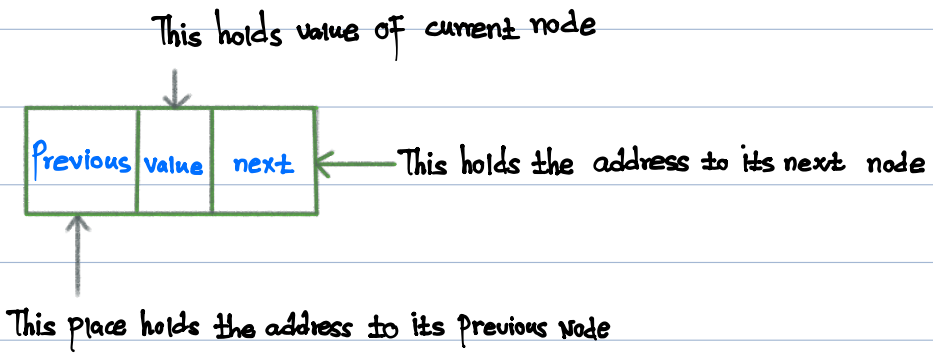
Doubly Linked List

in Doubly linked list every node has the reference of its previous and next node.



It can observed that every node is pointing to its previous node and its next node.

The first node which is also known as head of a linked list, its previous value is No

This means, Head's previous reference holder points to nothing.

The last node is known as tail of a linked list and its next points to nothing.

Advantages

1. Doubly linked list can be traversed in both directions: forward and backward.
2. Given reference to a node, Insertion and deletion before and after that node can be done at constant time (OCU).
3. Given reference to a node, deletion of that node be done at constant time (OCU).

Disadvantages

1. Extra space is used to keep previous references

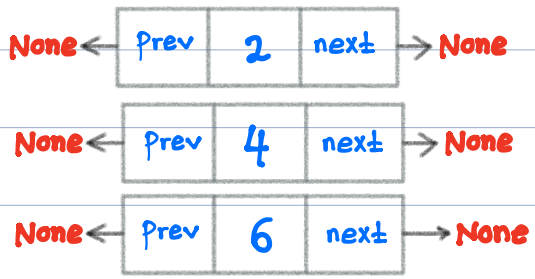
Example: Construct the given linked list using python.


```
Class Node:
    def __init__(self, data):
        self.data = data
        self.next = None
        self.Prev = None
```

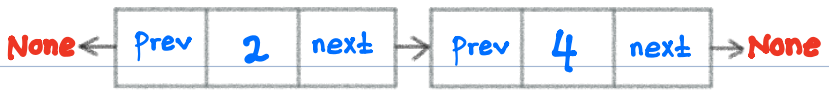
head = Node (2)

temp = Node (4)

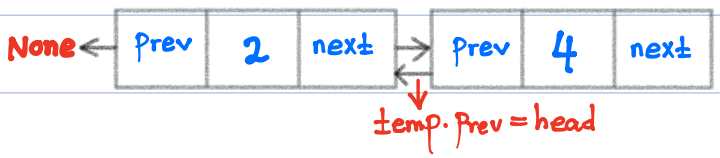
temp1 = Node (6)



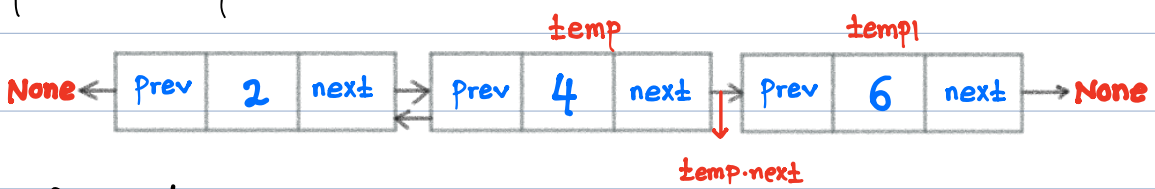
head.next = temp



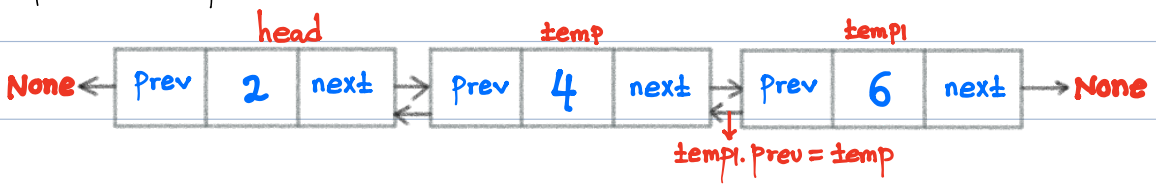
temp.prev = head



temp.next = temp1



temp1.prev = temp



Note: head.Prev and temp1.next are pointing to none, because when building a node from node class, Prev and next, both of them are having values None and unless we change them, they remains the same. for example for node temp we change both of them.