 :
        return 0
    elif n == 1 or n == 2 :
        return 1
    else :
        return fibonacci(n-1) + fibonacci(n-2)
```

```
2)
    Import fibonacci # import module
    a = fibonacci.fibonacci(10)
    print(a)
```

output :

→ 55

Experiment - 23

Objective :- Program to create, concatenate and print a string

Code

```
# Creating String  
# Str1
```

~~Str1 = "Jayesh"~~

~~Str2 = " "~~

~~Str3 = " Lores Maggie"~~

~~# printing result~~

result = Str1 + Str2

print (result)

output:-

Jayesh loves Maggie

Experiment-23

Objective :- Define a module and import a specific function in that module to another programs

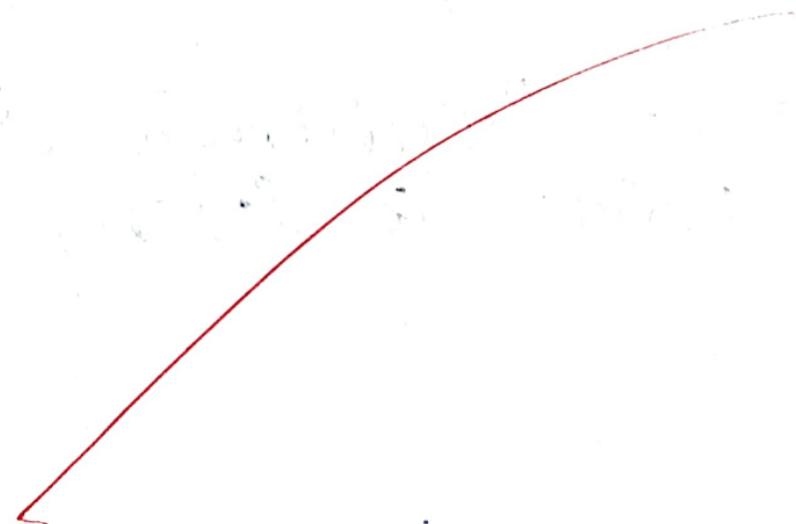
Code

```
import math  
# Imported module  
# making Sqrt  
result = math.sqrt(20)  
# print result  
print(result)
```

Note : Here we are finding square root of a number using math.sqrt

output

4.47213595



Experiment-24

Objective :- Print prime numbers less than

Code

```
def is_prime(num):
```

```
    if num <= 1:
```

```
        return False
```

~~```
for i in range(2, int(num**0.5)+1):
```~~~~```
    if num % i == 0:
```~~~~```
 return False
```~~

Return True

```
def print_prime_lessThan(limit):
```

```
 prime = [num for num in range
```

```
(2, limit)
```

```
 if is_prime(num)
```

```
 print ("Prime numbers less than",
```

```
limit, " are:", prime)
```

# Example usage:

```
print_prime_lessThan(50)
```

output

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47