**Understand Linked Lists**

**What is a Linked List?**

• A linear data structure made of nodes connected by pointers.  
• Each node contains data and a reference (link) to the next node.  
• Does not store elements in contiguous memory locations.  
• Grows and shrinks dynamically during runtime.  
• Efficient for insertion and deletion operations.  
• Not suitable for direct (random) access like arrays.

**Singly Linked List:**  
• Each node points to the next node.  
• Operations go one direction only (forward).  
• More memory-efficient than doubly linked list.  
• Simpler to implement and manage.  
• Deletion requires tracking the previous node manually.

**Doubly Linked List:**  
• Each node points to both next and previous nodes.  
• Allows bidirectional traversal (forward and backward).  
• Uses more memory (extra pointer for previous).  
• Easier deletion and reverse traversal.  
• Slightly more complex to implement.

**Circular Linked List:**  
• The last node points back to the first node.  
• Forms a closed loop (no null).  
• Can be singly or doubly linked.  
• Useful in round-robin scheduling or buffering.