**Concept of Recursion**  
Recursion is a programming technique where a function calls itself to solve a problem. This approach is effective for problems that can be broken down into smaller, self-similar subproblems

A recursive function always has two main components.

**Base Case**: A simple condition that terminates the recursion. This is the scenario where the function does not call itself and instead returns a direct result. Without a base case, a recursive function would run indefinitely, leading to a stack overflow error.

**Recursive Step**: The part of the function where it calls itself, but with modified arguments that move the problem closer to the base case. This step reduces the complexity of the problem with each call.

Recursion can lead to more elegant and readable code for problems with a repetitive structure, as it often avoids complex nested loops.

## Setup

For this financial forecasting tool, we will model future value calculation as a recursive problem. The value after one period is the current value plus the growth during that period. This can be expressed recursively.

The formula for the value in the next period is:  
FutureValue = CurrentValue \* (1 + growthRate)

The recursive definition is:

**Base Case**: If there are no more periods to calculate (periods = 0), the future value is the current value.

**Recursive Step**: The future value for n periods is calculated by finding the value after one period and then using that result as the new current value for the remaining n-1 periods.