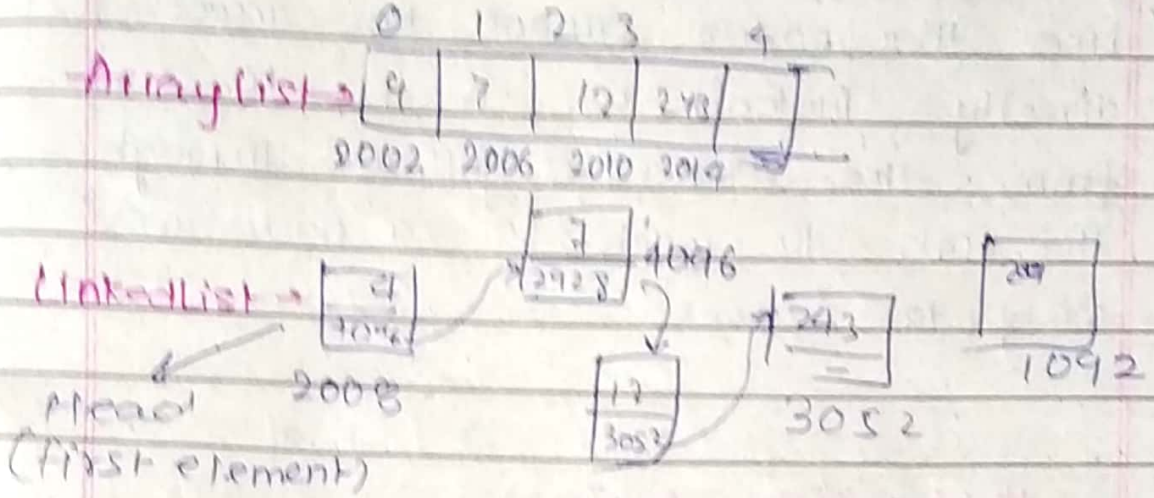
LinkedList



In arraylist, elements are in continguous fashion.

But not in linkedlist, they are not in contiguous fashion in memory

⇒ LinkedList are linear data structures where the elements are not stored in contiguous locations and every elements is a separate object with a data part and address part.
The elements are linked using pointers and addresses. Each element is known as node. Due to the ~~dynamically~~ dynamicity and ease of insertions and deletions, they are prefferred over the arrays.

It also has a few disadvantages like the nodes cannot be accessed directly instead we need to start from the head & follow through the link to reach to a node we wish to access.

LinkedList in Java.

To store the elements in a linked list we use a doubly linked list which provides a linear data structure and also used to inherit an abstract class and implement list and deque interfaces. In Java, LinkedList class implements the list interface.

The LinkedList class also consists of various constructors and methods like other Java collections.

```
import java.util.*;

package LinkedLists;
public class MainLinkedList {

    public static void main(String[] augs){

        List<Integer> ll = new LinkedList<>();
        ll.add(12);
        ll.add(2);
        ll.add(32);

        System.out.println(ll);
    }
}
```

output

[12, 2, 32]

```
package LinkedLists;
import java.util.*;

public class MainLinkedList {

    public static void main(String[] args){

        List<Integer> ll = new LinkedList<>();
        List<Integer> al = new ArrayList<>();

        System.out.println(ll.get(1));
        getTimeDiff(al);
        getTimeDiff(ll);
    }
    static void getTimeDiff(List<Integer>
                                    list){

        long start = System.currentTimeMillis();

        for(int i=0; i<10000; i++){
            list.add(i);
        }
        long end = System.currentTimeMillis();

        System.out.println(list.getClass().getName()
                + " " + (end - start));
    }
```

JoAdd ☺

Code of Linked list

```
package linkedList;

public class MyLinkedList {
    Node head;
    static class Node {
        int data;
        Node next;

        public Node(int data) {
            this.data = data;
            next = null;
        }
    }

    void add(int data) {
        Node toAdd = new Node(data);
        if(head == null) { head = toAdd; return; }
        Node temp = head;
        while (temp.next != null) {
            temp = temp.next;
        }
        temp.next = toAdd;
    }
}
```

```
void print() {
    Node temp = head;
    while (temp != null) {
        System.out.println(temp.data + " ");
        temp = temp.next;
    }
}
```