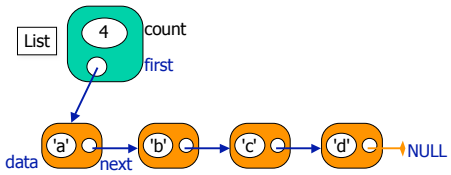
Beispiel: Verkettete Listen (Linked Lists)

- Sehr häufige dynamische Datenstruktur
- Besteht aus Folge von (gleichartigen) Elementen
  - Jedes kennt Vorgänger und Nachfolger
  - Man kennt den Anfang (Kopf, head) der Liste

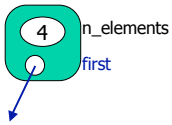


- Ein Element:

```
struct ListElement
{
    float x, y;
    int z;
    ListElement* next;
};
```

- Der "Anker":

```
struct List
{
    ListElement* first;
    ListElement* last;
    int n_elements;
    ...
};
```

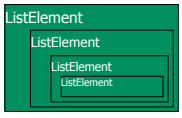


- Achtung:

```
struct ListElement
{
    float x, y;
    int z;
    ListElement next;
};
```

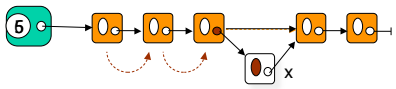
klappt nicht (* fehlt)!

- Das wäre eine "rekursive" Datenstruktur ☺



Einfügen

- Durch "Umbiegen"der Zeiger




```
// insert x (= ListElement*) after n-th element
ListElement* e = list.first;
int i = 1;
while ( i < n && e->next )
{
    i ++ ;
    e = e->next;
}
// Nachbedinung: e zeigt auf Elem.,
// hinter dem x eingefuegt werden soll
X->next = e->next;
e->next = x;
```

```
// insert x (= ListElement*) after n-th element
if ( n == 0 )
{
    // Einfügen als neuer Kopf der Liste
    x->next = list.first;
    list.first = x;
}
else
{
    ListElement* e = list.first;
    int i = 1;
    while ( i < n && e->next )
    {
        i ++ ;
        e = e->next;
    }
    // Nachbedinung: e zeigt auf Elem.,
    // hinter dem x eingefuegt werden soll
    x->next = e->next;
    e->next = x;
}
```

Code mit "Randfall"

Entfernen (ohne Löschen)

- Durch analoges Umbiegen der Zeiger