

- Inhärent dynamische Datenstruktur zur Verwaltung einer Folge von Elementen
- je verwaltetem Element ein Listenknoten, der neben dem Element einen Verweis (Zeiger) auf den jeweils nächsten Listenknoten umfasst
- Dynamische Reservierung des Speichers für einen Listenknoten beim Einfügen eines Elementes, Freigabe beim Entfernen eines Elementes

```
struct ListNode {  
    ??? payload;  
    struct ListNode next;  
};
```

*Hier irrelevant*



*/\* struct ListNode ist noch unvollständig  
=> Größe von next nicht berechenbar \*/*

```
struct ListNode {  
    ??? payload;  
    struct ListNode *next;  
};
```



*/\* OK: Größe von next berechenbar, da  
Zeigertyp \*/*

```
struct ListNode {  
    ??? payload;  
    struct ListNode *next;  
};  
  
/* eigener Typname für Listen */  
typedef struct ListNode *List;
```



```
/* forward reference */  
typedef struct ListNode *List;  
  
struct ListNode {  
    ??? payload;  
    List next;  
};
```

*Vorwärtsreferenz gültig  
weil Zeigertyp*

