During my research on recursion, I stumbled upon a valuable resource-an insightful article on GeeksforGeeks titled Introduction to Recursion. This article, available at https://www.geeksforgeeks.org/introduction-to-recursion-2/, offers a comprehensive understanding of recursion. It delves into its definition, the steps to implement recursive algorithms, and provides examples in various programming languages. The article also covers crucial topics such as base cases, recursive cases, and potential issues like stack overflow errors. With its clear explanations and code examples, this article is a must-read for anyone interested in learning about recursion.

Recursion is a programming technique where a function calls itself. Key aspects include:

• **Base case**: A condition to stop the recursion and prevent infinite loops.

• **Recursive case**: Breaking down the problem into smaller subproblems.

• **Termination**: Ensuring the base case is eventually reached.

• **Combining solutions**: Using results from subproblems to solve the original problem.

Recursion, a powerful programming technique where a function calls itself, finds numerous practical applications. These include tree/graph traversal, sorting algorithms, divide-and-conquer problems, and even generating fractals. While recursion can sometimes be less efficient than iteration due to increased memory usage, it offers elegant solutions to complex issues. Understanding recursion requires practice, but its potential uses make it a valuable skill for any programmer.