# Chapter 5 : Recursion

Aviral Janveja

## 1 Introduction

Recursion is a process where a function calls itself in order to solve a problem. It's commonly used for tasks that can be broken down into smaller, similar subproblems. Let us look at a simple example for the same :

```
def factorial(n):
    if n == 0: # Base case
        return 1
    else:
        return n * factorial(n - 1) # Recursive step

# Example usage:
result = factorial(5)
print("Factorial of 5 is :", result)
```

In the above example, the factorial function calculates the factorial of a non-negative integer. It does so by recursively calling itself with a smaller argument (n - 1) until it reaches the **base case** (n == 0), at which point it returns 1. Then, as the recursion unwinds, it multiplies the return value of each recursive function call by the current value of n.

So, when you call factorial (5), it would calculate 5 * factorial (4), then 4 * factorial (3), then 3 * factorial (2), then 2 * factorial (1), and finally it reaches 1 * factorial (0), at which point it returns 1. Then, the recursion unwinds and the final result is computed as 1 * 1 * 2 * 3 * 4 * 5, which is 120. Checkout an enhanced version of the above example here : recursion.py, further you can use pythontutor.com to visualize the recursive process above, step by step.

### 1.1 Base Case

In recursion, the **base case** is the condition that determines when the recursive calls should stop. It is the termination condition that prevents the function from calling itself indefinitely, thereby ensuring that the recursion eventually ends.

In the above example of the factorial function, the base case is when n is equal to 0. When n becomes 0, the function returns 1 without making any further recursive calls. This is necessary to prevent infinite recursion, as without a base case, the function would keep calling itself indefinitely. So, the base case acts as the stopping condition for the recursion, allowing the recursive calls to unwind and the function to return a final result.

## 2 Inductive Reasoning

## 3 Tower of Hanoi Example

Important example. Hard to solve iteratively. but recursion provides a clean solution.

## 4 Fibonacci Series Example

## 5 Palindrome