

## Linked List

Linked list is a common data structure used in computer science. There are many kinds of linked lists, but we will discuss singly linked lists (SLL) here.

In an SLL, each node is composed of a value 'data' and a pointer 'next' to the next node in the sequence. This structure allows for efficient insertion or removal of elements from any position in the sequence.

SLL's can be used to implement several other common abstract data types, including lists (the abstract data type), stacks, queues, associative arrays (dictionaries), etc.

Generally, we also store a pointer 'head' which points to the first element of the linked list.

The SLL supports the following operations:

**Insert:** We insert a new node = (d,null) into an SLL. Here, d is its data value and its 'next' parameter is null at the moment.

We first make the 'next' pointer of this node to point to the current head, and then the head pointer to point to the new node. This way, we insert the node to the front of the SLL.

**Delete:** We take the node as input whose next node has to be deleted. That way, we just set the value of node.next to node.next.next, and destroy node.next.

**Search:** In order to search the SLL for a value, we start with the head, and loop until the value is found, that is, node.data = value, or return False otherwise.

In each iteration of the loop, we check if the current node's data is equal to the input value, if it is not, we make the current node equal to current node.next.