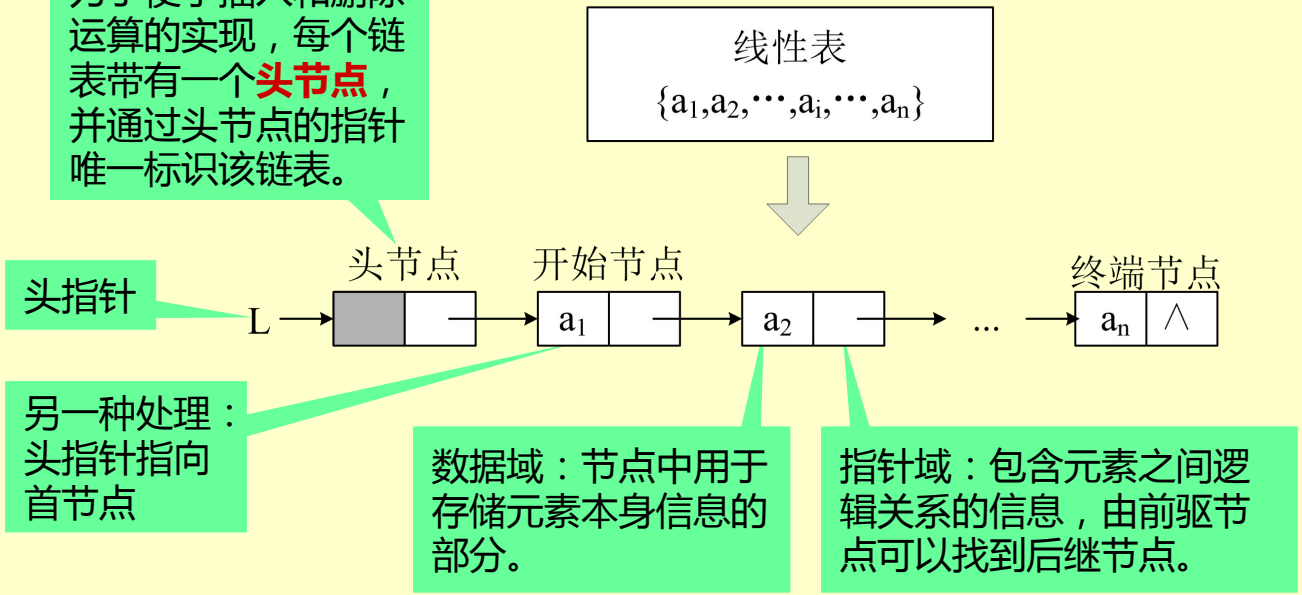
本节主题：

线性表的链式存储

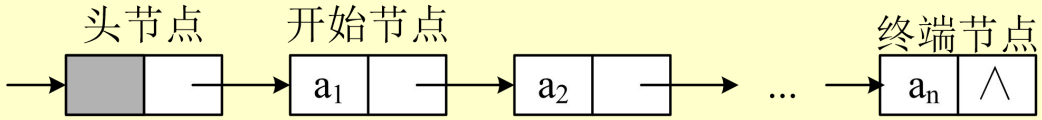
认识链表

为了便于插入和删除运算的实现，每个链表带有一个**头节点**，并通过头节点的指针唯一标识该链表。



- 在每个节点中除包含有数据域外，只设置一个指针域，用以指向其后继节点，这样构成的链接表称为**线性单向链接表**，简称**单链表**。
- 其他链表
  - 双链表
  - 循环链表
  - ……

单链表的存储结构




```
typedef int ElemType;  //数据域可以为其他类型
typedef struct LNode   //定义单链表节点类型
{
   ElemType data;     //数据域
   struct LNode *next; //指针域，指向后继节点
} LinkList;
```

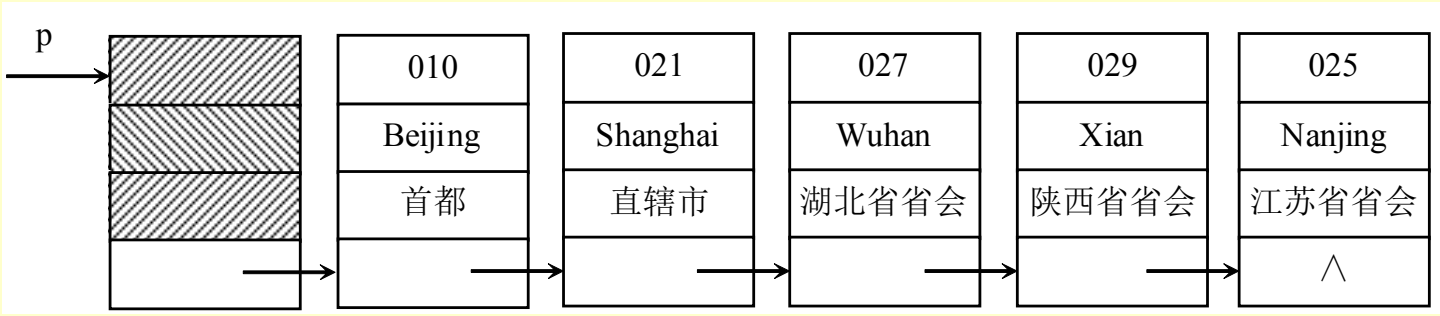
- 在单链表中，由于每个节点只包含有一个指向后继节点的指针，所以当访问过一个节点后，只能接着访问它的后继节点，而无法访问它的前驱节点。

示例

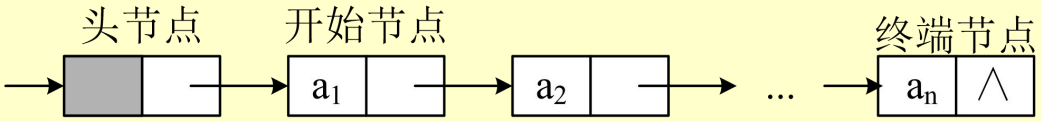
区号	城市名	说 明
010	Beijing	首都
021	Shanghai	直辖市
027	Wuhan	湖北省省会
029	Xian	陕西省省会
025	Nanjing	江苏省省会

```
typedef struct
{
    char code[4];
    char name[16];
    char describe[32];
}ElemType;
```

```
typedef struct LNode
{
    ElemType data;
    struct LNode *next;
} LinkList;
```



存储密度



- 存储密度：节点数据本身所占的存储量和整个节点结构中所占的存储量之比。

$$\text{存储密度} = \frac{\text{节点数据本身占用的空间}}{\text{节点占用的空间总量}}$$

- 意义
  - 存储密度越大，存储空间的利用率就越高。
- 比较
  - 顺序表的存储密度为1（若不考虑顺序表中的空闲区）
  - 链表的存储密度小于1
    - 若单链表的节点数据均为整数，指针所占的空间和整数相同，则单链表的存储密度为50%
    - 数据域所占空间越多，存储密度越高