Linked Lists

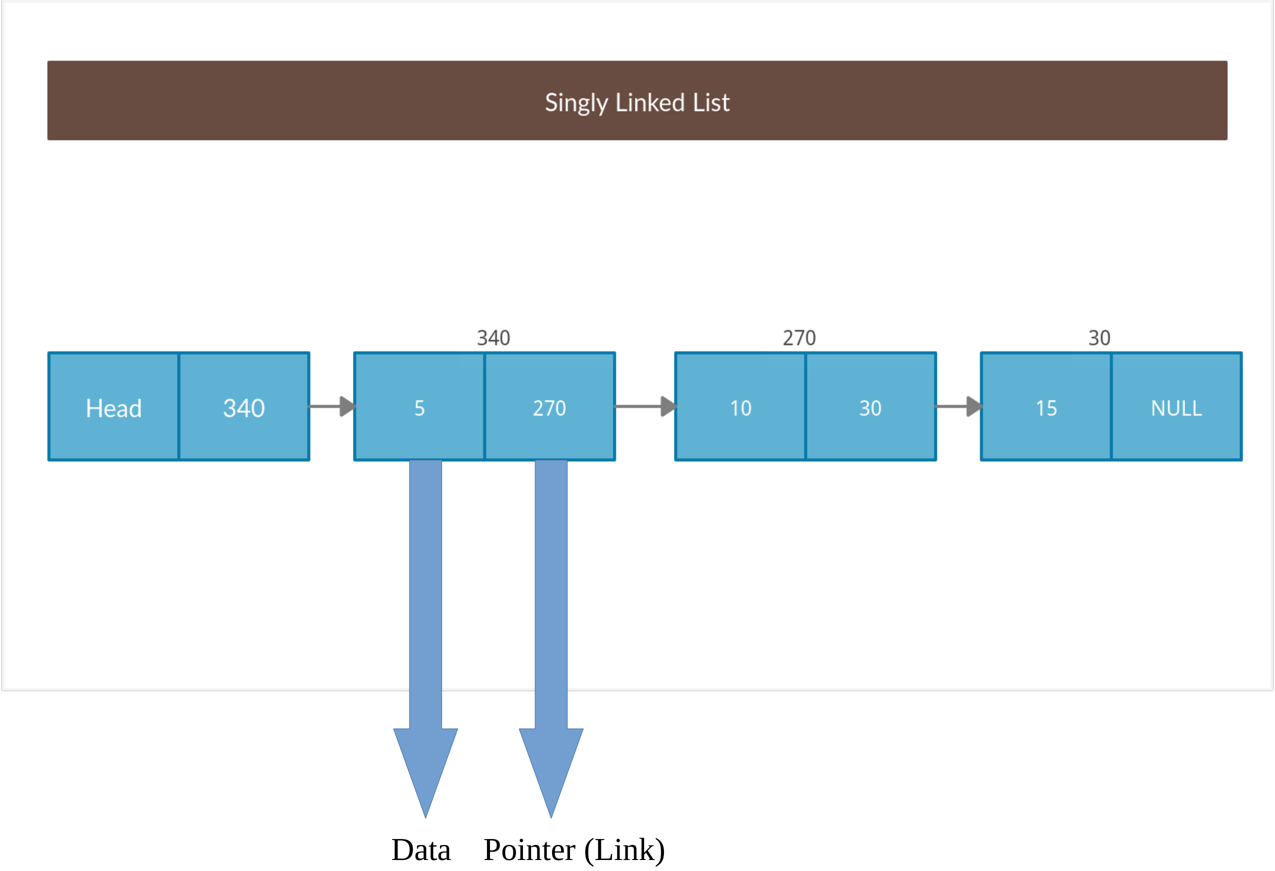
Linear collection of Data points of the same/homogenous data type
Dynamic Data Structure: elements are not stored in contiguous memory space loations

Data Elements in a linked list are called **Nodes**
Each Node consists of a Data and Link
Data represents value stored in the node
Link (Pointer) represents address of next node stored in succession

- Size can grow and shrink as per the data points needed to be stored
- Insertion and deletion are efficient as there is no shifting of elements as in an array
- Extra memory required to store points
- Access of elements through indexing not possible, we need to traverse from the head till the datapoint we are looking for

Singly Linked List

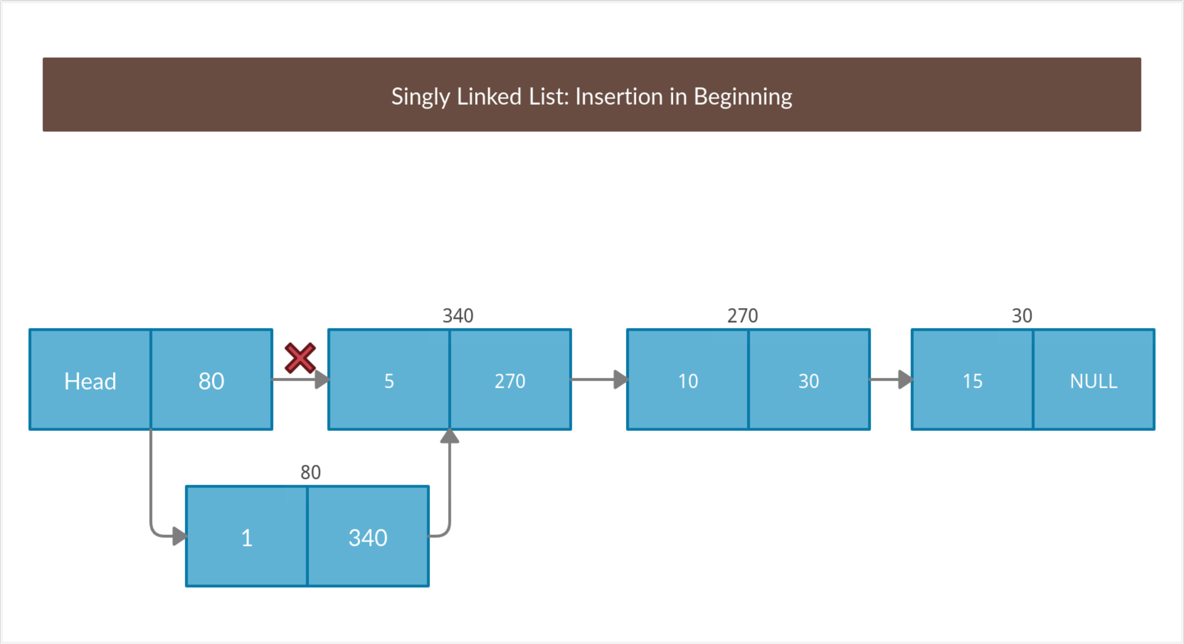
Head points to First Node
Each Node in between is pointed at by the previous node, and itself points at the next node
Last Node points to null



Insertion in Singly Linked List

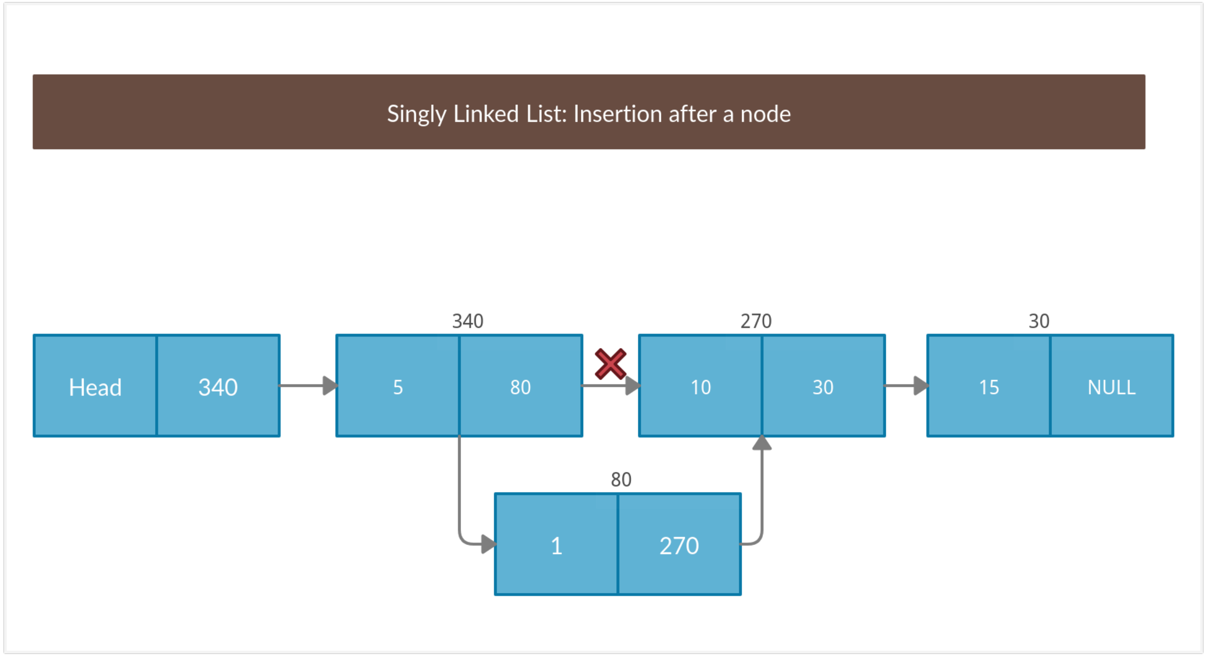
Insertion Complexity for Singly Linked List
At beginning = O(1)
After a node/end = O(n)

1) In the Beginning



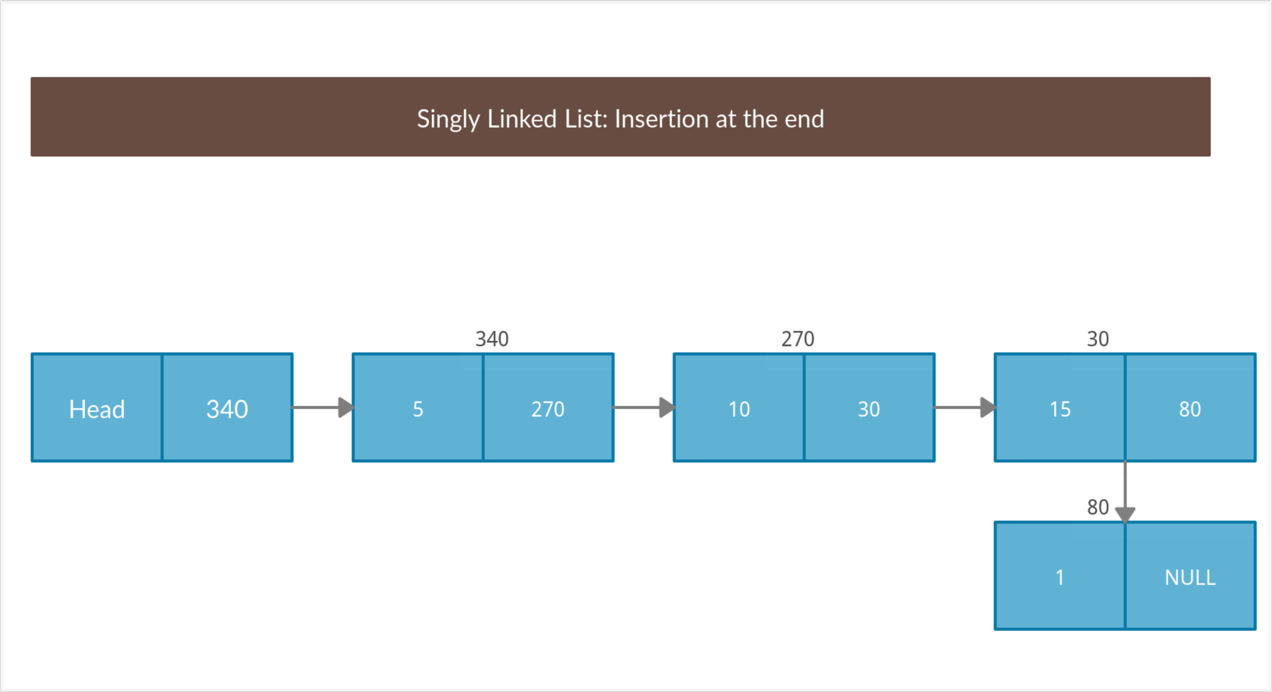
- New Node points where the head node is pointing
- Head node is updated to point at the new node

2) After a Node



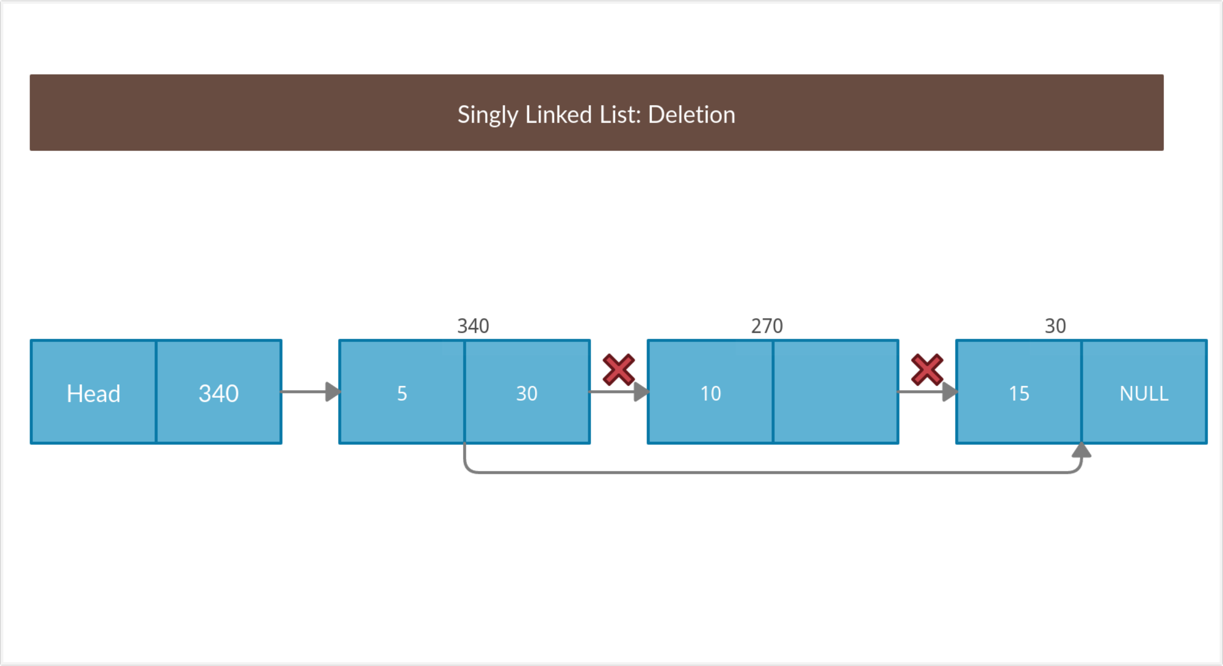
- Let the node after which to insert new element be A
- New node points where A is pointing
- A is updated to point at new node

3) In the end



- The Last node points at the new node
- The new node points at NULL

Deletion



- Let the node to be deleted be B
- Node pointing at B is updated to point at the node that B points to
- Prior Links of B are deleted and Memory space of B is free