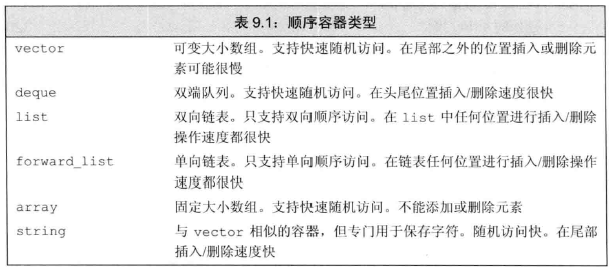
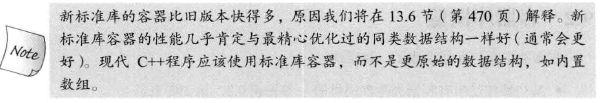
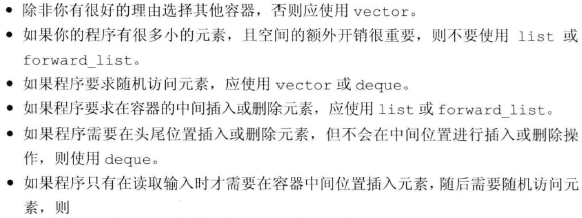
顺序容器：元素位置与加入容器时的顺序相对应

下面是顺序容器的类型以及它们的优缺点:

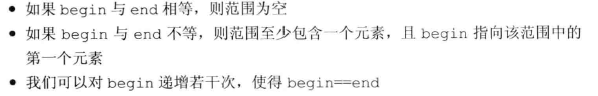




以下是选择容器的时候可以参考的一些建议：



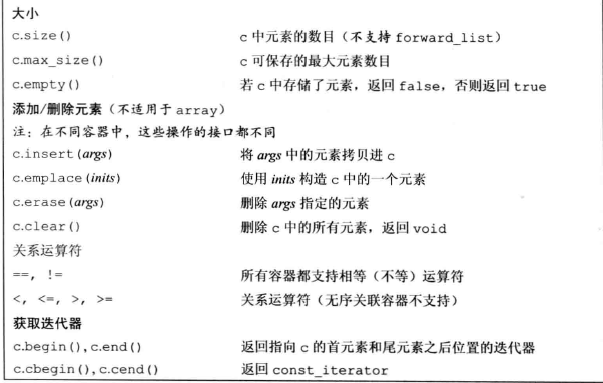
使用左闭合范围使用的编程假设：

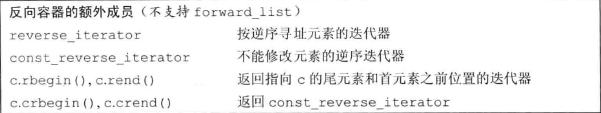


、

下面是一些所有容器都共有的操作：

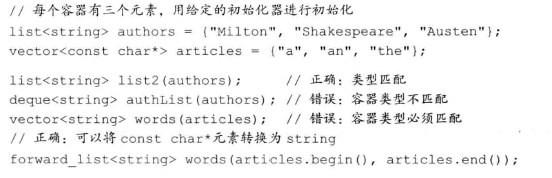


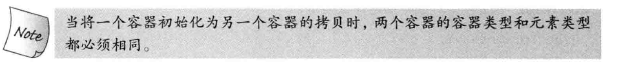




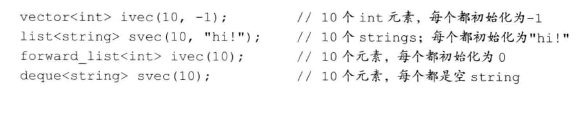
初始化的例子：



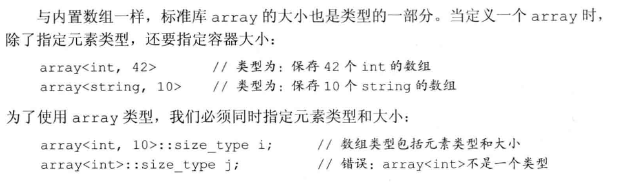




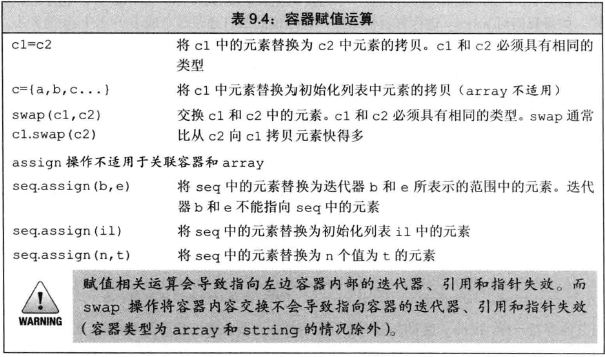




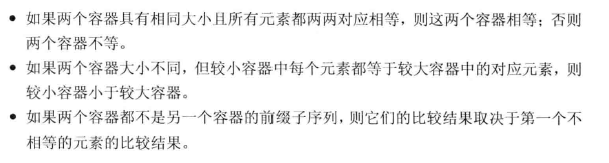




**赋值**



两个容器比较的准则：



**swap函数：**

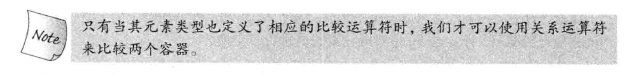
**swap只是交换了两个数据结构，并没有交换元素**

**也就是说交换前后，指针，迭代器等没有变化，他们仍然指向先前的元素，只是原来是vec1[3],变成vec2[3]但是有两个例外**

1. **String**
2. **Array**

**对于string而言，使用swap后，指针和迭代器等会失效，而对于array，会真正的交换元素，所以swap对于array而言和元素数成正比，迭代器和指针仍然不变，但是由于元素值交换，所以对应的元素值改变.**





响应的元素类型必须支持比较操作。