

Pre Requisites:

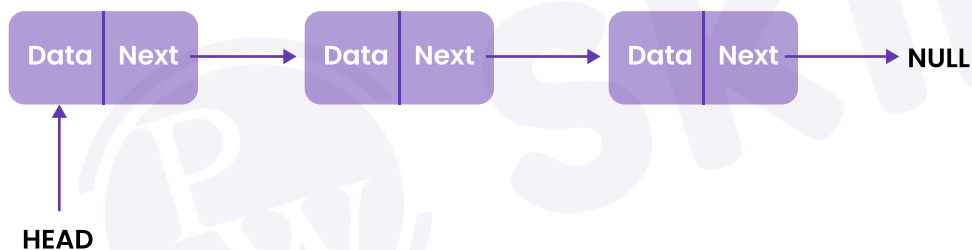
- Basic JAVA syntax

List of concepts involved :

- Introduction to linked list
- Types of linked list
- Node in a linked list
- Insertion in a linked list
- Display of a linked list
- Deletion in a linked list
- Reverse string iteratively
- Reverse string recursively
- Middle of a linked list
- Cycle detection in a linked list

What is a linked list ?

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image:

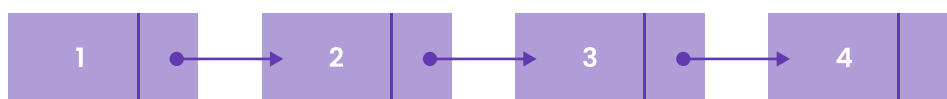


The biggest feature of a linked list is that the elements are not at contiguous memory locations. There is a possibility that the first element or node(to be more precise) is at the 1200th memory location while the second node is at the 3210th location.

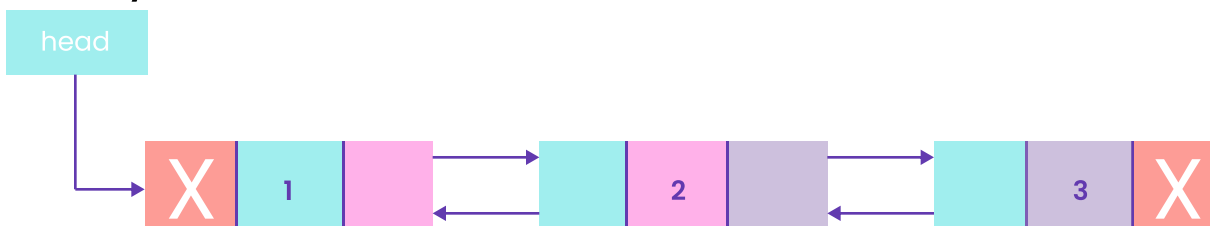
In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

There are different types of linked list :

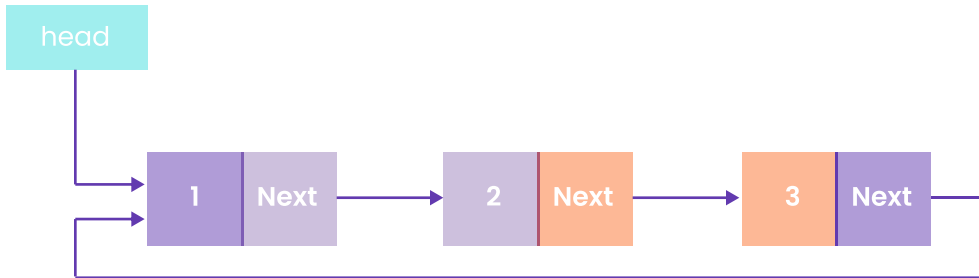
• Singly linked list:



• Doubly linked list



• Circular linked list



Node :

- A node is the most basic element of a linked list. It has essentially two components. One is the data and the other one is a pointer or address of the next node.
- The first node of any linked list is termed as head node while the last node is referred as tail node. A tail node has an address field as "null" since there is no node ahead.

Implementation of a node in a singly linked list where each node has one data element and one address field.

Code : [LP_Code1.java](#)

Approach : Create a class Node which has two attributes: data and next. Next is a pointer to the next node.

Note : here note point is that the data can be an integral value , if you want to store string the data type can be changed to string as in "String name".

Insertion in a singly linked list :

Usually there are 3 types of insertion in a linked list:

- Insert at front
- Insert at middle
- Insert at last

Insertion at last in a linked list :

Code : [LP_Code2.java](#)

Approach :

- addNodeAtLast() will add a new node to the list:
 - Create a new node.
 - It first checks whether the head is equal to null which means the list is empty.
 - If the list is empty, both head and tail will point to the newly added node.
 - If the list is not empty, the new node will be added to the end of the list such that tail's next will point to the newly added node. This new node will become the new tail of the list.

Displaying the linked list: To see what we have stored in the linked list we need to print our linked list. We can traverse over the linked list just as we traverse on an array.

Code : [LP_Code3.java](#)

Nodes :
1 2 3 4