Recursion

Thursday, July 10, 2025 8:04 PM

Phenomenun of Function calling itself until the termination condition becomes true is known as recursion.

Syntax:

i. Without return value:

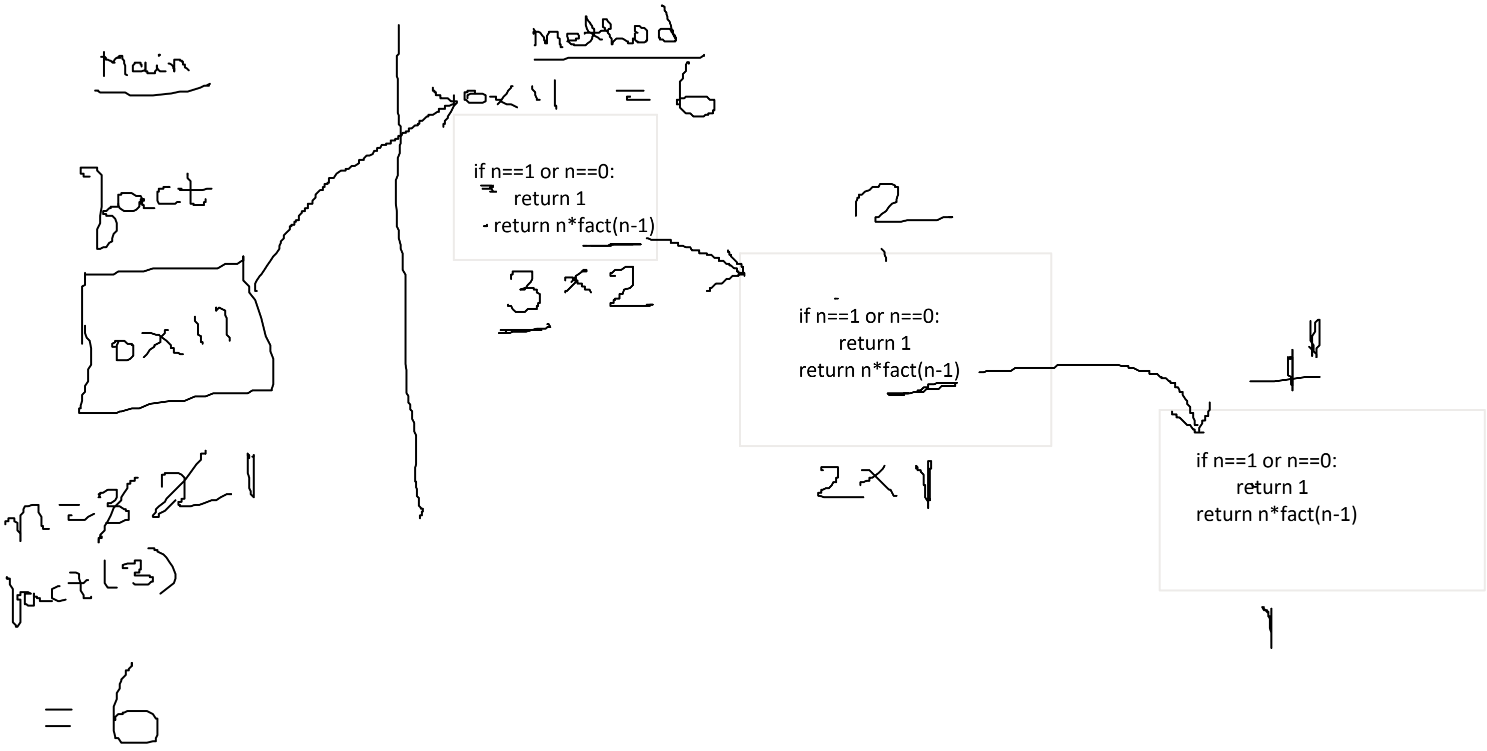
```
Def fname(args):
        If term_condn:
                Return
        Fname(args)
Fname(values)
```

ii. With return value:

```
Def fname(args):
        If term_condn:
                Return value
        Return fname(args)
Print(fname(values))
```

Eg:

```
#WAP to find the factorial of a number
def fact(n):
   if n==1 or n==0:
      return 1
   return n*fact(n-1)
print(fact(3))
```



```
if n==1 or n==0:
      return 1
  return n*fact(n-1)
```

```
if n==1 or n==0:
      return 1
return n*fact(n-1)
```

```
if n==1 or n==0:
      return 1
return n*fact(n-1)
```

As per python 3.7 we can create 990 blocks only, but now it have been updated to 1023 blocks.
To increase the number, we have to use:
import sys
sys.setrecursionlimit(value)

Steps to convert looping statement to recursion:

i. Initialization of all the looping variables should be done in the formal arguments only.
ii. The termination condition should be written exactly opposite to the looping condition in form of if statement.
iii. Return the total result inside the termination condition.
iv. Logic of the program should be written as it is.
v. Updation of the looping variable should be done in the recursive call.