Linked Lists – A Beginner's Guide

What is a Linked List?

Imagine a chain of boxes, where each box holds some info and a pointer (like an arrow) to the next box. That's a linked list! It's a way to store items one after another but with more flexibility than arrays.

Each box (called a node) contains:

- Data (your actual value)

- Reference/pointer to the next (and sometimes previous) box

Thus, A **Linked List** is a linear data structure where elements (nodes) are linked using pointers or references. Each node contains data and reference(s), giving the structure dynamic memory allocation and efficient insert/delete operations compared to arrays.

Types of Linked Lists

- **Singly Linked List:** Each node points to the next node. Traversal is unidirectional.
- **Doubly Linked List:** Each node points to both previous and next nodes. Traversal can be bidirectional.
- **Circular Linked List:** The last node points back to the first, forming a circle. It can be singly or doubly circular.

Key Operations (Insert, Delete, Traverse)

- **Insertion:** Add a node at the beginning, end, or given position.
- **Deletion:** Remove a node from the beginning, end, or by value.
- **Traversal:** Visit all nodes to perform actions like print or search

For the beginner's: The most senior's and my own experience say, to start for the linked list, from very basic prefer the Jenny's Lecture channel. To understand linked list and implement, you need not to remember something, it just starts making sense when u get to know.

This is the firstmost topic in my DSA learning journey, also afterwards you can shift to the Striver's AtoZ DSA sheet.

Additional Learning Resources

- Interactive Tutorial: GeeksforGeeks Linked List in Java
- Beginner's Video: Linked List Introduction – YouTube
- Complete Types & Diagrams: W3Schools Linked List Types
- Step-by-step Practice: Java LinkedList Practice Problems
- Visual Animations: Programiz - Linked List Data Structure