a given programme any number of times.

The Syntax of a function () looks as follows :-

```
def func1 ():        ] declaration
    print ("Hello")
```

Keyword

Function Call :-

Whenever we want to call a function we put the name of the function followed by parenthesis as follows -

```
func1()
```

function defination :-

The part containing the exact set of instruction which are executed during the function call.

Q.1 WAP to greet a user with "good day" Name using function ?

→

```
def name (name1):
    print ("Good day" + name1)
name1 = input ("Enter user name")
name (name1)
```

Output :- Enter user Name :- Sakshi
Good day Sakshi

10-1-23

Types of function :

There are two :

1. Built in () -
2. User defined ()

Example of Built :

len(), print(

* Function With Argum

Some values

We can put this

→ A function ca

```
def greet (
    gr = "Hell
    return. gr
a = greet ("
print (a)
```

* Default Param

default a

10-1-23

PAGE NO.:

Types of function in python.

There are two types of function in python.

1. Built in () — Already present in python
2. User defined () — define by the user.

Example of Built in () into →

len(), print(), range() etc-

* **function With Argument :—** A function can accept some values it can work with.

We can put this values in paranthesis ()

→ A function can also return values as on below.

```
def greet (name):
    gr = "Hello" + name
    return gr
a = greet ("Meera")
print (a)
```

function declaration block.

function Call

* **Default Parameter value :—** We can have a value as default argument in a function

If is Specify

Name = "Stranger"
in the line Containing def. this value is used when now argument are passed.

```
def greet (Name = "Stanger")
    gr = "Hello" name
    print (gr)
a = greet. ("Meera")
```

Output :-

* Recursion :-
Recursion is a function which call itself it is used to directly use Mathematicly formula as a function.

factorial :- 2! / 2 = 1 × 2 = 2
3! = 1 × 2 × 3 =
4! = 1 × 2 × 3 × 4 =
fact (n-1) n

fact (n) = fact (n-1) n 0! = 1
1! = 1

def function (n)
n x fact (n-1)
4 x fact (3)
4x3 fact (2)
4x3x2 fact (1)
4x3x2x1 fact (0)
4x3x2x1x0 fact (0-1)

1 *
```
def recur_factorial (n):
    if n==1:
        return n
    else:
        return n*
recur-factorial (n-1)
num = 4
if num < 0:
    print (" Sorry, factorial does not exist for
            nagative numbers")
elif num == 0:
    print (" The factorial of 0 is 1")
else:
    print (" The factorial of ", num "is",
           recur - factorial (num))
```

Output :—

PAGE NO.:

Chapter - 6 Practice Set

1 Write a program to find Greatest of four numbers Entered by the user?

→