

Doubly Link-List

D.L.L OVERVIEW

- ✗ D.L.L. Linked lists
 - + Abstract data type (ADT)
- ✗ Basic operations of linked lists
 - + Insert, find, delete, print, etc.
- ✗ Variations of linked lists
 - + Doubly linked lists
 - + Circular D.L.L.
- ✗ Use of both Method :
 - + (LIFO – FIFO)

The main difference between Single Linked List and Double Linked List is that a node in the single linked list stores the address of the next node while a node in a double linked list stores the address of the next node and the previous node.

Types of Linked List

3

There are mainly three types of linked list

➤ **Singly linked list**

- Each node has only one link part that contains the address of next node.

➤ **Circular linked list**

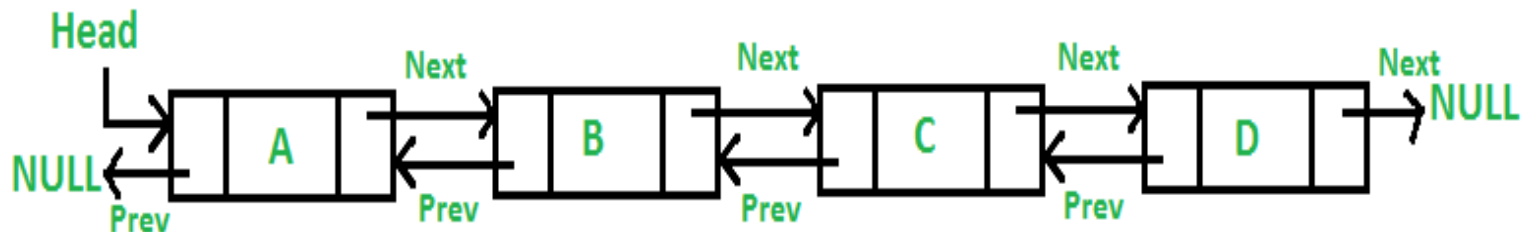
- In this linked list the linked field of the last node contain the address of the first node of list

➤ **Doubly linked list**

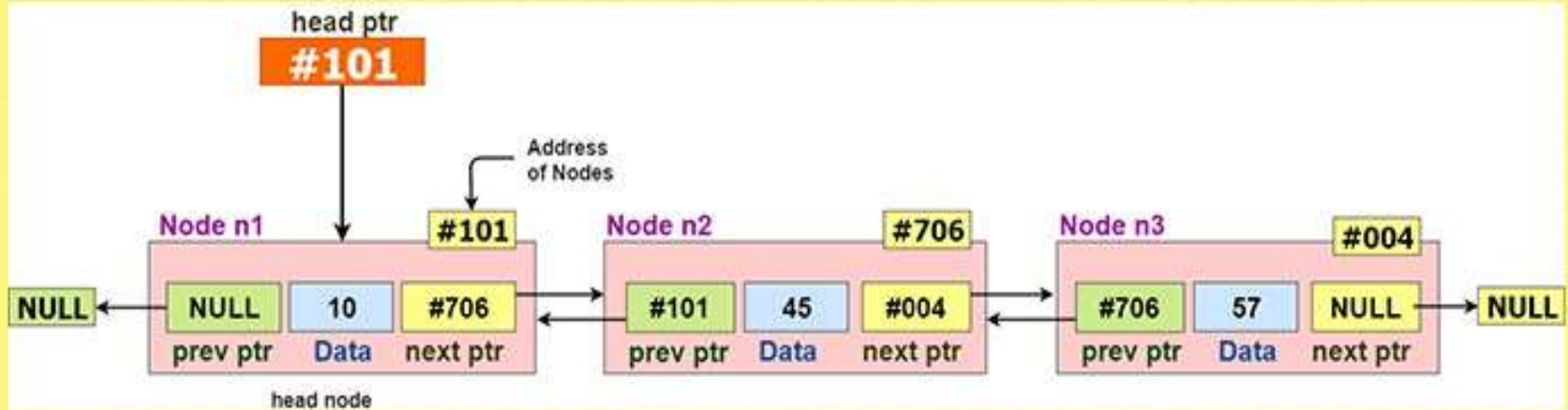
- In this linked list all nodes are linked together by multiple number of links which help in accessing both the successor and predecessor node from the given node position

DOUBLY LINKED LIST

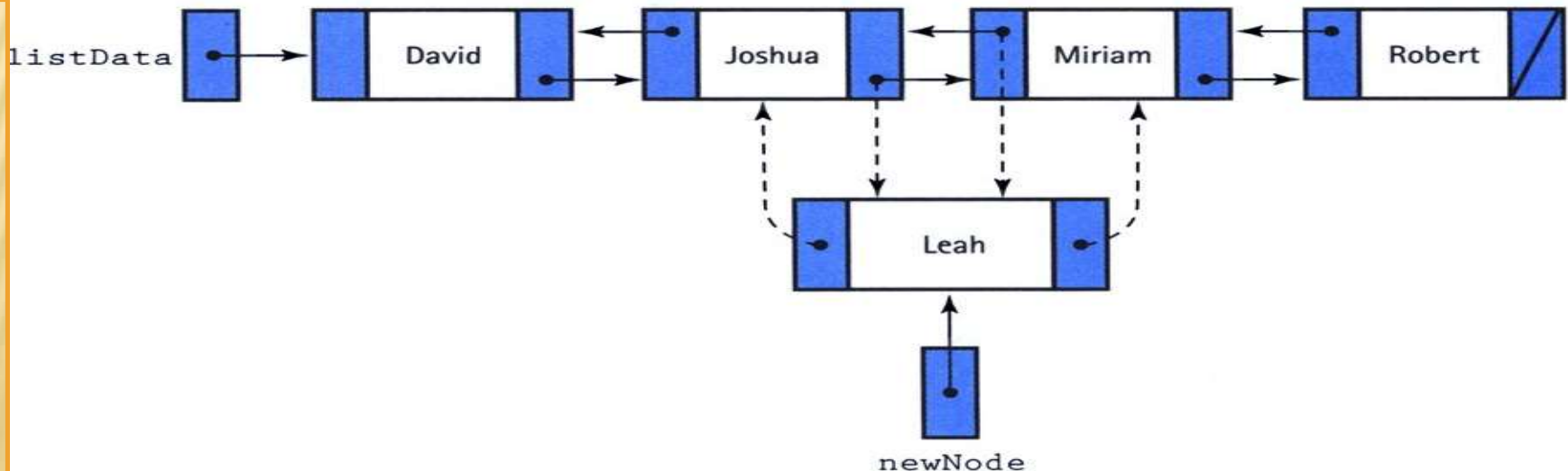
- ✖ A doubly linked list is a linked data structure that consists of a **set** of sequentially linked records called nodes.
- ✖ Each node contains two fields, called links, that are **references** to the previous and to the next node in the sequence of nodes. and one data field.



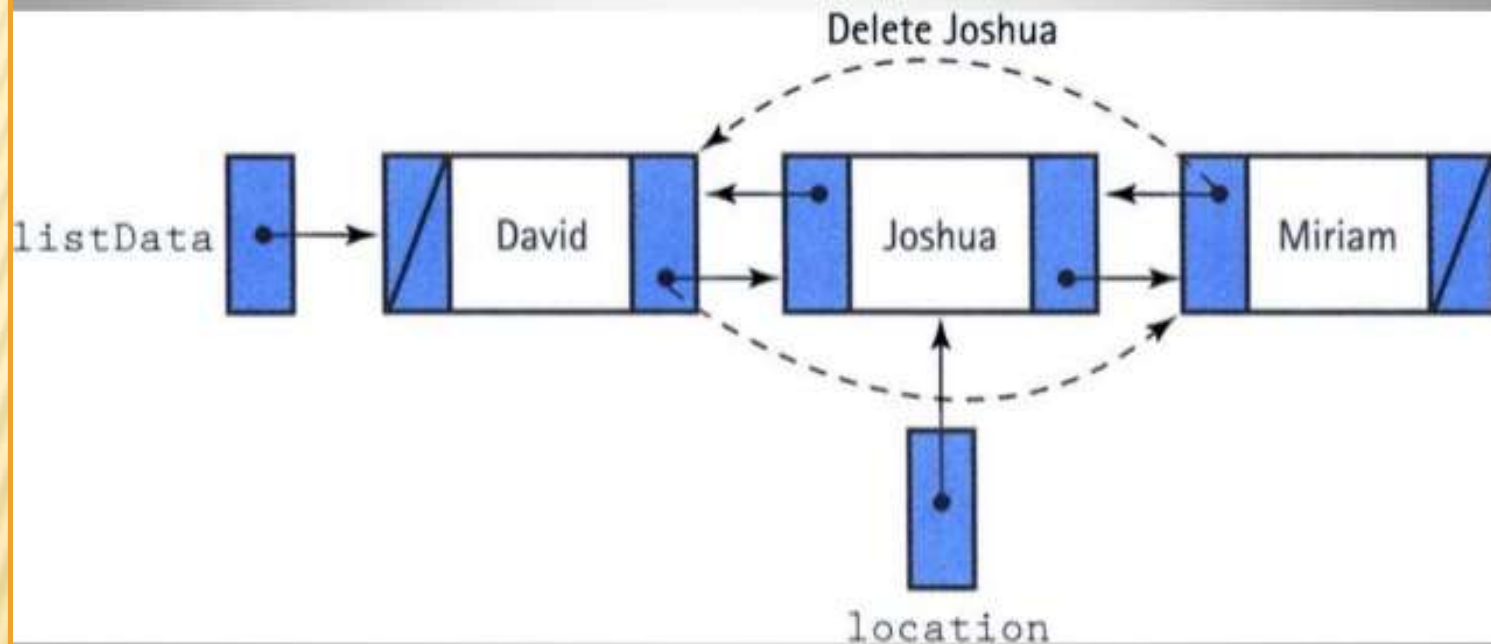
DOUBLY LINKED LIST



Inserting into a doubly linked list



Delete an element at any place doubly linked list



BASIC OPERATIONS

- ✕ Following are the basic operations supported by a list.
- ✕ **Insertion** – Adds an element at the beginning of the list.
- ✕ **Deletion** – Deletes an element at the beginning of the list.
- ✕ **Insert Last** – Adds an element at the end of the list.
- ✕ **Delete Last** – Deletes an element from the end of the list.
- ✕ **Insert After** – Adds an element after an item of the list.
- ✕ **Delete** – Deletes an element from the list using the key.
- ✕ **Display forward** – Displays the complete list in a forward manner.
- ✕ **Display backward** – Displays the complete list in a backward manner.

✖ Advantages over singly linked list

- 1) A DLL can be traversed in both forward and backward direction.
- 2) The delete operation in DLL is more efficient if pointer to the node to be deleted is given.

✖ Disadvantages over singly linked list

- 1) Every node of DLL Require extra space for an previous pointer. It is possible to implement DLL with single pointer though.
- 2) All operations require an extra pointer previous to be maintained. For example, in insertion, we need to modify previous pointers together with next pointers.

SINGLE LINKED LIST VERSUS DOUBLE LINKED LIST

SINGLE LINKED LIST

A linked list that contains nodes which have a data field and a next field which points to the next node in the line of nodes

Allows traversing in one direction through the elements

Requires less memory as it stores only one address

Complexity of insertion and deletion at a known position is $O(n)$

DOUBLE LINKED LIST

A linked list that contains the data field, next field that points to the next node and a previous field that points to the previous node in the sequence

Allows traversing in both directions (backward and forward)

Requires more memory as it stores two address

Complexity of insertion and deletion at a known position is $O(1)$