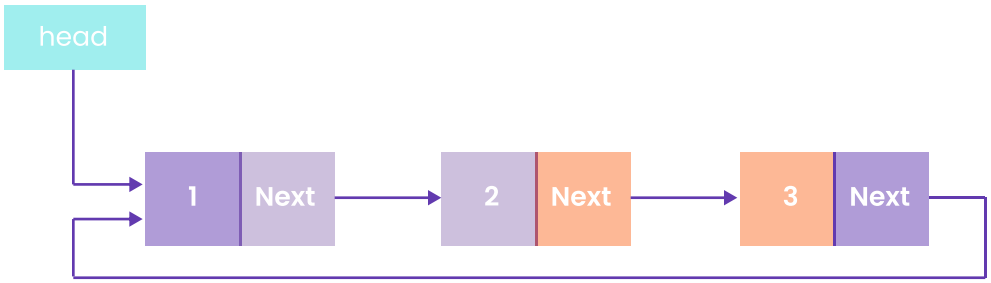
- **Circular linked list**



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
```

Approach :

- In the displayNodes() function, we have simply started traversing the linked list using a while loop until the node does not become null.
- We keep on printing the data stored in the current node and move to the next node.

Deletion in a singly linked list :

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

- Delete at front
- Delete at middle
- Delete at last

Deletion from any general position in a linked list :

Code : LP_Code4.java

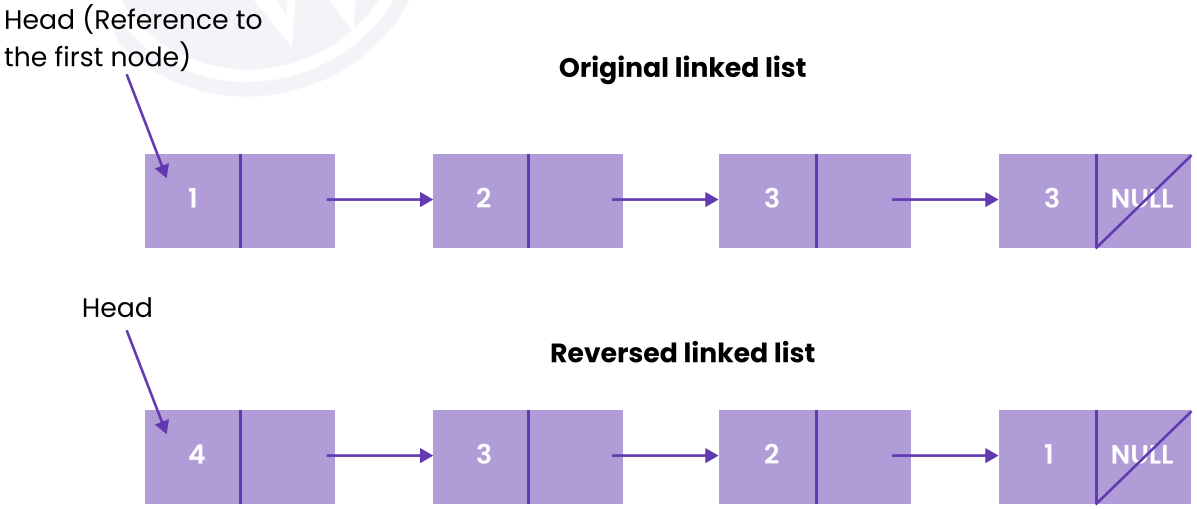
Output :

```
Created Linked list is:
4 3 2 1
Linked List after Deletion of 2:
4 3 1
```

Approach :

- To delete a node from the linked list, we need to do the following steps:
  - Find the previous node of the node to be deleted.
  - Change the next of the previous nodes.
  - Free memory for the node to be deleted.

Reverse a linked list :



A linked list can be reversed in two different ways :

One can be an iterative approach to reverse the linked list and the second could be a recursive approach.
Iterative approach :

Code : LP_Code5.java