

## 创建节点

```
//定义一个结点模板
template<typename T>
struct Node {
    T data;
    Node *next;
    Node() : next(nullptr) {}
    Node(const T &d) : data(d), next(nullptr) {}
};
```

## 删除 p 结点后面的元素

```
template<typename T>
void Remove(Node<T> *p) {
    if (p == nullptr || p->next == nullptr) {
        return;
    }
    auto tmp = p->next->next;
    delete p->next;
    p->next = tmp;
}
```

## 在 p 结点后面插入元素

```
template<typename T>
void Insert(Node<T> *p, const T &data) {
    auto tmp = new Node<T>(data);
    tmp->next = p->next;
    p->next = tmp;
}
```

## 遍历链表

```
template<typename T, typename V>
void Walk(Node<T> *p, const V &vistor) {
    while(p != nullptr) {
        vistor(p);
        p = p->next;
    }
}

int main(){
    int sum = 0;
    Walk(p, [&sum](const Node<int> *p) -> void { sum += p->data; });
}
```