

---

# 数据结构

## 线性表之单链表

---

创客学院 小美老师

# 线性表的链式存储结构

将线性表 $L=(a_0, a_1, \dots, a_{n-1})$ 中各元素分布在存储器的不同存储块，称为结点，通过地址或指针建立元素之间的联系



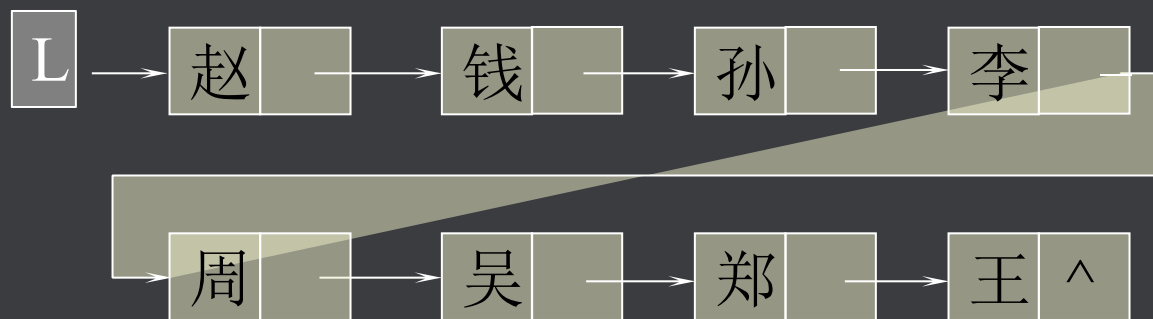
结点的data域存放数据元素 $a_i$ ，而next域是一个指针，指向 $a_i$ 的直接后继 $a_{i+1}$ 所在的结点。



# 线性表的链式存储结构

设线性表 $L=(\text{赵}, \text{钱}, \text{孙}, \text{李}, \text{周}, \text{吴}, \text{郑}, \text{王})$ , 各元素在存储器中的分布如图

地址	data	Next
100	李	142
106	钱	112
112	孙	100
118	王	NULL
124	吴	136
130	赵	106
136	郑	118
142	周	124



带头结点的单链表:



# 线性表的链式存储结构

结点类型描述：

```
typedef struct node
{
    data_t data; //结点的数据域//
    struct node *next; //结点的后继指针域//
}listnode, *linklist;
```

若说明

listnode A;

linklist p = &A;



# 线性表的链式存储结构

设p指向链表中结点 $a_i$



获取 $a_i$ , 写作:  $p \rightarrow \text{data}$ ;

而取 $a_{i+1}$ , 写作:  $p \rightarrow \text{next} \rightarrow \text{data}$

若指针 $p$ 的值为NULL, 则它不指向任何结点, 此时取 $p \rightarrow \text{data}$ 或 $p \rightarrow \text{next}$ 是错误的。

# 线性表的链式存储结构

可调用C语言中malloc()函数向系统申请结点的存储空间

```
linklist p;
```

```
p = (linklist)malloc(sizeof(listnode));
```

则创建一个类型为linklist的结点，且该结点的地址已存入指针变量p中：



扫一扫，获取更多信息



THANK YOU