

📄 目录

📁 Python教程

Python简介

安装Python

第一个Python程序

Python基础

函数

高级特性

函数式编程

高阶函数

map/reduce

filter

sorted

返回函数

匿名函数

装饰器

偏函数

模块

面向对象编程

面向对象高级编程

错误、调试和测试

IO编程

进程和线程

正则表达式

常用内建模块

常用第三方模块

virtualenv

图形界面

网络编程

电子邮件

访问数据库

Web开发

异步IO

实战

FAQ

期末总结

关于作者



廖雪峰

北京 朝阳区

+ 加关注

廖雪峰

自己的Python课程

Python商业爬虫全解码

让天下没有爬不到的数据！

Python爬虫 + 数据分析

Python机器学习 + 深度学习

找廖雪峰老师

廖雪峰老师

自己的Java课程

Java高级架构师

更专业 更权威

源码分析专题 + 微服务架构专题

高并发分布式专题 + 性能优化专题

找廖雪峰老师

sorted

阅读: 612039

排序算法

排序也是在程序中经常用到的算法。无论使用冒泡排序还是快速排序，排序的核心是比较两个元素的大小。如果是数字，我们可以直接比较，但如果是字符串或者两个dict呢？直接比较数学上的大小是没有意义的，因此，比较的过程必须通过函数抽象出来。

Python内置的 `sorted()` 函数就可以对list进行排序：

```
>>> sorted([36, 5, -12, 9, -21])
[-21, -12, 5, 9, 36]
```

此外，`sorted()` 函数也是一个高阶函数，它还可以接收一个 `key` 函数来实现自定义的排序，例如按绝对值大小排序：

```
>>> sorted([36, 5, -12, 9, -21], key=abs)
[5, 9, -12, -21, 36]
```

key指定的函数将作用于list的每一个元素上，并根据key函数返回的结果进行排序。对比原始的list和经过 `key=abs` 处理过的list：

```
list = [36, 5, -12, 9, -21]

keys = [36, 5, 12, 9, 21]
```

然后 `sorted()` 函数按照keys进行排序，并按照对应关系返回list相应的元素：

```
keys排序结果 => [5, 9, 12, 21, 36]
最终结果      => [5, 9, -12, -21, 36]
```

我们再看一个字符串排序的例子：

```
>>> sorted(['bob', 'about', 'Zoo', 'Credit'])
['Credit', 'Zoo', 'about', 'bob']
```

默认情况下，对字符串排序，是按照ASCII的大小比较的，由于 `'Z' < 'a'`，结果，大写字母 `Z` 会排在小写字母 `a` 的前面。

现在，我们提出排序应该忽略大小写，按照字母序排序。要实现这个算法，不必对现有代码大加改动，只要我们能用一个key函数把字符串映射为忽略大小写排序即可。忽略大小写来比较两个字符串，实际上就是先把字符串都变成大写（或者都变成小写），再比较。

这样，我们给 `sorted` 传入key函数，即可实现忽略大小写的排序：

```
>>> sorted(['bob', 'about', 'Zoo', 'Credit'], key=str.lower)
['about', 'bob', 'Credit', 'Zoo']
```

要进行反向排序，不必改动key函数，可以传入第三个参数 `reverse=True`：

```
>>> sorted(['bob', 'about', 'Zoo', 'Credit'], key=str.lower, reverse=True)
['Zoo', 'Credit', 'bob', 'about']
```

从上述例子可以看出，高阶函数的抽象能力是非常强大的，而且，核心代码可以保持得非常简洁。

小结

`sorted()` 也是一个高阶函数。用 `sorted()` 排序的关键在于实现一个映射函数。

练习

假设我们用一组tuple表示学生名字和成绩：

```
L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]
```

请用 `sorted()` 对上述列表分别按名字排序：

```
# -*- coding: utf-8 -*-

L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]

def by_name(t):
    pass

L2 = sorted(L, key=by_name)
print(L2)
```

▶ Run

再按成绩从高分到低分排序：

```
# -*- coding: utf-8 -*-

L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]

def by_score(t):
    pass

L2 = sorted(L, key=by_score)
print(L2)
```

▶ Run

参考源码

do\_sorted.py

读后有收获可以请作者喝咖啡，读后有疑问请加群讨论：

还可以分享给朋友：

分享到微博

python免费公开课

编程学习网

授课模式：在线直播+课后视频，从零基础到中高级开发工程师

查看详情

python免费公开课

编程学习网

授课模式：在线直播+课后视频，从零基础到中高级开发工程师

查看详情

廖雪峰

官方 独家

Python商业爬虫全解码

找廖雪峰老师

Python全栈实战课程限时免费领取

腾讯课堂

120天腾讯课堂老师带你从零基础到项目实战，全系统学习爬虫、数据分析、Web开发等Python开发技术

立即申请

ACM金牌得主

全球顶尖名企一线数据科学家倾力指导

人工智能与自然语言/计算机视觉课程培训

Artificial Intelligence For NLP/CV Courses

无offer退全款


Python全栈实战课程限时免费领取

腾讯课堂

120天腾讯课堂老师带你从零基础到项目实战，全系统学习爬虫、数据分析、Web开发等Python开发技术

立即申请

评论



分别按名字排序&分数从高分到低分排序：

夜跑急行列车4番线 created at 7 hours ago, Last updated at 7 hours ago

```
L = [('Bob', 75), ('Adam', 92), ('Bart', 66), ('Lisa', 88)]

def by_name(t):
    n=t[0]
    return n

def by_score(t):
    n=-t[1]
    return n

L1 = sorted(L, key=by_score)
L2 = sorted(L, key=by_name)
print(L1)
print(L2)
```

📄 全部内容

🏠 首页