

Tail Recursion

What is Tail Recursion?

- A recursive function call is tail recursive when the recursive call is the last thing executed by the function.
- A recursive function is said to be tail recursive if there is nothing to do after the function returns except return its value.

Factorial Example

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

Factorial Tail Recursive

```
def factorial(n, a = 1):  
    if n == 0:  
        return a  
    else:  
        return factorial(n-1, a*n)
```

Benefits of Tail Recursion

Tail recursive functions can be optimized by the compiler. Since the recursive call is the last statement, there is nothing left to do in the current function, so saving the current function's stack frame is of no use.