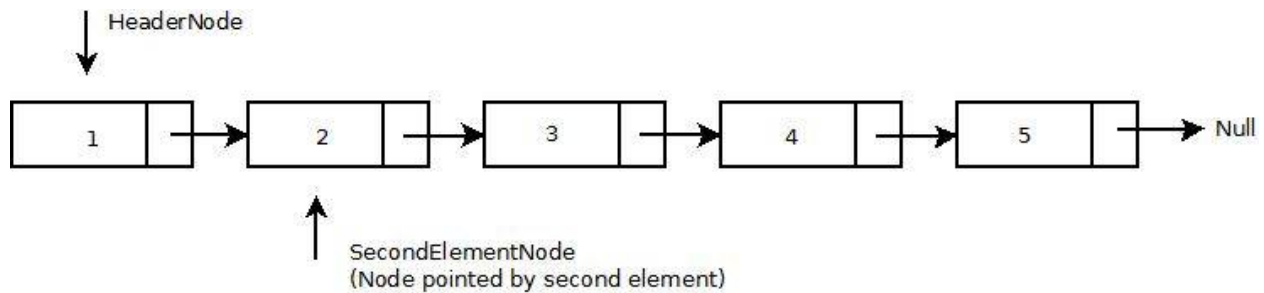


Data Structure: Singly Linked list

Singly linked list is a basic linked list type. Singly linked list is a collection of nodes linked together in a sequential way where each node of singly linked list contains a data field and an address field which contains the reference of the next node.

To perform any operation on a linked list we must keep track/reference of the first node which may be referred by **head pointer variable**. In singly linked list address field of last node must contain a NULL value specifying end of the list.



Basic structure of a singly linked list

Each node of a singly linked list follows a common basic structure. In a node we can store more than one data fields but we need at least single address field to store the address of next connected node.

```
struct node {
    int data;           // Data
    struct node * next; // Address
};
```

Advantages of Singly linked list

There are several points about singly linked list that makes it an important data structure.

- Insertion and deletion of elements doesn't require movement of all elements when compared to an array.
- Can allocate or deallocate memory easily when required during its execution. (Dynamic Memory allocation)

Disadvantages of Singly linked list

- Linked Lists are sequential access while Arrays are random access.
- Extra memory space is required in Linked list for pointers.

DSA Lab Assignment#2 (Batch:Monday 10.00-12.00)

Write C Code for the following:

- Q1.** To create a Singly Linked List to store positive integers in the order they are inputted.
- Q2.** To count the number of nodes in a Singly Linked List.
- Q3.** To split the Singly Linked List (Created in Q1) in two linked list (List-1 should contain only even values while list-2 contains only odd values).
- Q4.** To display all nodes of a given Singly Linked List. (Traversal of Singly Linked List)