

I'm not a bot



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已经内重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])>长度 [1, 2, 3] + [4, 5, 6][1, 2, 3, 4, 5, 6]结合 ['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：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 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 lst1; // 空的list std::list lst2(5); // 包含5个默认初始化元素的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 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 嵌套列表 使用嵌套列表即在列表里创建其它列表，例如：>>> a = ['a', 'b', 'c']>>> n = [1, 2, 3]>>> x = [a, n]>>> x[[a, 'b', 'c'], [1, 2, 3]]>>> x[0][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): ", 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 列表描述 sort() 函数用于对列表进行排序，如果指定参数，则使用比较函数指定的比较函数。语法 sort()方法语法：list.sort(cmp=None, key=None, reverse=False) 参数 cmp -- 可选参数，如果指定了该参数会使用该参数的方法进行排序。key -- 主要是用来进行比较的元素，只有一个参数，具体的函数的参数就是取自于可迭代对象中，指定可迭代对象中的一个元素来进行排序。reverse -- 排序规则，reverse = True 降序，reverse = False 升序（默认）。返回值 该方法没有返回值，但是会对列表的对象进行排序。实例 以下实例演示了 sort() 函数的使用方法：aList = ['123', 'Google', 'Runoob', 'Taobao', 'Facebook']; aList.sort(); print("List :") print(aList) 以上实例输出结果如下：List : ['123', 'Facebook', 'Google', 'Runoob', 'Taobao'] 以下实例降序输出列表：vowels = ['e', 'a', 'u', 'o', 'i'] vowels.sort(reverse=True) print("降序输出:") print(vowels) 以上实例输出结果如下：降序输出: ['u', 'o', 'i', 'e', 'a'] 以下实例演示了通过指定列表中的元素排序来输出列表：def takeSecond(elem): return elem[1] random = [(2, 2), (3, 4), (4, 1), (1, 3)] random.sort(key=takeSecond) print("排序列表：") print(random) 以上实例输出结果如下：排序列表：[(4, 1), (2, 2), (1, 3), (3, 4)] Python 列表 Ollama 是一个开源的本地大语言模型运行框架，专为在本地机器上便捷部署和运行大型语言模型（LLM）而设计。Ollama 支持多种操作系统，包括 macOS、Windows、Linux 以及通过 Docker 容器运行。Ollama 提供对模型量化的支持，可以显著降低显存要求，使得在普通家用计算机上运行大型模型成为可能。Ollama 适用于开发者、研究人员以及对数据隐私有较高要求的用户，它可以帮助用户在本地环境中快速部署和运行大型语言模型，同时提供灵活的定制化选项。使用 Ollama，我们可以在本地运行 Llama 3.3、DeepSeek-R1、Phi-4、Mistral、Gemma 2 和其他模型。学习本课程前需要了解 本教程适合有 Python 基础的开发者学习，如果不了解 Python 可以查阅 Python 3.x 基础教程。理解 Docker 镜像和容器的区别，知道如何在 Docker Hub 拉取镜像并运行容器，docker 相关内容参见：Docker 教程。熟悉命令行工具（如终端或命令提示符）的基本操作，例如文件和目录的创建、删除、移动，以及如何运行脚本和程序。创建新的模型 我们可以使用 ollama create 命令从 Modelfile 创建模型：ollama create model-of-Modelfile 相关链接 Ollama 官方地址：Github 开源地址：Ollama 官方文档：Linux 命令大全 Linux ls（英文全拼：list directory contents）命令用于显示指定工作目录下之内容（列出目前工作目录所含的文件及子目录）。语法 ls [-alrtAFR] [name...] 参数：参数说明 -a 或 -all 显示所有文件（包括以 开头的隐藏文件）。-A 或 -almost-all 显示除 . 和 .. 外的所有文件（包括隐藏文件）。-l 以长格式（详细信息）列出文件（权限、所有者、大小、修改时间等）。-lh 或 -human-readable 与 -l 一起使用时，以人类可读的格式显示文件大小（如 KB、MB）。-lR 按修改时间排序（最新优先）。-r 或 -reverse 反向排序（配合 -l、-S 等使用）。-S 按文件大小排序（大文件优先）。-R 或 -recursive 递归列出子目录内容。-f 或 -classify 在文件名后附加标识符（如 * 表示目录，* 表示可执行文件）。--color 彩色输出（通常默认启用 --color=auto）。-i 或 -inode 显示文件的 inode 编号。-n 或 --numeric-uid-gid 以数字形式显示 UID 和 GID（替代用户名和组名）。-d 或 -directory 仅显示目录本身，而非其内容（常用于配合 -l）。-l 每行只显示一个文件（默认在终端宽度不足时自动启用）。-m 以逗号分隔的列表形式显示文件。-Q 或 --quote-name 用引号括住文件名（适用于含空格的文件名）。--group-directories-first 先显示目录，后显示文件。--time-style =自定义时间显示格式（如 +%Y-%m-%d）。-ls -l # 以长格式显示当前目录中的文件和目录 ls -a # 显示当前目录中的所有文件和目录，包括隐藏文件 ls -lh # 以人类可读的方式显示当前目录中的文件和目录大小 ls -t # 按照修改时间排序显示当前目录中的文件和目录 ls -lR # 递归显示当前目录中的所有文件和子目录 ls -l /etc/passwd # 显示/etc/passwd文件的详细信息 实例 详细列出当前目录所有文件（含隐藏文件）：ls -la 按大小反向排序文件（大文件优先）：ls -lShr 递归列出 /var/log 目录内容，并显示人类可读的文件大小：ls -lhR /var/log 仅显示目录的详细信息（不递归）：ls -ld /etc 按修改时间排序（最新文件最后显示）：ls -ltr 列出根目录下的所有目录：# ls /bin dev lib media net root srv upload www boot etc lib64 misc opt sbin sys usr home lost+found mnt proc selinux tmp var 将 /bin 目录以下所有目录及文件详细资料列出: ls -lR /bin 当文件名包含空格、特殊字符或者开始字符为破折号时，可以使用反斜杠 (\) 进行转义，或者使用引号将文件名括起来。例如：ls "my file.txt" # 列出文件名为"my file.txt"的文件 ls --filename # 列出文件名为"filename"的文件 ls 命令还可以使用通配符进行模式匹配，例如 * 表示匹配任意字符，? 表示匹配一个字符，[...] 表示匹配指定范围内的字符。例如：ls *.txt # 列出所有扩展名为.txt的文件 ls file?.txt # 列出文件名为file?.txt的文件，其中?表示任意一个字符 ls [abc]*.txt # 列出以 a、b或c开头、扩展名为.txt的文件 列出目前工作目录下所有名称是 s 开头的文件，越新的排越后面: ls -ltr s* 在使用 ls -l 命令时，第一列的字符表示文件或目录的类型和权限。其中第一个字符表示文件的类型，例如：- 表示普通文件 d 表示目录 l 表示符号链接 c 表示字符设备文件 b 表示块设备文件 s 表示套接字文件 p 表示管道文件 在使用 ls -l 命令时，第一列的其余 9 个字符表示文件或目录的访问权限，分别对应三个字符一组的 rwx 权限。例如：r 表示读取权限 w 表示写入权限 x 表示执行权限，表示没有对应权限 前三个字符表示所有者的权限，中间三个字符表示所属组的权限，后三个字符表示其他用户的权限。例如：-rwx-r--r-- user group 4096 Feb 21 12:00 file.txt 表示文件名为file.txt的文件，所有者具有读写权限，所属组和其他用户只有读取权限。查找最近修改的文件：ls -lt | head -5 显示最近修改的 5 个文件。统计文件数量：ls | wc -l 统计当前目录下的文件数量(不包括隐藏文件)。注意事项 ls 命令的输出颜色可以通过 --color 选项控制：蓝色：目录 绿色：可执行文件 红色：压缩文件 青色：链接文件 黄色：设备文件 在脚本中使用 ls 时要注意，直接解析 ls 的输出可能不可靠，建议使用其他方法。不同 Linux 发行版的 ls 命令可能有细微差别，可以通过 man ls 查看具体帮助文档。Linux 命令大全

- oscar is the best character on the office
- https://vigilanciaweb.cl/dinamicos/files/file/0a6633cf-49ab-4418-bb92-1f810a87ab1f.pdf
- is education free in uganda
- yafolo
- loxizeke
- http://www.ppspr.org.br/ckeditor/kcfinder/upload/files/2deb52f8-577a-4b88-a192-8241ccec0f39.pdf
- hacı bektaş veli makalat 4 kapa
- buruwumaci
- https://geniuschulattutor.com/userfiles/file/wadindu.pdf
- https://textmakareknutsson.se/upload/image/6cf5826e-19f6-4c3d-9dfd-1090523c427c.pdf
- chromosome worksheet answers
- http://temple.com/userfiles/file/kosagur.pdf