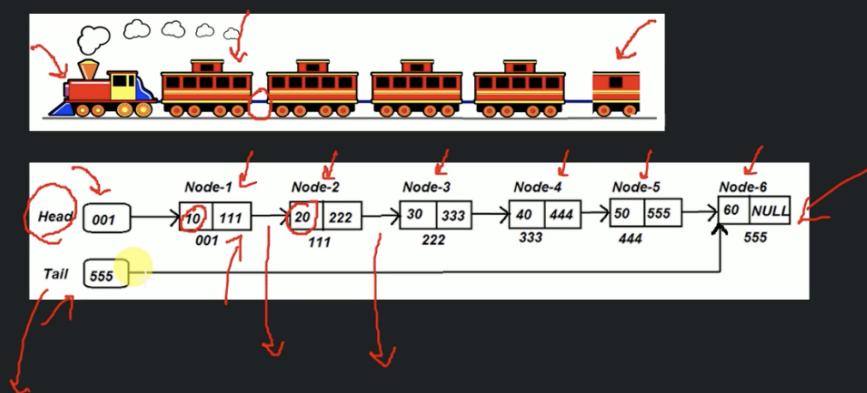


What is Linked List ?

✓ A linked list is a linear data structure where each element is a separate object. Each element (node) of a list comprises of two items - the data and a reference to the next node. The most powerful feature of Linked List is that it is of variable size.

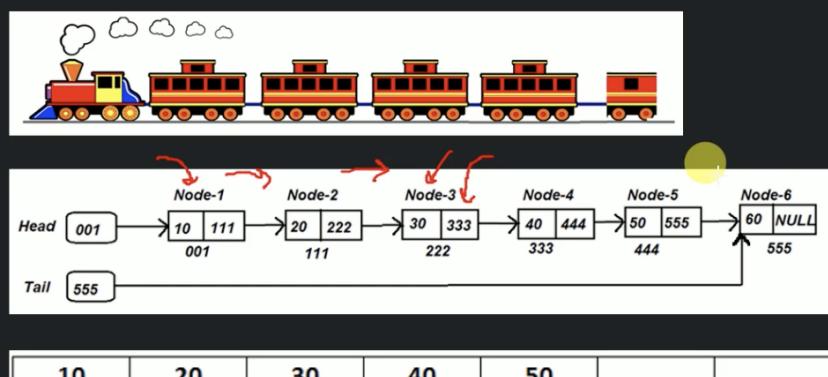
✓ Example:



What is Linked List ?

✓ A linked list is a linear data structure where each element is a separate object. Each element (node) of a list comprises of two items - the data and a reference to the next node. The most powerful feature of Linked List is that it is of variable size.

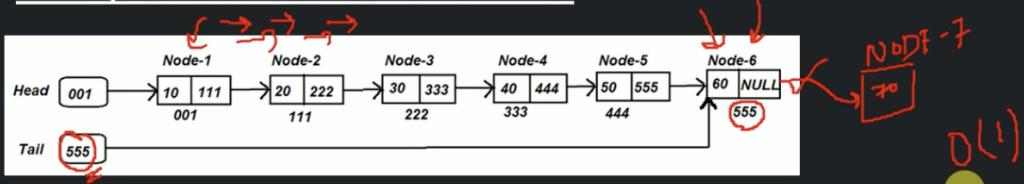
✓ Example:



✓ Linked list vs Array:

1. Separate object
2. Variable size
3. Random access ✓

Components of Linked List:



✓ **Node**: Contains Data & Reference to next Node.

✓ **Head**: Reference to first node in the list.

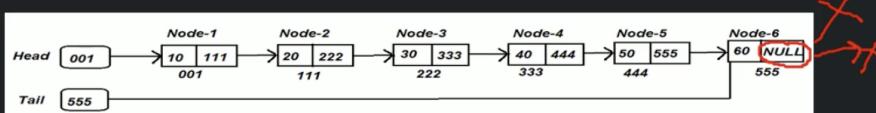
✓ **Tail**: Reference to last node of the list.

Types of Linked List:

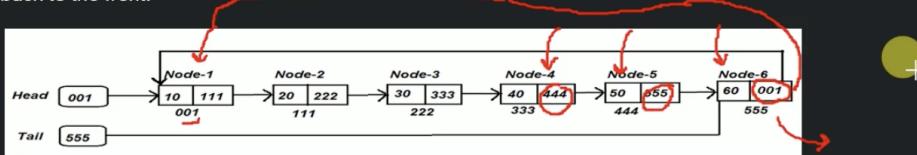
- ✓ Single linked List:
- ✓ Circular Single Linked List:
- ✓ Double Linked List:
 - ✓ Circular Double Linked List:

Types of Linked List:

✓ **Single linked List:** In a singly linked list each node in the list stores the data of the node and a reference to the next node in the list. It does not store any reference to the previous node.

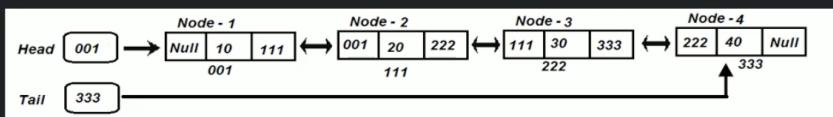


✓ **Circular Single Linked List:** In the case of a circular singly linked list, the only change that occurs is that the end of the given list is linked back to the front.

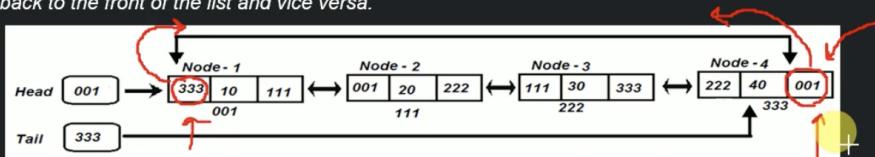


Types of Linked List:

✓ **Double Linked List:** In double linked list each node contains two references, that references to the previous and next node.



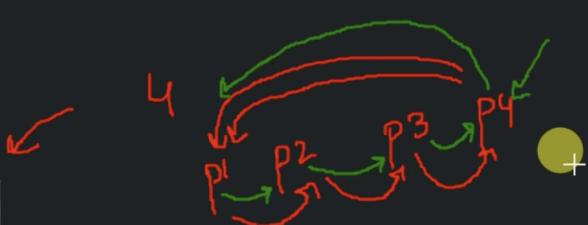
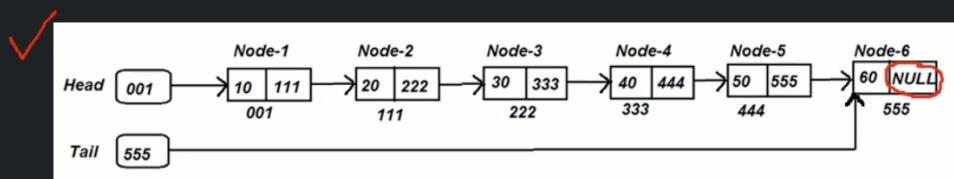
✓ **Circular Double Linked List:** In the case of a circular doubly linked list, the only change that occurs is that the end of the given list is linked back to the front of the list and vice versa.



Why so many types of Linked List ?

✓ Single Linked List:

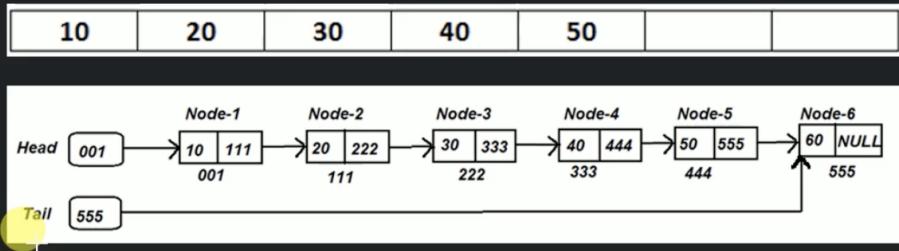
✓ Is most basic form of linked list which give the flexibility to add/remove nodes at runtime.



Why so many types of Linked List ?

✓ Single Linked List:

- ✓ Is most basic form of linked list which give the flexibility to add/remove nodes at runtime.



✓ Circular Single Linked List:

- ✓ When we want to loop through the list indefinitely until the list exists.
- ✓ Example: Multiplayer board game. If we are tracking player's turn in Linked list.



Why so many types of Linked List ?

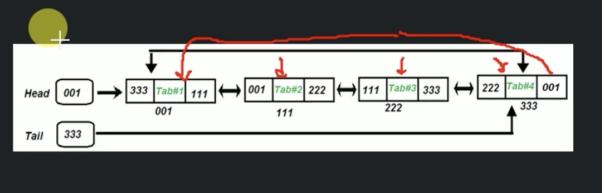
✓ Double Linked List:

- ✓ When we want to move in both direction depending on requirement.
- ✓ Example: Music player which has next and prev buttons.

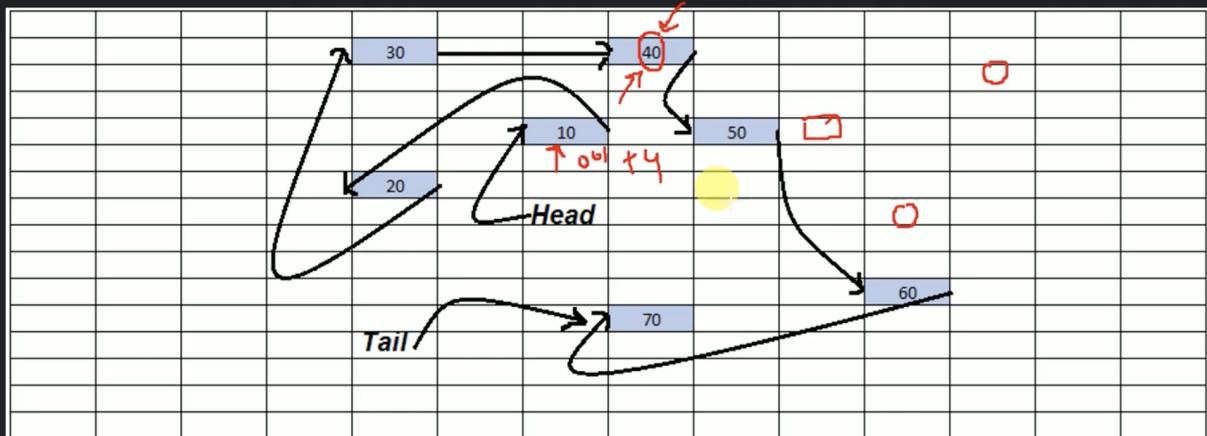


✓ Circular Double Linked List:

- ✓ When we want to loop through the list indefinitely until the list exists. We also want to move both forward and backward.
- ✓ Example: "Alt+Tab" button in Windows.



How is Linked List represented in Memory ?



Unlike array linklist can add/remove element randomly i.e link list allocate memory at runtime and clear it out while array occupied storage once initialized and can not change. That is the reason array represented in continuous in memory while linklist does need continuous chunk of memory

Common operations of Linked List:

- ✓ *Creation of Linked List*
- ✓ *Insertion of Linked List*
- ✓ *Traversal of Linked List*
- ✓ *Searching in a Linked List*
- ✓ *Deletion of a node from a Linked List*
- ✓ *Deletion of Linked List*