

```
typedef char          *Element;  
typedef struct ListNode *List;  
struct ListNode { Element elem; List next; };
```

- Repräsentation der leeren Liste als NULL-Zeiger

## Leere Liste

```
List emptyList() {  
    return NULL;  
}
```

## Test auf leere Liste

```
int isEmpty(List l) {  
    return l == emptyList();  
}
```

- Erzeugen eines Listenknotens

```
List cons(List l, Element e) {  
    List res = malloc( sizeof(struct ListNode) );  
    /* oder: malloc( sizeof(*l) ); */  
    if(res == NULL) {  
        exit(1);  
    }  
    res->elem = e;  
    res->next = l;  
    return res;  
}
```

Standardfehler sizeof(List) würde  
nur die Größe vom Listepointers liefern  
• Typen vor & nach malloc MÜSSEN  
sich unterscheiden

