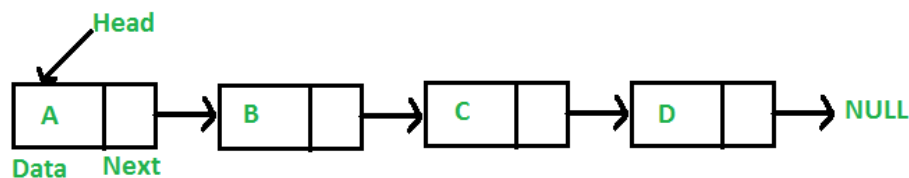
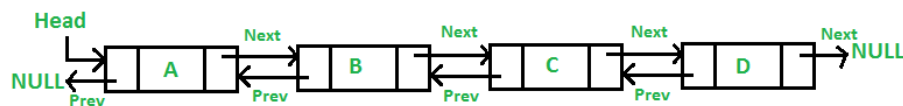


# Singly Vs Doubly Linked List (Advantages & Disadvantages)

**Introduction to Singly linked list :** A singly linked list is a set of nodes where each node has two fields 'data' and 'link'. The 'data' field stores actual piece of information and 'link' field is used to point to next node. Basically the 'link' field stores the address of the next node.



**Introduction to Doubly linked list :** A **Doubly Linked List (DLL)** contains an extra pointer, typically called *previous pointer*, together with next pointer and data which are there in singly linked list.



## Singly linked list vs Doubly linked list

Aa Singly linked list (SLL)	≡ Doubly linked list (DLL)
<u>SLL nodes contains 2 field -data field and next link field.</u>	DLL nodes contains 3 fields -data field, a previous link field and a next link field.
<u>Untitled</u>	
<u>In SLL, the traversal can be done using the next node link only. Thus traversal is possible in one direction only.</u>	In DLL, the traversal can be done using the previous node link or the next node link. Thus traversal is possible in both directions (forward and backward).
<u>The SLL occupies less memory than DLL as it has only 2 fields.</u>	The DLL occupies more memory than SLL as it has 3 fields.

Aa Singly linked list (SLL)	≡ Doubly linked list (DLL)
<u>Complexity of insertion and deletion at a given position is <math>O(n)</math>.</u>	Complexity of insertion and deletion at a given position is $O(n / 2) = O(n)$ because traversal can be made from start or from the end.
<u>Complexity of deletion with a given node is <math>O(n)</math>, because the previous node needs to be known, and traversal takes <math>O(n)</math>.</u>	Complexity of deletion with a given node is $O(1)$ because the previous node can be accessed easily
<u>We mostly prefer to use singly linked list for the execution of stacks.</u>	We can use a doubly linked list to execute heaps and stacks, binary trees.
<u>When we do not need to perform any searching operation and we want to save memory, we prefer a singly linked list.</u>	In case of better implementation, while searching, we prefer to use doubly linked list.
<u>A singly linked list consumes less memory as compared to the doubly linked list.</u>	The doubly linked list consumes more memory as compared to the singly linked list.