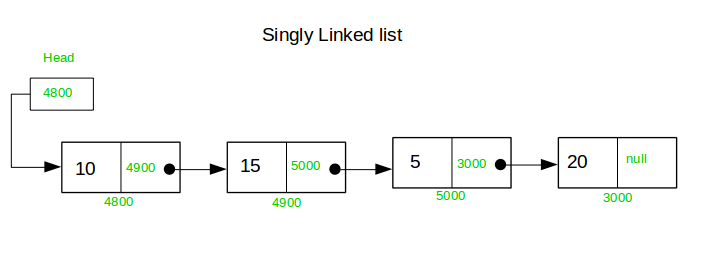
Linked List

* What is Linked List?

Linked list is a sequence of list that connected with each other or linked with other list. Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. A **linked list** is a common data structure made of a chain of nodes in which each node contains a value and a **pointer** to the next node in the chain. The **head pointer** points to the first node, and the last element of the list points to **null**. When the list is empty, the head pointer points to **null**.

* Types of Linked List
* **Single Linked List(SLL)**

In singly linked list lists are connected or linked on only one way or forward way here no two way linking is happen ex- first list is connected with second with the memory location of first one and third list store the second ones memory location. Here reverse linking is not allow if we can do then it will be considered as doubly linked list. Each element in a linked list is called a **node**. A single node contains data and a pointer to the next node which helps in maintaining the structure of the list. A **singly linked list** is a type of linked list that is unidirectional, that is, it can be traversed in only one direction from head to the last node.



The first node is called the **head**; it points to the first node of the list and helps us access every other element in the list. The last node, also sometimes called the **tail**, points to NULL which helps us in determining when the list ends.

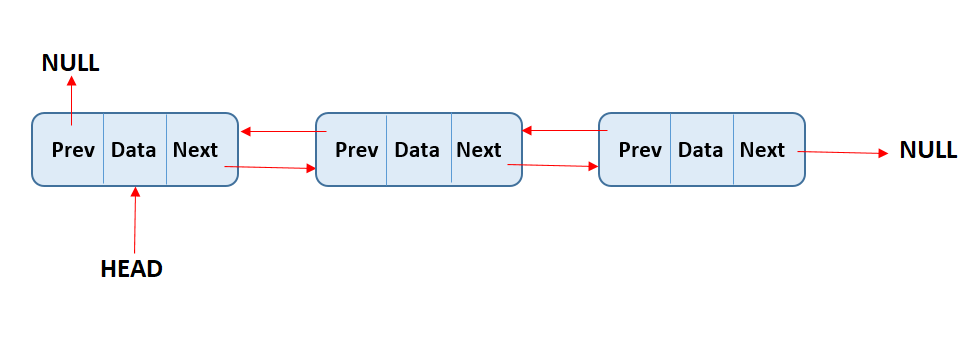
Reference- https://www.educative.io/edpresso/what-is-a-singly-linked-list

* **Doubly Linked List(DLL)**

A doubly linked list contains a pointer to the next node as well as the previous node. This ensures that the list can be traversed in both directions. **doubly linked list** (DLL) shines. DLLs are an extension of basic linked lists with only one difference. Doubly linked list is a same as a singly linked list

DLL node has **three** fundamental members:

* the data
* a pointer to the next node
* a pointer to the previous node



* **Circular Linked List(CLL)**

A circular linked list is a variation of a normal linked list. In a **circular linked list**, as the name suggests, the list does not end; instead, it loops around. The last element of a circular linked list points to the head instead of pointing to null. A circular linked list can be implemented as a singly linked list or a doubly linked list.

There are three fundamental operations that every linked list does:

1. Add element
2. Delete element
3. Display list

Reference -https://www.educative.io/edpresso/what-is-a-circular-linked-list

* Implementation of Linked List in Python

*# Node class*

**class** **Node**:

*# Function to initialize the node object*

**def** **\_\_init\_\_**(self, v):

self**.**val **=** v *# Assign value*

self**.**next **=** None *# Initialize next as null*

*# Linked List class*

**class** **ListNode**:

*# Function to initialize the Linked List*

**def** **\_\_init\_\_**(self):

self**.**head **=** None

reference-https://www.interviewbit.com/tutorial/linked-list-implementation