

# C++ list::splice()函数详解

list::splice实现list拼接的功能。将源list的内容部分或全部元素删除，拼插入到目的list。

函数有以下三种声明：

一：void splice ( iterator position, list<T,Allocator>& x );

二：void splice ( iterator position, list<T,Allocator>& x, iterator it );

三：void splice ( iterator position, list<T,Allocator>& x, iterator first, iterator last );

解释：

position 是要操作的list对象的迭代器

list&x 被剪的对象

对于一：会在position后把list&x所有的元素到剪接到要操作的list对象

对于二：只会把it的值剪接到要操作的list对象中

对于三：把first 到 last 剪接到要操作的list对象中

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    list<int>li1,li2;
    for(int i=1;i<=4;i++) li1.push_back(i),li2.push_back(i+10);
    // li1 1 2 3 4
    // li2 11 12 13 14
    list<int>::iterator it=li1.begin();
    it++;

    li1.splice(it,li2);//1 11 12 13 14 2 3 4
    if(li2.empty()) cout<<"li2 is empty"<<endl;

    li2.splice(li2.begin(),li1,it);
    cout<<*it<<"    chen"<<endl;
    /*
    li1 1 11 12 13 14 3 4
    li2 2
    这里的it的值还是2 但是指向的已经是li2中的了
    */

    it=li1.begin();
    advance(it,3);//advance 的意思是增加的意思，就是相当于 it=it+3;这里指向13
    li1.splice(li1.begin(),li1,it,li1.end()); //13 14 3 4 1 11 12 可以发现it到li1.end()被剪贴到li1.begin()前面了
    for(list<int>::iterator it=li1.begin();it!=li1.end();++it) cout<<*it<<"    ";
    cout<<endl;
    for(list<int>::iterator it=li2.begin();it!=li2.end();++it) cout<<*it<<"    ";
    return 0;
```