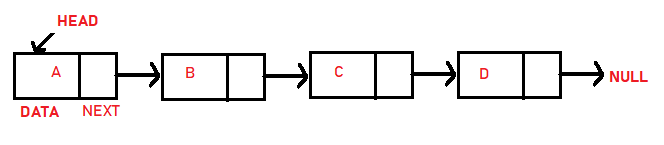
In this article, we have covered the applications of Linked List, Circular Linked List and Doubly Linked List. We start with the basics of Linked List and then, move to applications of the different types of Linked List.

**Linked list 🤔**

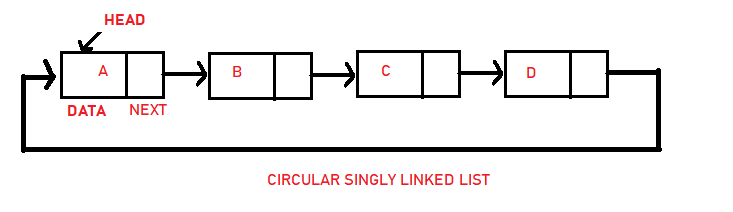
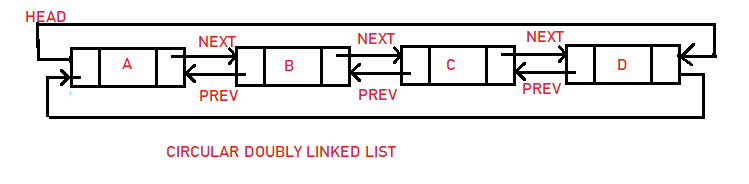
**The linked list** is a linear data structure, that has a sequence of continuous nodes. A single node is just the object which contains things like, "**data**" and a "**next**" pointer which points to the next node in case of singly and circular linked list, and also "**previous**" pointer which will point to the previous node in case of a doubly linked list. In this way, it forms **a chain-like structure**. It can easily perform insertion, append, deletion operations without reorganization of the entire list because memory allocation is done during run-time but if we want the same operations on an array that needs to allocate fixed memory, its run time will get more expensive. So, this is the most important factor for using linked lists in case of memory and storage than arrays.  
Linked list can perform operations from basics to a higher level. As it is **dynamic**, an increase or decrease in size/length can be done as per need. It is abundantly used in real-life applications because of such smooth functioning and many benefits to the memory.

So, we will be discussing the **applications** of all three types of Linked List as following:

***Singly Linked List:****This Linked list is unidirectional, which will start from the head and end up with a tail containing data and pointer to the next node in a particular node.  
*

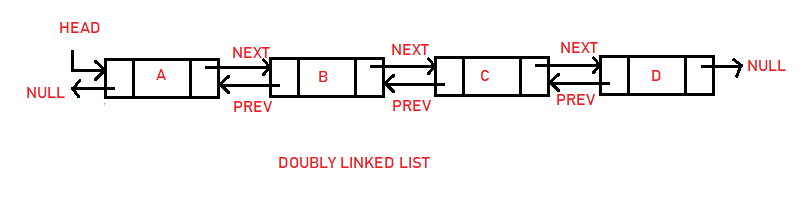
**Applications of Singly Linked List are as following:**

1. It is used to implement **stacks** and **queues** which are like fundamental needs throughout computer science.
2. To prevent the collision between the data in the **hash map**, we use a singly linked list.
3. If we ever noticed the functioning of a casual notepad, it also uses a singly linked list to perform undo or redo or deleting functions.
4. We can think of its use in a photo viewer for having look at photos continuously in a slide show.
5. In the system of train, the idea is like a singly linked list, as if you want to add a Boggie, either you have to take a new boggie to add at last or you must spot a place in between boggies and add it.

***Circular Linked List:****This linked list performs a circular form, as its first node points to its next node, and the last node points to the first head node forming a circle. Both singly and doubly linked lists can be made into a circular linked list.  
  
*

**Applications of Circular Linked List are as following:**

1. It can also be used to implement queues by maintaining a pointer to the last inserted node and the front can always be obtained as next of last.
2. Circular Doubly Linked Lists are used for the implementation of advanced data structures like Fibonacci Heap. Also reference to the previous node can easily be found in this.
3. It is also used by the Operating system to share time for different users, generally uses a Round-Robin time-sharing mechanism (this algorithm comes from the round-robin principle, where each person gets an equal share of something in turns. It is the oldest, simplest scheduling algorithm, which is mostly used for multitasking).
4. Multiplayer games use a circular list to swap between players in a loop.
5. In photoshop, word, or any paint we use this concept in undo function.

***Doubly Linked List:****It is a complex type of linked list in which each node apart from storing its data has two links. The first link points to the previous node in the list and the second link points to the next node in the list.  
*

**Applications of Doubly Linked List are as following:**

1. Great use of the doubly linked list is in navigation systems, as it needs front and back navigation.
2. In the browser when we want to use the back or next function to change the tab, it used the concept of a doubly-linked list here.
3. Again, it is easily possible to implement other data structures like a binary tree, hash tables, stack, etc.
4. It is used in music playing system where you can easily play the previous one or next one song as many times one person wants to. Basically it provides full flexbility to perform functions and make the system user-friendly.
5. In many operating systems, the thread scheduler (the thing that chooses what processes need to run at which times) maintains a doubly-linked list of all the processes running at any time. This makes it easy to move a process from one queue (say, the list of active processes that need a turn to run) into another queue (say, the list of processes that are blocked and waiting for something to release them).
6. It is used in a famous game concept which is a deck of cards.

**Questions**

***Why insertion is faster in linked list?*** *LinkedList's each element maintains two pointers (addresses) which points to the both neighbor elements in the list  
Inserts Performance: LinkedList add method gives O(1) performance while ArrayList gives O(n) in worst case.*

***Does linked list have index?*** *It's important to mention that, unlike an array, linked lists do not have built-in indexes. This means that in order to find a specific point in the linked list, you need to start at the beginning and traverse through each node, one by one, until you find what you're looking for.*