

## 单向链表

原博客来自: [C++ 单链表基本操作分析与实现 - Sam大叔 - 博客园 \(cnblogs.com\)](http://cnblogs.com)

### 链表的定义

链表是一种物理存储上单元上非连续, 非顺序的存储结构, 与之前所提及的顺序表不同, 其数据元素的逻辑顺序是通过链表中的指针链接次序实现的。其物理上并不是存储在一块连续的内存上, 而是分开存储, 依靠每个结点中存有下一个结点的地址进行访问。链表由一系列结点组成, 每个结点包括两个部分: 一个是存储数据元素的数据域, 另一个是存储下一个结点的指针域, 对比线性表, 链表能够更方便地进行插入和删除操作。

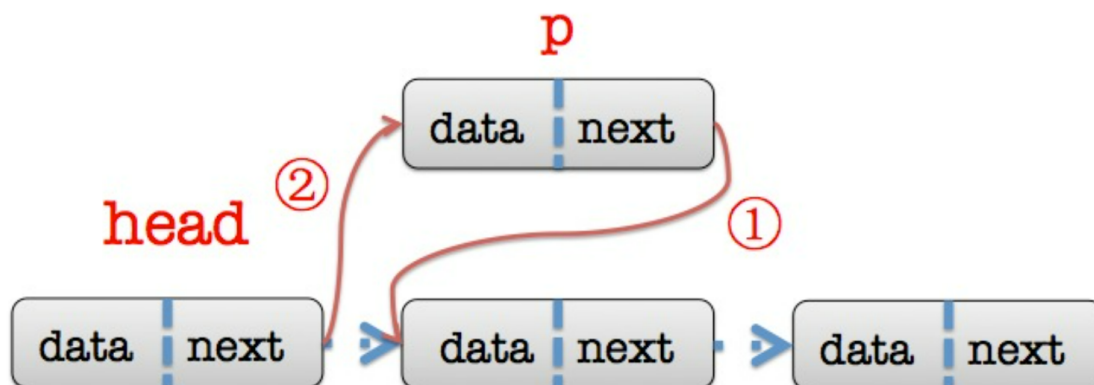
### 创建头结点

在堆区开辟一块内存存放Node, 将Node的next置为NULL



### 从头插入一个新的结点

new一个新的结点p, 将p的next指向head->next指向的地址, 然后将head->next重新指向p。



### 查找数据d的上一个结点的位置

先定义一个结点, 另其为头结点位置, 然后遍历整个链表, 如果有 $p \rightarrow next \rightarrow data == d$ , 那么就终止循环, 返回p

### 删除指定结点

先遍历到指定结点的前一个结点, 然后通过将前一个结点的next指针指向指定结点的下一个结点, 也就是略过欲删除的指定结点, 直接指向下一个结点, 然后删除指定结点。



### 修改指定结点

先通过查找算法找到d的上一个结点位置，然后将其data修改成指定值即可



### 链表反转

为了能够反转链表，那么先定义三个临时结点指向头结点之后的第一个结点p，第二个结点q和第三个结点m，将p->next置为空，然后将q->next置为p，然后将p向后移动一个结点，也就是p=q，然后再将q向后移动一个结点，即q=m，最后把m向后移动一个结点，m = m->next。一直循环直到m=NULL，最后一步将q->next = p。那么此时q变为第一个元素，head->next = q

