Python Recursive Function

Anoop S Babu
Faculty Associate
Dept. of Computer Science & Engineering
bsanoop@am.amrita.edu

Function Call inside another Function

```
def mul(a,b):
    return a*b
def sum(x,y,z):
    return x+mul(y,z)    # call for mul()function

n1 = 4; n2 = 3; n3 = 2
print("{}+{}*{} = {}".format(n1,n2,n3,sum(n1,n2,n3)))
```

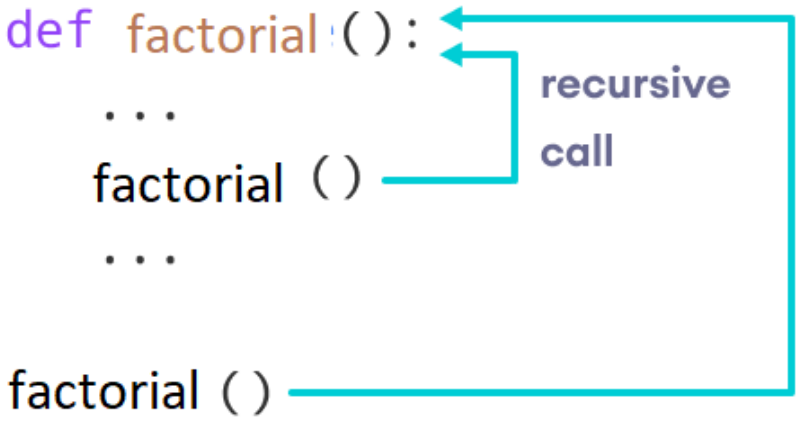
Output

```
4+3*2 = 10
```

Python Recursive Function

- A recursive function is a function that **calls itself** during its execution.
- Best example of a recursive function – ***factorial.***
  - n! = n * (n-1)!

Working of Recursive Function



Example of a Recursive Function

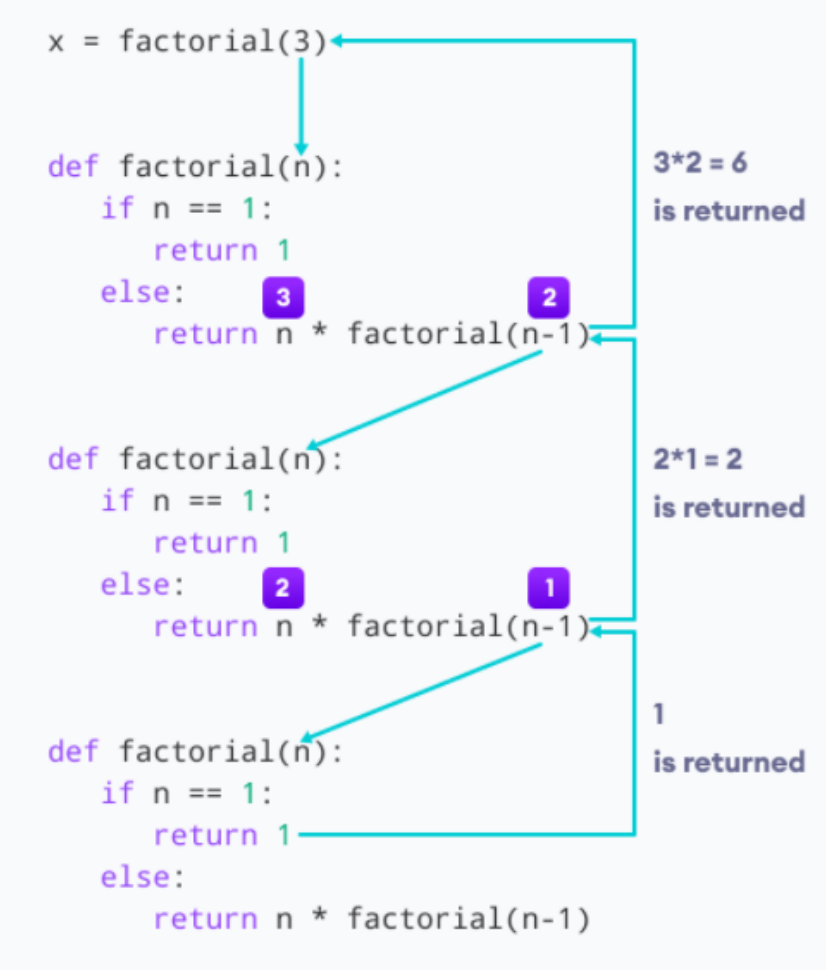
```
def factorial(number):
    if number == 1 or number == 0:
        return 1
    else:
        return (number * factorial(number - 1))

n = 5
print(n, "! = ", factorial(n))
```

Output

```
5 ! =  120
```

Working of a Recursive Factorial Function



Depth of Recursion

- Python interpreter **limits the depths of recursion** to help avoid infinite recursions, resulting in stack overflows.
- By default, the maximum depth of recursion is 1000.
- If the limit is crossed, it results in RecursionError

Example

```
def recursor():
    recursor()
recursor()
```

Output

```
Traceback (most recent call last):
  File "<string>", line 3, in <module>
File "<string>", line 2, in recursor
  File "<string>", line 2, in recursor
  File "<string>", line 2, in recursor
  [Previous line repeated 996 more times]
RecursionError: maximum recursion depth exceeded
```