



# ECAP770

## ADVANCE DATA STRUCTURES

---

**Ashwani Kumar**  
Assistant Professor

# Learning Outcomes



After this lecture, you will be able to

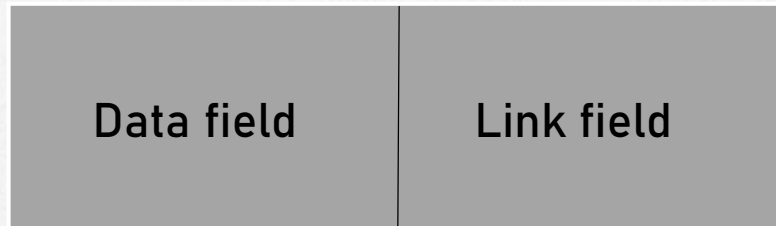
- Know about linked List

# Linked list

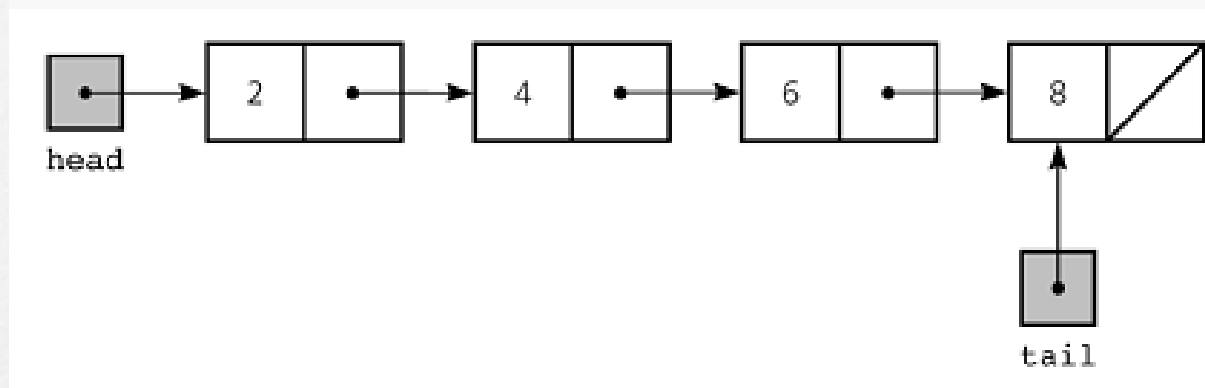
- linked list is a linear data structure
- Elements are not stored at contiguous memory locations
- A node contains two fields: where each node contains a data field and a reference(link) to the next node in the list.

# Linked list

- A linked list is collection of nodes: Data field and address field/ link field.



# Linked list



# Why linked list

Due to the shortcomings in arrays

- Size
- Memory allocation
- Memory efficiency
- Execution time



# Arrays vs. Linked list

## Arrays

Data elements are stored in contiguous locations in memory.

Insertion and Deletion operations are costlier since the memory locations are consecutive and fixed.

Memory is allocated during the compile time (Static memory allocation).

Size of the array must be specified at the time of array declaration/initialization.

## Linked list

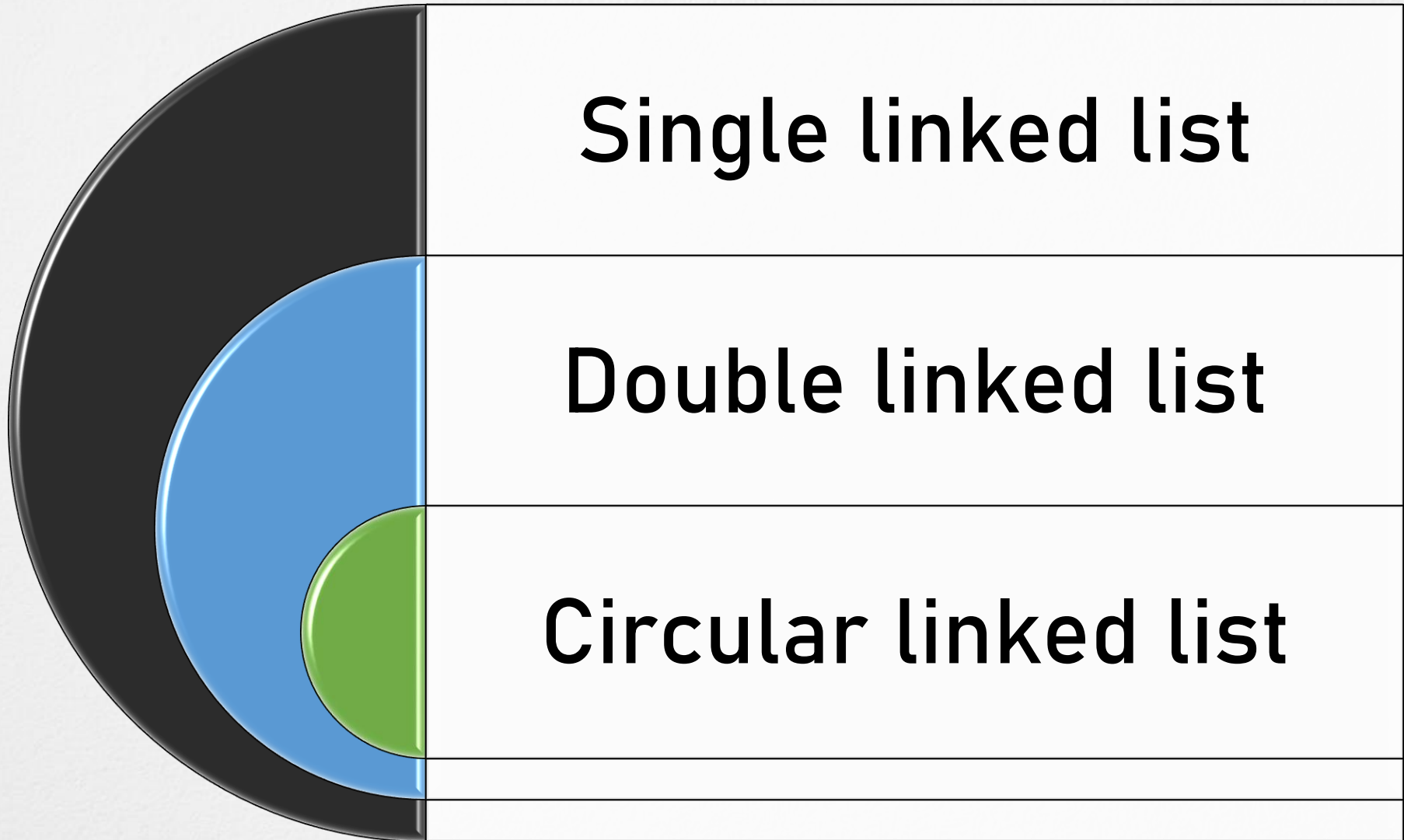
New elements can be stored anywhere and a reference is created for the new element using pointers.

Insertion and Deletion operations are fast and easy in a linked list.

Memory is allocated during the run-time (Dynamic memory allocation).

Size of a Linked list grows/shrinks as and when new elements are inserted/deleted.

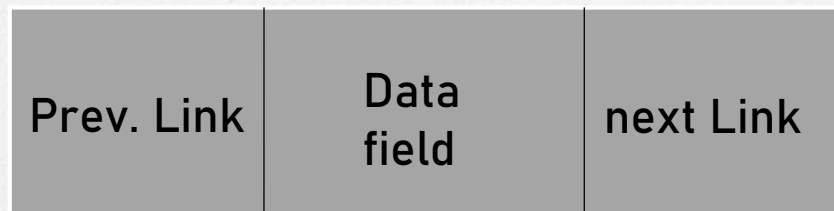
# Types of linked list



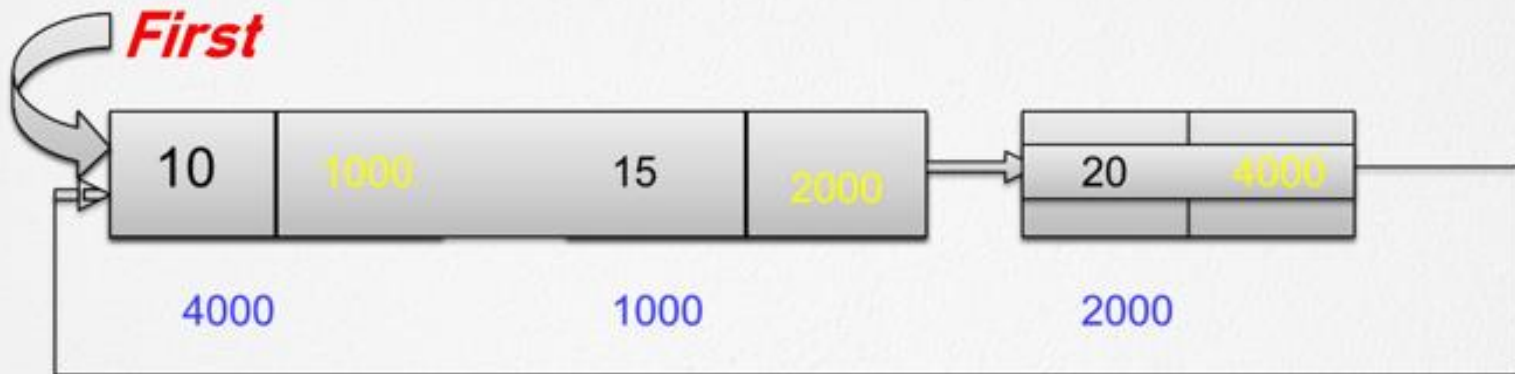


# Double linked list

- A doubly-linked list is a linked data structure that consists of a set of sequentially linked records called nodes.
- Each node contains two fields, called links, that are references to the previous and to the next node in the sequence of nodes.



# Circular linked list



# Advantages of Linked Lists

- Size of linked lists is not fixed, they can expand and shrink during run time.
- Insertion and Deletion Operations are fast and easier in Linked Lists.
- Memory allocation is done during run-time (no need to allocate any fixed memory).
- Data Structures like Stacks, Queues, and trees can be easily implemented using Linked list.

# Program

```
#include<stdio.h>

#include<stdlib.h>

struct Node{

int data;

struct Node* next;

};

main(){

struct Node* head=NULL;

struct Node* second= NULL;

struct Node* third= NULL;
```

# Program

```
head=(struct  
Node*)malloc(sizeof(struct Node));
```

```
second=(struct  
Node*)malloc(sizeof(struct Node));
```

```
third=(struct  
Node*)malloc(sizeof(struct Node));
```

```
head->data=1;
```

```
head->next=second;
```

```
second->data=2;
```

```
second->next=third;
```

```
third->data=3;
```

- third->next=NULL;
- printf("List created");
- }



That's all for now...