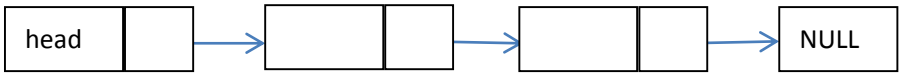
链表原理

(1) 链表:

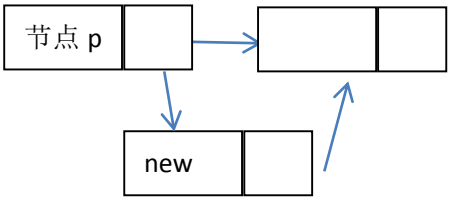
单链表：

```
struct node {
    int data;              //数据域，可以是任意数据类型
    struct node *next;     //指针域
}
```



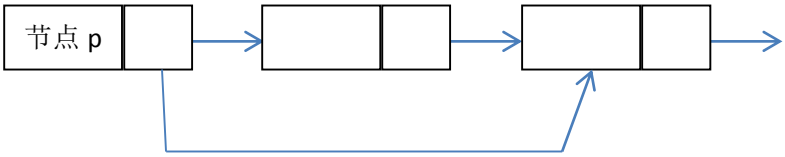
初始化： 申请内存空间 struct node *h; h->next = NULL;

插入:




```
New->next = p->next;
p->next = new;
```

删除：




```
q = p->next;
p->next = q->next;
free(q);
```

遍历：

```
P = head->next;   //head 是头节点
While (p != NULL)
{
    //操作
    P = p->next;
}
```

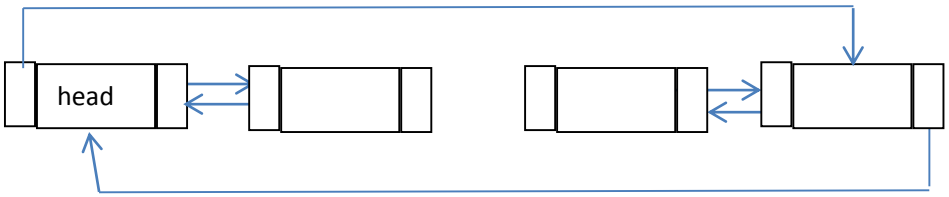
单循环链表： 将单链表的最后一个节点指向头节点，其他操作跟单链表一样

遍历的判断：

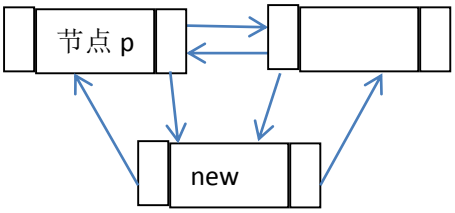
```
P = head->next;
while (p != head);
```

双循环链表：

```
struct node {
        int data;
        struct node *prev, *next;
};
```

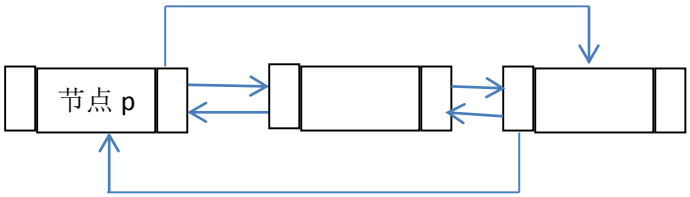


插入：




```
New-next = p->next;
p->next->prev = new;
new->prev = p;
p->next = new;
```

删除：




```
 q = p->next;
 p->next = q->next;
 q->next-prev = p;
 free(q);
```

遍历：

```
p = head->next //head 是头节点
while (p != head)
{
   //操作
   p = p->next;
}
```