

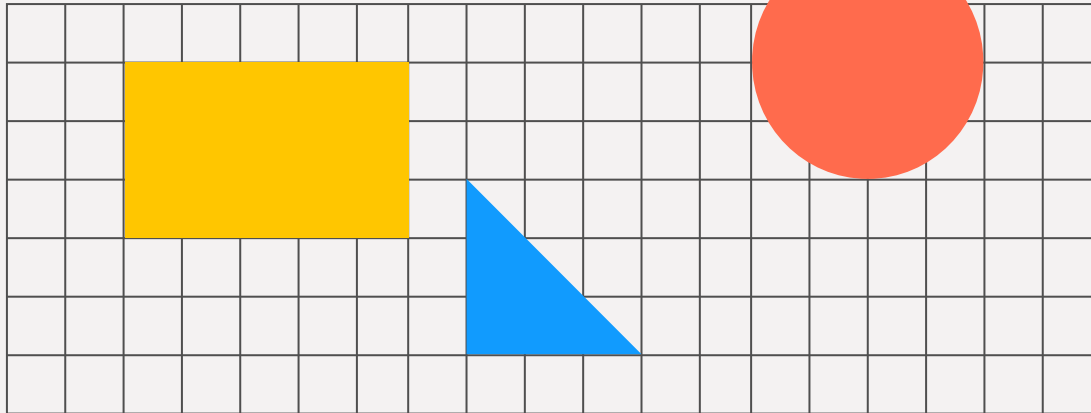
# Doubly Linked List

By:

C125 Mayank Sejpal

C126 Rushabh Shah

C127 Udit Suvarna



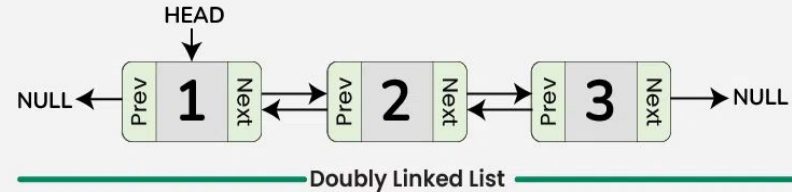
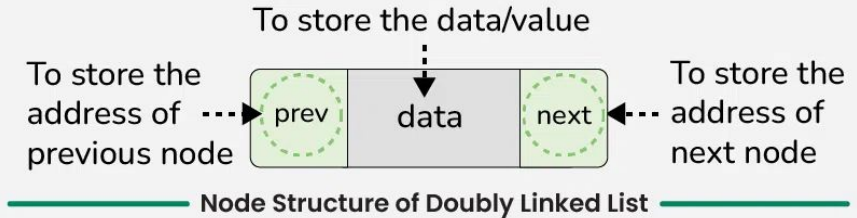
# Introduction



A doubly linked list is a linked data structure that consists of a set of sequentially linked records called nodes.

A doubly linked list is represented using nodes that have three fields:

1. Data
2. A pointer to the next node (next)
3. A pointer to the previous node (prev)



# Why is it Better?



Doubly linked list is better than other data structures in certain scenarios:

- Compared to arrays, it has faster insertion & deletion and also dynamic memory allocation.
- Compared to singly linked list, it allows bidirectional traversal.
- Compared to trees, it has faster insertion & deletion.

# Pros and Cons



## Pros

- Bidirectional traversal
- Easy insertion and deletion of nodes
- Implementation of stack and queue

## Cons

- More complex than singly linked list
- Larger memory overhead

# Real Life: Browser History



Doubly linked lists are used by web browsers for managing forward and backward navigation.

- Each visited page is stored as a node in the list.
- The previous pointer links to the last visited page, and the next pointer links to the next page.
- Clicking "Back" moves to the previous node.
- Clicking "Forward" moves to the next node.

# Code Implementation

```
@Rush-Code10 →/workspaces/codespaces-blank $ cd Project
@Rush-Code10 →/workspaces/codespaces-blank/Project $ javac Main.java
@Rush-Code10 →/workspaces/codespaces-blank/Project $ java Main
Enter number of elements:
5
Enter elements:
10 6 18 4 9
Original List:
10 <-> 6 <-> 18 <-> 4 <-> 9 <-> null
Sorted List:
4 <-> 6 <-> 9 <-> 10 <-> 18 <-> null
Enter element to search:
10
Element 10 found in the list.
```

```
@Rush-Code10 →/workspaces/codespaces-blank/Project $ java Main
Enter number of elements:
5
Enter elements:
20 15 17 10 9
Original List:
20 <-> 15 <-> 17 <-> 10 <-> 9 <-> null
Sorted List:
9 <-> 10 <-> 15 <-> 17 <-> 20 <-> null
Enter element to search:
22
Element 22 not found.
```