BECOME A DSA Champion

TechZeen

🔁 What is Recursion?

- Recursion is a technique where a function calls itself.
- Used to solve a big problem by breaking it into smaller subproblems of the same type.
- Every recursive function must have:
- Base Case → stops the recursion
- Recursive Case → continues the recursion

⭐ Why Do We Use Recursion?

- Makes complex problems easier to write and understand.
- Best for problems that have repetitive patterns or self-similar structure.
- Reduces long loops and nested loops.
- Common in:
- Tree & Graph traversal
- Divide and Conquer (Merge Sort, Quick Sort)
- Backtracking (Maze, N-Queen)
- Mathematical problems (factorial, Fibonacci)

🧠 Call Stack Memory

(How Recursion Works?)

- Every recursive call is stored in the Call Stack.
- Stack keeps track of:
- Function arguments
- Local variables
- Return address
- Recursion follows LIFO (Last In, First Out).
- When the base case is reached → stack starts returning back one by one.
- Too many recursive calls can cause Stack Overflow Error.

⚡ Types of Recursion

- Direct Recursion → function calls itself directly
- Indirect Recursion → function A calls B, B calls A
- Tail Recursion → recursive call at the end of function
- Non-Tail Recursion → recursive call in middle of logic
- Most recursion problems use non-tail recursion.