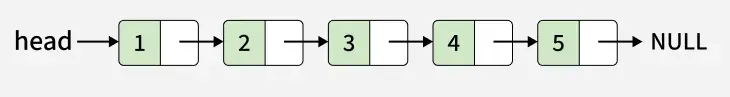
# Singly Linked List Tutorial

A **singly linked list** is data structure, it consists of nodes(nodes = data + reference). The next of the last node is null - indicating the end of the list.



### Understanding Node Structure

In a singly linked list, each **node** consists of two parts: **data** and **a pointer** to the next node. This structure allows nodes to be dynamically linked together, forming a chain-like sequence.

// Definition of a Node in a singly linked list

public class Node {

// Data part of the node

int data;

// Pointer to the next node in the list

Node next;

// Constructor to initialize the node with data

public Node(int data){

this.data = data;

this.next = null;

}

}

### Creating an Example Linked List of Size 3 to Understand Working

Create the first node

* Allocate memory for the first node and Store data in it.
* Mark this node as head.

Create the second node

* Allocate memory for the second node and Store data in it.
* Link the first node’s next to this new node.

Create the third node

* Allocate memory for the third node and Store data in it.
* Link the second node’s next to this node.
* Set its next to NULL to ensure that the next of the last is NULL.

public static void main(String[] args) {

// Create the first node (head of the list)

Node head = new Node(10);

// Link the second node

head.next = new Node(20);

// Link the third node

head.next.next = new Node(30);

// Link the fourth node

head.next.next.next = new Node(40);

}

}

Summary

Create a node class. Add item to the node