# SINGLY LINKED LIST

## 1. What is a Linked List ?

A linked list is a linear data structure in which data is stored in nodes. They are connected together by the address of one node being stored in another.

## 2. What is a singly linked list ?

A singly linked list is a type of linked list in which each node has two fields:
1. The data field to store the data
2. The next pointer to store the address of the next node

## 3. How to insert a new node at the end of a singly linked list ?
1. Traverse the list till you reach the last node.
2. Establish a connection from next of last node to the new node
3. New node now becomes your last node

## 4. How to insert a new node as the head node ?
1. Store the current head node in a temporary node
2. Make the new node as the head node
3. Make the next of new node point to the temporary node

## 5. How to insert a node in between two other nodes ?
1. Traverse till the position where the new node is to be inserted at
2. Store the node prior to this in a temporary node
3. Make the next of temporary node point to the new node
4. Make the next of new node point to this node

## 6. How to delete a node from the end of the list ?
1. Traverse till the end of the list
2. Store the node prior to the last node in a temporary node
3. Make the next of temporary node point to None

## 7. How to delete the head node of a singly linked list ?

When the head node is being removed, the second node needs to be made the head node.
1. Store the current head node in a temporary node
2. Make the second node as the head node
3. Nullify the next of previous head node by making the next of that node point to None

## 8. How to delete a node that is in between two other nodes of a singly linked list
1. Traverse till the position of the node that has to be deleted
2. Store the node prior to this node in a temporary node
3. Make the next of temporary node point to the next of this node
4. Nullify the next of this node by making its next point to None

## 9. What are the limitations of a singly linked list ?

1. It requires an additional storage to store the address of the next node
2. Depending upon the position of a node, the time taken to retrieve it could be costly
3. Revisiting the nodes is limited