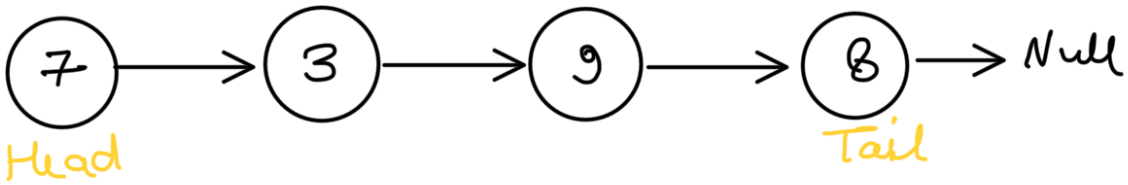
LINKED LIST



Singly Linked List

* Designing a Singly Linked List :

(i) Singly Linked List Object

(ii) get(index)

(iii) addAtHead(value)

(iv) addAtTail(value)

(v) addAtIndex(index, value)

(vi) deleteAtIndex(index)

```
class Node

    def __init__(self, value):
        self.value = value
        self.next = None

class SinglyLinkedList:

    def __init__(self):

        self.head = None
```

```
        self.tail = None
        self.size = 0

    def get(self, index):

        if index < 0 or index >= self.size:
            return -1

        counter = 0
        current = self.head

        while counter != index:
            current = current.next
            counter += 1
        return current

    def addAtHead(self, value):

        node = Node(value)
        if not self.head:
            self.head = node
            self.tail = node
        else:
            node.next = self.head
            self.head = node

        self.size += 1

    def addAtTail(self, value):

        node = Node(value)
        if not self.head:
            self.head = Node
            self.tail = Node
        else:
            self.tail.nxt = Node
            self.tail = Node
```

```
        self.size += 1

    def addAtIndex(self, index, value):

        if index < 0 or index > self.size:
            return 'invalid index'

        if index == 0:
            return self.addAtHead(value)

        if index == self.size:
            return self.addAtTail(value)

        node = Node(value)
        prev = self.get(index-1)
        temp = prev.next
        prev.next = node
        node.next = temp
        self.size += 1

    def deleteAtIndex(self, index):

        if index<0 or index >= self.size:
            return 'invalid index'

        if index == 0:
            temp = self.head
            self.head = temp.next
            self.size -= 1
            if self.size == 0:
                self.tail = None
            return temp.value

        if index == self.size - 1
            old_tail = self.tail
            new_tail = self.get(index-1)
            self.tail = new_tail
```

```
        new_tail.next = None
        self.size -= 1
        return old_tail.value

    prev = self.get(index - 1)
    deleted_node = prev.next
    prev.next = deleted_node.next
    self.size -= 1
    return deleted_node.value
```