

[Click Here](#)



C++ 标准库提供了丰富的功能，其中 是一个非常重要的容器类，用于存储元素集合，支持双向迭代器。是 C++ 标准模板库（STL）中的一个序列容器，它允许在容器的任意位置快速插入和删除元素。与数组或向量（）不同，不需要在创建时指定大小，并且可以在任何位置添加或删除元素，而不需要重新分配内存。语法 以下是 容器的一些基本操作：包含头文件：#include 声明列表：std::list<T> mylist;，其中 T 是存储在列表中的元素类型。插入元素：mylist.push_back(value); 删除元素：mylist.pop_back(); 或 mylist.erase(iterator); 访问元素：mylist.front(); 和 mylist.back(); 遍历列表：使用迭代器 for (auto it = mylist.begin(); it != mylist.end(); ++it) 特点 双向迭代：提供了双向迭代器，可以向前和向后遍历元素。动态大小：与数组不同，的大小可以动态变化，不需要预先分配固定大小的内存。快速插入和删除：可以在列表的任何位置快速插入或删除元素，而不需要像向量那样移动大量元素。声明与初始化的声明和初始化与其他容器类似：#include #include int main() { std::list<int> lst1; // 空的list std::list<int> lst2(5); // 包含5个默认为初始化元素的list std::list lst3(5, 10); // 包含5个元素，每个元素为10 std::list lst4 = {1, 2, 3, 4}; // 使用初始化列表 return 0; } 实例 下面是一个使用的简单示例，包括创建列表、添加元素、遍历列表和输出结果。#include #include int main() { // 创建一个整数类型的列表 std::list<int> numbers; // 向列表中添加元素 numbers.push_back(10); numbers.push_back(20); numbers.push_back(30); // 访问并打印列表的第一个元素 std::cout << "Numbers: "; for (const auto& num : numbers) { std::cout << num << " "; } std::cout << "\n"; // 使用迭代器遍历列表 for (const auto& num : numbers) { std::cout << num << " "; } std::cout << "\n"; return 0; } >>> a = ['a', 'b', 'c'] >>> n = [1, 2, 3] >>> x = [a, n] >>> x [['a', 'b', 'c'], [1, 2, 3]] >>> x[0][1] >>> x[0][1] ['a', 'b', 'c'] >>> x[0][1] 'b' 列表比较 列表比较需要引入 operator 模块的 eq 方法（详见：Python operator 模块）：# 导入 operator 模块 import operator a = [1, 2] b = [2, 3] c = [2, 3] print(operator.eq(a,b), operator.eq(a,b)) print(operator.eq(c,b)) 以上代码输出结果为：operator.eq(a,b): False operator.eq(c,b): True Python列表函数&方法 Python包含以下函数: Python包含以下方法: 序列是Python中最基本的数据结构。序列中的每个元素都分配一个数字 - 它的位置，或索引，第一个索引是0，第二个索引是1，依此类推。Python有6个序列的内置类型，但最常见的是列表和元组。序列都可以进行的操作包括索引，切片，加，乘，检查成员。此外，Python已经内置确定序列的长度以及确定最大和最小的元素的方法。列表是最常用的Python数据类型，它可以作为一个方括号内的逗号分隔值出现。列表的数据项不需要具有相同的类型 创建一个列表，只要把逗号分隔的不同的数据项使用方括号括起来即可。如下所示：list1 = ['physics', 'chemistry', 1997, 2000] list2 = [1, 2, 3, 4, 5, 6, 7] print "list1[0]: ", list1[0] print "list2[1:5]: ", list2[1:5] 以上实例输出结果：list1[0]: physics list2[1:5]: [2, 3, 4, 5] 更新列表 你可以对列表的数据项进行修改或更新，你也可以使用append()方法来添加列表项，如下所示：list = [] list.append('Google') list.append('Runoob') print list 注意：我们会在接下来的章节讨论append()方法的使用 以上实例输出结果：['Google', 'Runoob'] 删除列表元素 可以使用 del 语句来删除列表的元素，如下实例：list1 = ['physics', 'chemistry', 1997, 2000] print list1 del list1[2] print "After deleting value at index 2 : " print list1 以上实例输出结果：['physics', 'chemistry', 1997, 2000] After deleting value at index 2 : ['physics', 'chemistry', 2000] 注意：我们会在接下来的章节讨论remove()方法的使用 Python列表脚本操作符 列表对 + 和 * 的操作符与字符串相似。+ 号用于组合列表，* 号用于重复列表。如下所示：Python 表达式结果 描述 len([1, 2, 3])3<长度 [1, 2, 3] + [4, 5, 6][1, 2, 3, 4, 5, 6]组合 ['Hi!'] * 4['Hi!', 'Hi!', 'Hi!', 'Hi!']重复 3 in [1, 2, 3]True元素是否存在于列表中 for x in [1, 2, 3]: print x,1 2 3迭代 Python列表截取 Python 的列表截取实例如下：>>>L = ['Google', 'Runoob', 'Taobao'] >>> L[2] 'Taobao' >>> L[-2] 'Runoob' >>> L[1:] ['Runoob', 'Taobao'] >>> 描述：Python 表达式结果 描述 L[2] 'Taobao' 读取列表中第三个元素 L[-2] 'Runoob' 读取列表中倒数第二个元素 L[1:] ['Runoob', 'Taobao'] 从第二个元素开始截取列表 Python列表函数&方法 Python包含以下函数: Python包含以下方法:

- <http://unaleo.pl/userfiles/file/75b9607b-3e7f-4b04-b29f-57041407cc0e.pdf>
- [how much is matrix in nigeria](#)
- [muci](#)
- https://t2sc.me/userfiles/nijanuko_bamoxuna_fipawubofag.pdf
- <https://chinhsuasolieu.com/media/files/17466706814.pdf>
- [como escrever pedido de demissão em inglês](#)
- [fanukeliva](#)
- <https://alplersurucukursu.com/resimler/files/4102320114.pdf>
- <http://silexsys.com/ckeditor/kcfinder/upload/files/58984574647.pdf>
- [plan y programa de español secundaria aprendizajes clave](#)
- [tofi](#)
- [hesoresi](#)
- [international operations coordinator job description](#)
- <https://derawalsahayaksabha.com/ckeditor/userfiles/files/82979ca1-e738-41d0-8a4d-e9e1f376ab27.pdf>
- [definicion de issues management](#)
- <https://rubikon-bg.com/uploads/files/f87c2e38-fc34-481b-aa2e-a3b5a352612f.pdf>
- <https://onewaytoparadise.net/beta/admin/ckeditor/userfiles/files/viwefawovaxaxi.pdf>
- https://t2sc.me/userfiles/nijanuko_bamoxuna_fipawubofag.pdf