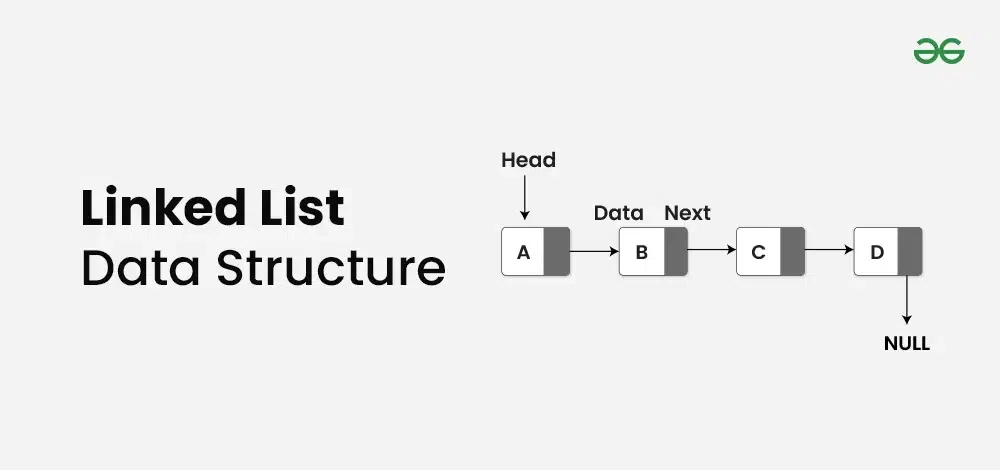
It consists of nodes where each node contains **data** and a**reference (link)** to the next node in the sequence.

This allows for dynamic memory allocation and efficient **insertion** and **deletion** operations compared to arrays.



**What is a Linked List?**

A **linked list**is a linear data structure that consists of a series of nodes connected by pointers.

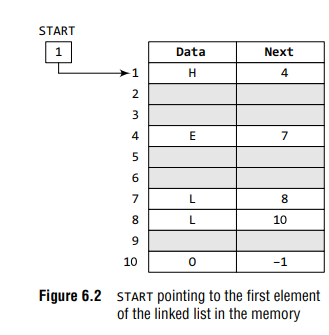
Each node contains **data** and a **reference(link)** to the next node in the list. Unlike **arrays, linked lists**allow for efficient **insertion** or **removal** of elements from any position in the list, as the nodes are not stored contiguously in memory.

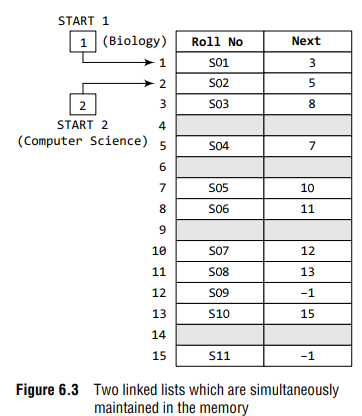
**Linked List:**

* **Data Structure:** Non-contiguous
* **Memory Allocation:** Dynamic
* **Insertion/Deletion:** Efficient
* **Access:** Sequential

**Array:**

* **Data Structure:** Contiguous
* **Memory Allocation:** Static
* **Insertion/Deletion:** Inefficient
* **Access:** Random



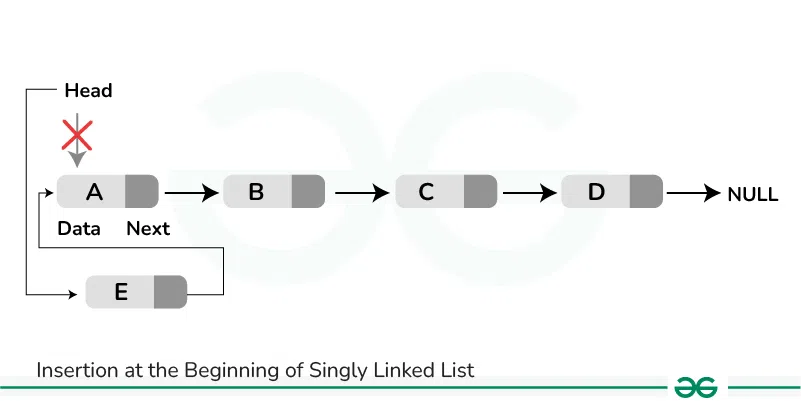


**What is Singly Linked List?**

*A****singly linked list****is a linear data structure in which the elements are not stored in contiguous memory locations and each element is connected only to its next element using a pointer.*

***In singly linked list Link(pointer linking) hamesa aage ki taraf hota hai peeche ki taraf nhi***

## [**How to Insert a Node at the Front/Beginning of Linked List**](https://www.geeksforgeeks.org/insert-a-node-at-front-beginning-of-a-linked-list/)



**Complexity Analysis:**

* **Time Complexity:** O(1), We have a pointer to the head and we can directly attach a node and change the pointer. So the Time complexity of inserting a node at the head position is O(1) as it does a constant amount of work.
* **Auxiliary Space:**O(1)