**Code**

def factorial(n):

return 1 if n == 0 else n \* factorial(n - 1)

print(factorial(3))

**1. Function Definition**

* The function factorial(n) is defined.
* It uses **recursion**, which means the function calls itself to solve smaller subproblems.

**2. Base Case**

return 1 if n == 0

* If n is 0, the function immediately returns 1.
* This is the **base case** that prevents the recursion from going infinitely.

**3. Recursive Case**

else n \* factorial(n - 1)

* If n is not 0, the function calculates n \* factorial(n - 1).
* This reduces the problem size by subtracting 1 from n in each recursive call.

**4. Execution of factorial(3)**

Let’s break it step by step for factorial(3):

**Step 1: First Function Call (factorial(3))**

* n = 3
* Since n != 0, the function goes to the recursive case:

return 3 \* factorial(3 - 1) # This becomes 3 \* factorial(2)

* The result depends on the calculation of factorial(2).

**Step 2: Second Function Call (factorial(2))**

* n = 2
* Since n != 0, the function goes to the recursive case:

return 2 \* factorial(2 - 1) # This becomes 2 \* factorial(1)

* The result depends on the calculation of factorial(1).

**Step 3: Third Function Call (factorial(1))**

* n = 1
* Since n != 0, the function goes to the recursive case:

return 1 \* factorial(1 - 1) # This becomes 1 \* factorial(0)

* The result depends on the calculation of factorial(0).

**Step 4: Fourth Function Call (factorial(0))**

* n = 0
* Since n == 0, the function hits the base case:

return 1

* This value 1 is returned to the previous call (factorial(1)).

**5. Returning Values**

Now the recursion "unwinds" step by step:

**Step 3 (Returning to factorial(1)):**

* factorial(1) calculates:

1 \* 1 = 1

* Returns 1 to the previous call (factorial(2)).

**Step 2 (Returning to factorial(2)):**

* factorial(2) calculates:

2 \* 1 = 2

* Returns 2 to the previous call (factorial(3)).

**Step 1 (Returning to factorial(3)):**

* factorial(3) calculates:

3 \* 2 = 6

* Returns 6.

**6. Final Output**

print(factorial(3)) # Output: 6

* The function prints 6, which is the factorial of 3