# Recursive Functions

A recursive function is a function that calls itself. It can help solve problems by breaking them down into smaller versions of the same problem.

## Basic Structure

def function\_name():  
 if base\_condition:  
 return result  
 else:  
 return function\_name()

## Example 1: Factorial

def factorial(n):  
 if n == 0:  
 return 1  
 return n \* factorial(n - 1)  
  
print(factorial(5)) # 120

## Example 2: Fibonacci

def fibonacci(n):  
 if n == 0:  
 return 0  
 if n == 1:  
 return 1  
 return fibonacci(n - 1) + fibonacci(n - 2)  
  
print(fibonacci(6)) # 8

## Example 3: Countdown

def countdown(n):  
 if n == 0:  
 print("Done!")  
 else:  
 print(n)  
 countdown(n - 1)  
  
countdown(3)

## Important Notes

- Recursive functions must have a base case to avoid infinite recursion.

- Too many recursive calls can cause a RecursionError.

## Exercise

Write a recursive function that calculates the sum of all natural numbers up to n.  
  
Example:  
sum\_to\_n(5) → 1 + 2 + 3 + 4 + 5 = 15