

向量

向量是一个能存放任意类型的动态数组

定义vector:

```
#include<vector>
```

导入vector

```
vector<int> v;
```

动态数组：元素个数 可变

函数：push_back()增加元素到向量后面

```
v.push_back(10);
```

↵

▪ 例子：vector 的存储和遍历 ↵

注意掌握 vector 构造的 4 个常见方法： ↵

- (1) vector():创建一个空 vector ↵
- (2) vector(n):创建一个元素个数为 n 的 vector ↵
- (3) vector(n,t):创建一个元素个数为 n 且值为 t 的 vector ↵
- (4) vector(begin,end):复制[begin,end)区间内另一个数组的元素到 vector 中 ↵

↵

查看vector中的长度

```
cout << v.size()<<endl;
```

当vector中有元素时，可以用下标访问

```
for(int i=0;i<v.size;i++){  
    cout << v[i] << " ";  
}
```

创建一个元素个数为10的vector

```
vector<int> v(10);
```

创建一个元素个数为10的vector长度为80

```
vector<int> v(10,80);
```

用数组初始化vector

```
int a[] = {10,20,30,40,50};  
vector<int> v(a,a+5);
```

sizeof: 计算变量占用的字节数

```
sizeof(a)/sizeof(int)
```

计算数组实际长度

insert(位置, 个数 n, 元素) ↵ 插入 n 个相同的元素到指定位置 ↵

```
vector<int> v;  
v.push_back(10);  
v.push_back(20);  
v.push_back(30);  
v.insert(v.begin()+1,100); //向vector中下标为1的位置插入元素100  
v.insert(v.begin()+1,5,100); //向vector中下标为1的位置插入五个100
```

insert(位置, 元素): 必须提供位置指针

↵

▪ 例子: vector 的存储和遍历 ↵

注意掌握 vector 构造的 4 个常见方法: ↵

- (1) vector(): 创建一个空 vector ↵
- (2) vector(n): 创建一个元素个数为 n 的 vector ↵
- (3) vector(n,t): 创建一个元素个数为 n 且值为 t 的 vector ↵
- (4) vector(begin,end): 复制[begin,end)区间内另一个数组的元素到 vector 中 ↵

↵

```
vector<int> v2;  
  
v2.push_back(100);  
v2.push_back(200);
```

```
v2.push_back(300);
```

```
v.insert(v.begin()+1,v2.begin(),v2.end());//向vector下标为1的位置插入v2中下标为1到最后所有元素
```

##删除下标元素

```
v.erase(v.begin()+1);//删除下标为1的元素
```

从向量中下标为1的元素删除到最后

```
v.erase(v.begin+1,v.end());
```

首尾元素

第一个元素：

```
v.front();//一般用法：cout << "第一个元素"<<v.front()<<endl;
```

最后一个元素：

```
v.back();//一般用法：cout << "最后一个元素"<<v.back()<<endl;
```

[rbegin\(\)](#) ↗

反向迭代器，指向最后一个元素 ↗

[rend\(\)](#) ↗

反向迭代器，指向第一个元素之前的位置 ↗

交换下标元素

```
swap(v1,v2);
```

注意：这里的swap函数不是vector的特有的函数，而是公共的一个函数

例子：

```
vector<int> v1;  
v1.push_back(10);  
v1.push_back(20);  
v1.push_back(30);  
  
vector<int> v2;  
v2.push_back(100);
```

```
v2.push_back(200);  
v2.push_back(300);  
  
swap(v1,v2);
```

排序下标元素

```
sort(v.begin(),v.end())
```

例子：

```
int a[] = {5,3,4,1,2};  
vector<int> v(a,a+5);  
sort(v.begin(),v.end());
```

或者

```
reverse(v.begin(),v.end());
```

所以，上面的例子我们还可以这样排序

```
int a[] = {5,3,4,1,2};  
vector<int> v(a,a+5);  
reverse(v.begin(),v.end());
```

`v.resize(20);`

重置vector的元素大小为20

二维vector

```
vector<vector <int> > v;
```

注意：外层的>要加一个空格，因为>>表示右移运算符

初始化二维vector的大小

```
vector<vector <int> > v(20);
```

添加元素到二维vector中

```
for(int i=0;i<5;i++){
    for(int j=0;j<5;j++){
        v[i].push_back(i+j);
    }
}
```

```
for(int i = 0;i < 5;i++){
    for(int j = 0;j < 5;j++){
        cout<<v[i][j]<<" ";
    }
    cout<<endl;
}
```

遍历二维vector

vector STL函数速查表

函数名	函数说明
push_back(元素)	增加一个元素到向量后面
insert(位置,元素)	插入元素到向量的指定位置
insert(位置,个数n,元素)	插入n个相同的元素到指定位置
insert(位置,向量头指针first, 尾指针end)	将另一个向量从first开始到end结束（不包括end）之间的内容插入该向量的指定位置。
erase(位置)	删除指定位置的元素
erase(开始位置,结束位置)	删除向量中[first,last)中元素
pop_back()	弹出（删除）向量的最后一个元素
clear()	清除向量所有元素，size()变为0
运算符[i]	取向量下标为i的元素
front()	取向量第一个元素
back()	取向量最后一个元素
begin()	返回向量头指针（迭代器），指向第一个元素
end()	返回向量尾指针，指向向量最后一个元素的下一个位置
rbegin()	反向迭代器，指向最后一个元素
rend()	反向迭代器，指向第一个元素之前的位置
size()	返回向量中实际元素的个数。
resize(大小)	重新设定向量的大小，也就是可以保存元素的个数。
max_size()	得到vector最大可以是多大。
empty()	判断向量是否为空，等价于size()为0。
swap()	交换两个同类型向量的数据。