**\*\*\*\* LINKED LIST WITH IT’S TYPES \*\*\*\***

* A [linked list](https://www.geeksforgeeks.org/data-structures/linked-list/) is a linear data structure, in which the elements are not stored at contiguous memory locations.
* The elements in a linked list are linked using [pointers](https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/).
* In simple words, a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.

**Types Of Linked List:**

**1.**[**Singly Linked List**](https://www.geeksforgeeks.org/data-structures/linked-list/singly-linked-list/)

*It is the simplest type of linked list in which every node contains some data and a pointer to the next node of the same data type.*

The node contains a pointer to the next node means that the node stores the address of the next node in the sequence.

A single linked list allows the traversal of data only in one way.

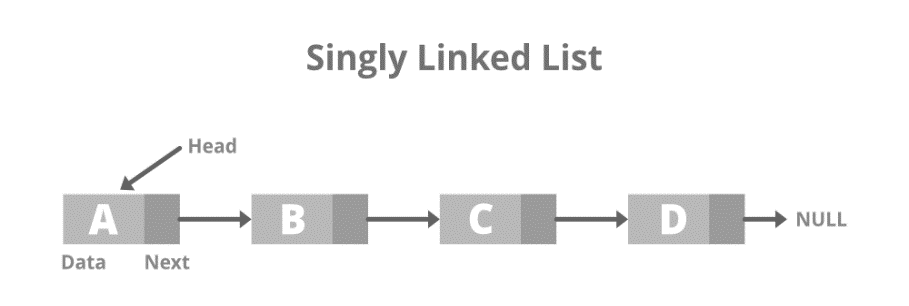
# Node of a singly linked list

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None



### ****2.****[****Doubly Linked List****](https://www.geeksforgeeks.org/doubly-linked-list/)

*A doubly linked list or a two-way linked list is a more complex type of linked list that contains a pointer to the next as well as the previous node in sequence.*

Therefore, it contains three parts of data, a pointer to the next node, and a pointer to the previous node. This would enable us to traverse the list in the backward direction as well.

# structure of Node

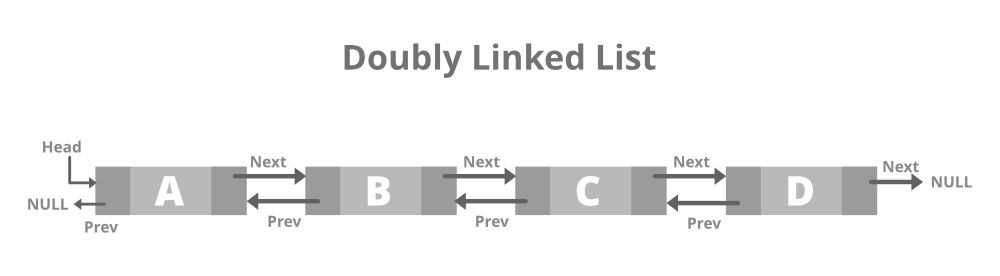
class Node:

def \_\_init\_\_(self, data):

self.previous = None

self.data = data

self.next = None



### ****3.****[****Circular Linked List****](https://www.geeksforgeeks.org/circular-linked-list/)

*A circular linked list is that in which the last node contains the pointer to the first node of the list.*

While traversing a circular linked list, we can begin at any node and traverse the list in any direction forward and backward until we reach the same node we started. Thus, a circular linked list has no beginning and no end.

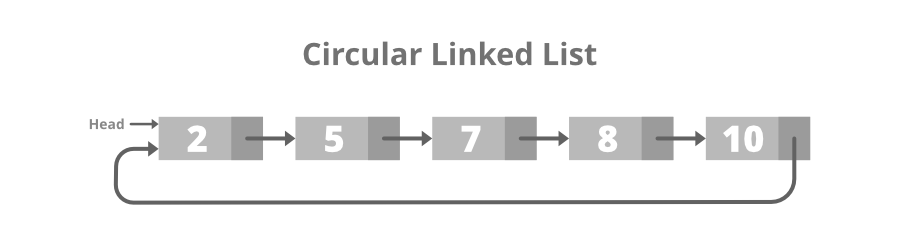
# structure of Node

class Node:

def \_\_init\_\_(self, data):

self.data = data

self.next = None



### 4. [Doubly Circular linked list](https://www.geeksforgeeks.org/circular-doubly-linked-list-meaning/)

A Doubly Circular linked list or a circular two-way linked list is a more complex type of linked list that contains a pointer to the next as well as the previous node in the sequence. The difference between the doubly linked and circular doubly list is the same as that between a singly linked list and a circular linked list. The circular doubly linked list does not contain null in the previous field of the first node.

# structure of Node

class Node:

def \_\_init\_\_(self, data):

self.previous = None

self.data = data

self.next = None

