A **linked list** is a fundamental data structure in computer science. 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**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 Lists vs Arrays](https://www.geeksforgeeks.org/linked-list-vs-array/)

### **Linked List:**

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

### **Array:**

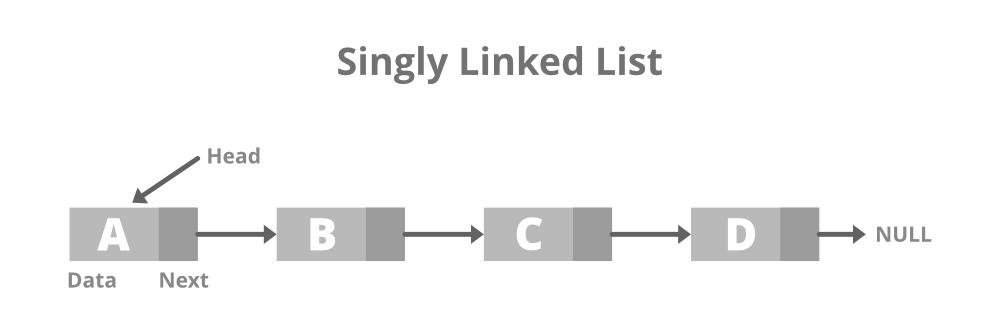
* **Data Structure:**Contiguous
* **Memory Allocation:**Static
* **Insertion/Deletion:** Inefficient
* **Access:**Random

## Types Of Linked List:

### **1.**[**Singly Linked List**](https://www.geeksforgeeks.org/singly-linked-list-definition-meaning-dsa/)

*It is the simplest type of linked list in which every node contains some data and a pointer to the next node of the same data type.*

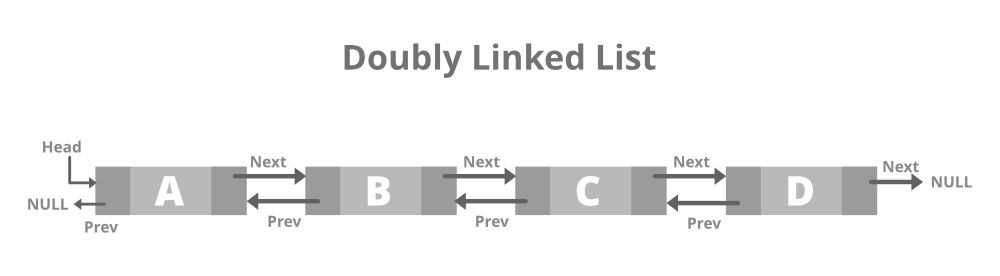
The node contains a pointer to the next node means that the node stores the address of the next node in the sequence. A single linked list allows the traversal of data only in one way. Below is the image for the same:



### **2.** [**Doubly Linked List**](https://www.geeksforgeeks.org/doubly-linked-list/)

*A doubly linked list or a two-way linked list is a more complex type of linked list that contains a pointer to the next as well as the previous node in sequence.*

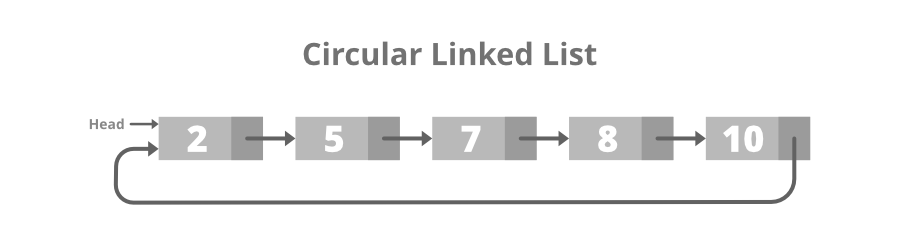
Therefore, it contains three parts of data, a pointer to the next node, and a pointer to the previous node. This would enable us to traverse the list in the backward direction as well. Below is the image for the same:



### **3.**[**Circular Linked List**](https://www.geeksforgeeks.org/circular-linked-list/)

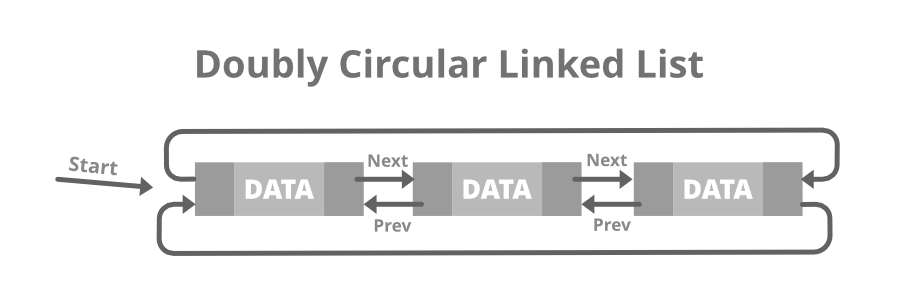
*A circular linked list is that in which the last node contains the pointer to the first node of the list.*

While traversing a circular linked list, we can begin at any node and traverse the list in any direction forward and backward until we reach the same node we started. Thus, a circular linked list has no beginning and no end. Below is the image for the same:



### **4.**[**Doubly Circular linked list**](https://www.geeksforgeeks.org/circular-doubly-linked-list-meaning/)

A Doubly Circular linked list or a circular two-way linked list is a more complex type of linked list that contains a pointer to the next as well as the previous node in the sequence. The difference between the doubly linked and circular doubly list is the same as that between a singly linked list and a circular linked list. The circular doubly linked list does not contain null in the previous field of the first node. Below is the image for the same:



### **5.**[**Header Linked List:**](https://www.geeksforgeeks.org/header-linked-list-in-c/)

*A header linked list is a special type of linked list that contains a header node at the beginning of the list.*

So, in a header linked list **START** will not point to the first node of the list but **START** will contain the address of the header node. Below is the image for Grounded Header Linked List:

