

INDEX



- 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
 - 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

匿名函数

Reads: 16510514

当我们在传入函数时，有些时候，不需要显式地定义函数，直接传入匿名函数更方便。

在Python中，对匿名函数提供了有限支持。还是以 `map()` 函数为例，计算f(x)=x²时，除了定义一个 `f(x)` 的函数外，还可以直接传入匿名函数：

```
>>> list(map(lambda x: x * x, [1, 2, 3, 4, 5, 6, 7, 8, 9]))
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

通过对比可以看出，匿名函数 `lambda x: x * x` 实际上就是：

```
def f(x):
    return x * x
```

关键字 `lambda` 表示匿名函数，冒号前面的 `x` 表示函数参数。

匿名函数有个限制，就是只能有一个表达式，不用写 `return`，返回值就是该表达式的结果。

用匿名函数有个好处，因为函数没有名字，不必担心函数名冲突。此外，匿名函数也是一个函数对象，也可以把匿名函数赋值给一个变量，再利用变量来调用该函数：

```
>>> f = lambda x: x * x
>>> f
<function <lambda> at 0x101c6ef28>
>>> f(5)
25
```

同样，也可以把匿名函数作为返回值返回，比如：

```
def build(x, y):
    return lambda: x * x + y * y
```

练习

请用匿名函数改造下面的代码：

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

def is_odd(n):
    return n % 2 == 1

L = list(filter(is_odd, range(1, 20)))

print(L)
```

▶ Run

小结

Python对匿名函数的支持有限，只有一些简单的情况下可以使用匿名函数。

读后有收获可以支付宝请作者喝咖啡，读后有疑问请加微信群讨论：



还可以分享给朋友：

分享到微博

◀ Previous Page

Next Page ▶

Comments

Make a comment

Sign in to make a comment

