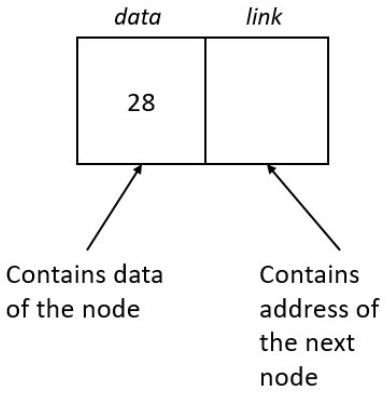
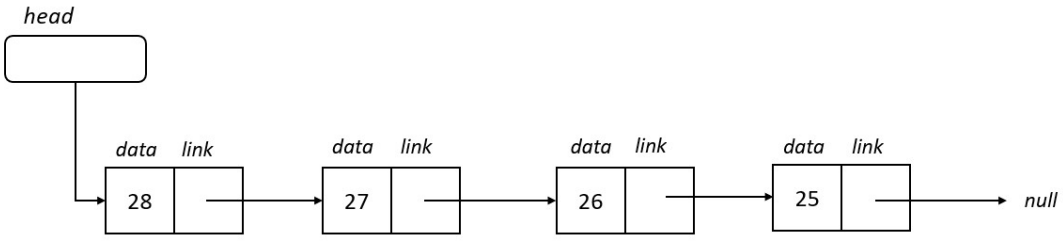
Linked List

1. It's a Physical Data Structure.
2. It is represented in form of Node.
    a. Every Node has two parts
        i. Data
        ii. Pointer to next Node
    b. Node is represented as a Object.
3. Address of the first node is stored in the head. So, if the pointer to head is lost, we lose access to the LinkedList. So, we should be very careful while coding.

Below is the representation of a single Node: -



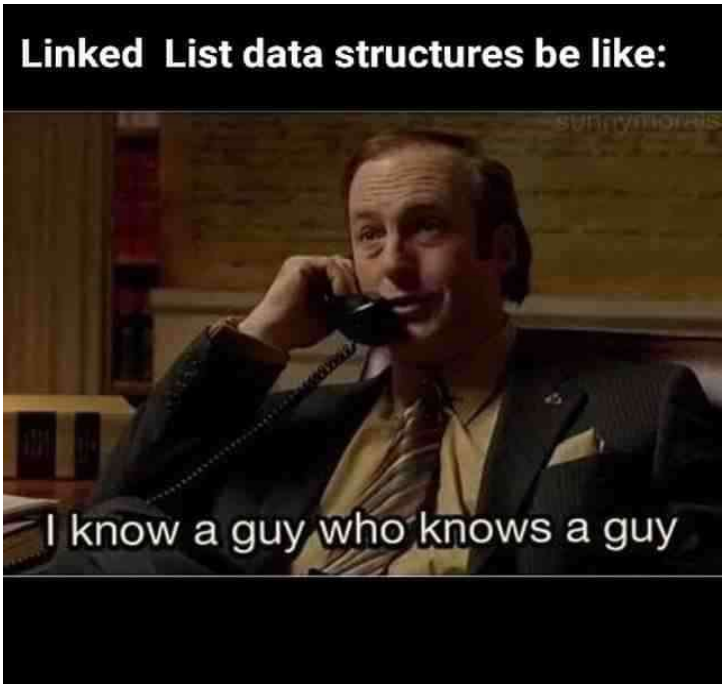
Below is simple representation of a Linked List: -



Simple linked list representation.

Mercy Technologies

Just a Meme: - 😊




```java
public class Node {
	public int data;
	public Node next;

	Node(){

	}
	public Node(int data) {
		this.data = data;
	}
}


public class LinkedListTraversal {
	private void LLTraversal(Node head) {
		if (head == null) // Base condition
			return;
		Node p = head; // Never do any direct operation on head
		while (p != null) {
			System.out.println(p.data);
			p = p.next;
		}
	}
}
```

Mercy Technologies

Class Explanation:

```
private static void linkedListTraversal(Node head) {
    if (head == null) // Base condition
        return;
    Node p = head; // Never do any direct operation on head
    while (p != null) {
        System.out.print(p.data);
        p = p.next;
    }
}
head-->28-->27-->26-->25-->null
                          ^
                          p
Output
======
28 27 26 25
```

Solve below Hacker Rank problem.

https://www.hackerrank.com/challenges/print-the-elements-of-a-linked-list/problem