

Vamei

编程，数学，设计

博客园 首页 订阅 管理

随笔-209 文章-1 评论-3802

Python进阶07 函数对象

作者: Vamei 出处: <http://www.cnblogs.com/vamei> 欢迎转载，也请保留这段声明。谢谢！

秉承着一切皆对象的理念，我们再次回头来看函数(function)。函数也是一个对象，具有属性（可以使用`dir()`查询）。作为对象，它还可以赋值给其它对象名，或者作为参数传递。

lambda函数

在展开之前，我们先提一下lambda函数。可以利用lambda函数的语法，**定义函数**。lambda例子如下：

```
func = lambda x,y: x + y
print func(3,4)
```

lambda生成一个函数对象。该函数参数为`x, y`，返回值为`x+y`。函数对象赋给`func`。`func`的调用与正常函数无异。

以上定义可以写成以下形式：

```
def func(x, y):
    return x + y
```

函数作为参数传递

函数可以作为一个对象，进行参数传递。函数名（比如`func`）即该对象。比如说：

```
def test(f, a, b):  
    print 'test'  
    print f(a, b)  
  
test(func, 3, 5)
```

test函数的第一个参数f就是一个函数对象。将func传递给f，test中的f()就拥有了func()的功能。

我们因此可以提高程序的灵活性。可以使用上面的test函数，带入不同的函数参数。比如：

```
test((lambda x,y: x**2 + y), 6, 9)
```

map() 函数

map()是Python的内置函数。它的第一个参数是一个函数对象。

```
re = map((lambda x: x+3), [1, 3, 5, 6])
```

这里，map()有两个参数，一个是lambda所定义的函数对象，一个是包含有多个元素的表。map()的功能是将函数对象依次作用于表的每一个元素，每次作用的结果储存于返回的表re中。map通过读入的函数(这里是lambda函数)来操作数据(这里“数据”是表中的每一个元素，“操作”是对每个数据加3)。

在Python 3.X中，map()的返回值是一个循环对象。可以利用list()函数，将该循环对象转换成表。

如果作为参数的函数对象有多个参数，可使用下面的方式，向map()传递函数参数的多个参数：

```
re = map((lambda x,y: x+y), [1, 2, 3], [6, 7, 9])
```

map()将每次从两个表中分别取出一个元素，带入lambda所定义的函数。

filter() 函数

filter函数的第一个参数也是一个函数对象。它也是将作为参数的函数对象作用于多个元素。如果函数对象返回的是True，则该次的元素被储存于返回的表中。

filter通过读入的函数来筛选数据。同样，在Python 3.X中，filter返回的不是表，而是循环对象。

filter函数的使用如下例：



```
def func(a):  
    if a > 100:  
        return True  
    else:  
        return False  
  
print filter(func, [10, 56, 101, 500])
```



reduce() 函数

reduce函数的第一个参数也是函数，但有一个要求，就是这个函数自身能接收两个参数。reduce可以累进地将函数作用于各个参数。如下例：

```
print reduce((lambda x, y: x+y), [1, 2, 5, 7, 9])
```

reduce的第一个参数是lambda函数，它接收两个参数x, y，返回x+y。

reduce将表中的前两个元素(1和2)传递给lambda函数，得到3。该返回值(3)将作为lambda函数的第一个参数，而表中的下一个元素(5)作为lambda函数的第二个参数，进行下一次的对lambda函数的调用，得到8。依次调用lambda函数，每次lambda函数的第一个参数是上一次运算结果，而第二个参数为表中的下一个元素，直到表中没有剩余元素。

上面例子，相当于((1+2)+5)+7)+9

根据mmufhy的提醒：`reduce()`函数在3.0里面不能直接用的，它被定义在了`functools`包里面，需要引入包，见评论区。

总结

函数是一个对象

用`lambda`定义函数

`map()`

`filter()`

`reduce()`

标签: [Python](#)

好文要顶

关注我

收藏该文



Vamei

关注 - 26

粉丝 - 4985

荣誉: [推荐博客](#)

[+加关注](#)

24

1

(请您对文章做出评价)

« 上一篇: [Python进阶06 循环对象](#)

» 下一篇: [Python进阶08 异常处理](#)

posted @ 2012-07-10 02:31 Vamei 阅读(32382) 评论(41) 编辑 收藏

评论列表

#1楼 2012-07-15 08:06 micfan

```
1 re = map((lambda x,y: x+y),[1,2,3],[6,7,9])
2 print(re)
3 =====Py3.2 output=====
4 <map object at 0x00C6C190>
```

支持(0) 反对(0)

#2楼 2012-12-16 17:41 leopardsaga

进阶系列的都不错

[支持\(0\)](#) [反对\(0\)](#)

#3楼 2012-12-16 17:43 leopardsaga

reduce的doc文档一直没看懂，不知是自己英语不地道，还是它写的太geek了，谢谢

[支持\(0\)](#) [反对\(0\)](#)

#4楼[楼主] 2012-12-16 18:54 Vamei

@ leopardsaga

我也没有读过

[支持\(0\)](#) [反对\(0\)](#)

#5楼 2013-02-19 18:39 mmufhy

reduce()函数在3.0里面不能直接用的，它被定义在了functools包里面，需要引入包

[支持\(0\)](#) [反对\(0\)](#)

#6楼[楼主] 2013-02-19 21:48 Vamei

@ mmufhy

谢谢你的提醒！

[支持\(0\)](#) [反对\(0\)](#)

#7楼[楼主] 2013-02-19 21:50 Vamei

@ mmufhy

我更新了原文。

[支持\(0\)](#) [反对\(0\)](#)

#8楼 2013-02-19 22:06 mmufhy

@ Vamei

写的很好啊。对于JAVA熟悉的人来说，非常容易上手。谢谢分享

[支持\(2\)](#) [反对\(0\)](#)

#9楼[楼主] 2013-02-19 23:25 Vamei

@ mmufhy

python有不少借鉴JAVA的地方。

[支持\(0\)](#) [反对\(0\)](#)

#10楼 2013-03-27 22:52 名而已

为什么我打印出来是这个样子呢？

```
re = map((lambda x:x+3),[1,3,5,6])
>>> print(re)
<map object at 0x023133D0>
```

支持(0) 反对(0)

#11楼[楼主] 2013-03-27 23:00 Vamei

@ 名而已

在Python 3中，返回的是iterator。

你可以用list(re)将它转成list

或者用for打印

支持(1) 反对(0)

#12楼[楼主] 2013-03-27 23:01 Vamei

@ 名而已

我在原文中有说明。

支持(0) 反对(0)

#13楼 2013-03-27 23:19 名而已

嗯嗯，明白了。

可以：

```
>>> re = map((lambda x:x+3),[1,3,5,6])
>>> list(re)
[4, 6, 8, 9]
```

也可以：

```
>>> for i in map((lambda x:x+3),[1,3,5,6]):
print(i)
```

```
4
6
8
9
```

>>>

支持(0) 反对(0)

#14楼 2013-11-18 21:30 kylinfish

在python里,对匿名函数`lambda x,y: x + y`,最容易犯的一个错误是容易写错,写成`lambda x,y: return x+y`这种形式,不知道为什么,多了一个`return`程序会报错?求解?

支持(0) 反对(0)

#15楼[楼主] 2013-11-20 12:18 Vamei

@ kylinfish

`lambda`是一种数学上的函数表达式,类似于

$y = 2x$

从数学上来讲,这已经表达了所谓的“返回值”的概念,即 y ,因变量。

从编程语言的角度上,你可以把它当作强制规定,记下来就好。或者把冒号理解为“`return`”的意思。

支持(1) 反对(0)

#16楼 2013-11-24 15:58 cczscq

很nice,最近也在学习python,脚本语言强大的优势,同样的功能比C,Java实现起来快捷多啦

支持(0) 反对(0)

#17楼 2013-12-29 14:45 _DN

这篇可不可以称作——Python函数式编程的Lambda演算。

支持(0) 反对(0)

#18楼[楼主] 2013-12-29 22:21 Vamei

@ _DN

嗯,确实是这样一些函数式的内容。我当时起名字是想更加强调它和“一切皆对象”这一理念的联系。

支持(0) 反对(0)

#19楼 2014-02-20 17:25 展望天空

很不错,一下子看到这里,继续~

支持(0) 反对(0)

#20楼[楼主] 2014-02-20 18:56 Vamei

@ 展望天空

哈哈，继续努力！

支持(0) 反对(0)

#21楼 2014-03-20 09:13 温柔流氓

@ Vamei

写的不错哦，就准备用你这教材入门，哈哈

支持(0) 反对(0)

#22楼 2014-04-14 11:51 sammys

为嘛有时候刷新页面空白的，上一篇下一篇有时候也不显示。

不点下一篇，还有什么办法直接到下一篇呢？？

支持(0) 反对(0)

#23楼[楼主] 2014-04-14 16:52 Vamei

@ sammys

这个可能是博客园服务器的问题，我也不清楚。

你可以回到目录页？

支持(0) 反对(0)

#24楼 2014-08-18 22:47 11_xiao_7

@ micfan

我的2.7貌似可以正常得到正确的结果啊，3.2不太清楚还

支持(0) 反对(0)

#25楼 2014-11-18 20:42 smartwolf111

支持一下，已经看了这么多了，收获也不少！

支持(0) 反对(0)

#26楼 2014-11-27 21:41 jeffsoft.h

它这个lambda表达式，如果方法体内有多个语句该怎么写呢？例如：

re=map(lambda x:if x>1: x+2 else: x-2,[3,-2])，不会这个labmda的方法体是只能有一个语句吧？

支持(0) 反对(0)

#27楼 2015-01-05 09:06 Teason1010

@ micfan

我也遇到一样的情况，根据返回结果看，该函数返回的是一个指针。打印时使用 `print(*re)` 即可。

支持(0) 反对(0)

#28楼 2015-03-18 15:41 liuwaner118

博主的教程思路清晰，太受用了

支持(0) 反对(0)

#29楼 2015-04-02 14:29 pythonerJ

感谢楼主，教程对我帮助很大。

对python函数调用了解一些了。。

支持(0) 反对(0)

#30楼 2015-04-13 19:55 笨鸟凌峰

感谢楼主，教程很通俗易懂，正在学习中~

有个小小疑问：我发现 `filter` 函数的函数对象返回值如果是某个值，该次的元素也会被存储到返回的 `list` 中，不过 `help` 文档解释也确实如 `lz` 所讲是指返回值为 `True`。个人觉得准确点说，应该是指返回值存在并且不为 `flase` 吧 -。-

支持(0) 反对(0)

#31楼 2015-08-13 11:13 老鼠子

博主教程的知识点罗列清楚，也有系统性，受用了

支持(0) 反对(0)

#32楼 2015-08-21 11:45 Nefeltari

@ jeffsoft.h

`lambda` 表达式在“:”后只能有一个表达式。也就是说，在 `def` 中，用 `return` 可以返回的也可以放在 `lambda` 后面，不能用 `return` 返回的也不能定义在 `python lambda` 后面。因此，像 `if` 或 `for` 或 `print` 这种语句就不能用于 `lambda` 中，`lambda` 一般只用来定义简单的函数。

支持(0) 反对(0)

#33楼 2015-11-17 21:44 米小马

`map` 函数的问题：

```
1 >>> re = map((lambda x,y: x+y),[1,2,3],[6,7,9])
2 >>> print(list(re))
3 [7, 9, 12]    #第一次打印结果
4 >>> print(list(re))
5 []            <b>#第二次打印结果为空表了???，这是为何</b>
6 >>>
```

filter函数也是类似的。

支持(0) 反对(0)

#34楼 2015-11-22 11:55 iSmile12

@ 米小马

我也刚遇到了和你差不多的情况，

```
1 m = map(lambda x: x + 3, [1, 2, 4, 8, 16])
2 n = list(m)
3 print(n)
4 for i in m:
5     print(i)
6 print(type(m))
7
8
9 [4, 5, 7, 11, 19]
10 <class 'map'>
```

支持(0) 反对(0)

#35楼 2015-11-25 14:21 米小马

@ iSmile12

寻找答案...希望有了解答相互告知

支持(0) 反对(0)

#36楼 2015-11-25 15:01 iSmile12

@ 米小马

个人理解，map在python 3里面已经是一个iterator，只能调用一次吧。。

支持(0) 反对(0)

#37楼 2015-11-27 14:05 米小马

@ iSmile12

嗯，有道理，不过不是太明确...

支持(0) 反对(0)

#38楼 2016-01-11 16:22 遠離塵世の方舟

Python 3.x: from functools import reduce

支持(0) 反对(0)

#39楼 2016-03-03 14:32 fiend_hu

filter()函数是剔除判断结果为False的值吧

支持(0) 反对(0)

#40楼 2016-04-07 17:17 wqh2016

受益匪浅

支持(0) 反对(0)

#41楼 2016-06-22 09:49 M-edea

```
1  def func(x,y):
2      print x**y
3  def test(f,a,b):
4      print 'test'
5      print f(a,b)
6  test (func,3,2)
```

输出的内容:

test

9

None

为什么多了个None啊?

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问](#)网站首页。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云一豆果美食、Faceu等亿级APP都在用



ActiveReports

企业级报表服务平台

单独部署、集成应用、报表制作、数据整合
权限管理、移动办公、二次集成开发

[立即了解](#)



极光推送

消息推送领导品牌全面升级





JIGUANG | 极光

[详情点击](#)


公告

你好，这里是Vamei，一名编程爱好者。我在博客里写了**Python/Linux/网络协议/算法/Java/数据科学**系列文章，从这里开始阅读。非常期待和你的交流。



Vamei ：我科研发出#功能性机器人#佳佳，据说以本校5位校花为原型。双...[图]

4月15日 22:33 | 微博

 weibo.com/vamei

我的微博

下列教程已经做成电子出版物，内容经过修订，也方便离线阅读：
协议森林

欢迎阅读我写的其他书籍：

现代小城的考古学家

天气与历史的相爱相杀

随手拍光影

昵称：Vamei

园龄：4年1个月

荣誉：推荐博客

粉丝：4985

关注：26

[+加关注](#)

[常用链接](#)

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

[我的标签](#)

[Python\(61\)](#)

[Java\(42\)](#)

[大数据\(22\)](#)

[Linux\(17\)](#)

[网络\(16\)](#)

[算法\(15\)](#)

[文青\(14\)](#)

[技普\(9\)](#)

[系列索引\(6\)](#)

[开发工具\(4\)](#)

[更多](#)

[系列文章](#)

[Java快速教程](#)

[Linux的概念与体系](#)

[Python快速教程](#)

[数据科学](#)

[协议森林](#)

[纸上谈兵：算法与数据结构](#)

[积分与排名](#)

[积分 - 659668](#)

[排名 - 122](#)

[最新评论](#)

[1. Re:Java基础11 对象引用](#)

[受教！](#)

--MissLost

[2. Re:Python快速教程](#)

看评论区一片喝彩！看来我得在此扎营了！

--测试小蚂蚁

3. Re:Python进阶06 循环对象

好好地列表解析变成了表推导

--ashic

4. Re:"不给力啊，老湿！": RSA加密与破解

感谢楼主精彩分享

--worldball

5. Re:概率论04 随机变量

你写的这一系列太棒了，刚加入博客园就在你这里学到了，我要转载过去学习一下

--yixius

6. Re:Python基础03 序列

挺好的教程、、、

--王小拽的号

7. Re:Python进阶07 函数对象

```
def func(x,y): print x**ydef test(f,a,b): print 'test' print f(a,b)test (func,3,2)
```

输出的内容:tes.....

--M-edea

8. Re:Python进阶02 文本文件的输入输出

@coderXT换行符: \n...

--行者之印

9. Re:数据科学

博主啊，这里是一枚即将大二的计算机新人，大一学了python，java，还有一些算法，数据结构，图论了，感觉我对数学又一些反感，但是听说离散数学对计算机专业的很重要，不知道怎么去学比较好呢，我想像您写.....

--Acokil

10. Re:为什么要写技术博

楼主是用自己自定义的模板吗？在博客园里找不到这种风格的blog模板？

--行者之印

11. Re:来玩Play框架01 简介

挖煤哥,我补充了一下Windows下的搭建play框架,希望有点帮助,谢谢!

--Sungeek

12. Re:来玩Play框架07 静态文件

```
@helper.form(action = routes.Application.upload, 'enctype ->
```

```
"multipart/form-data") {--action = rout.....
```

--quxiaozha

13. Re:来玩Play框架07 静态文件

该记录将/assets/下的URL，对应到项目的/public文件夹内的文件。比如在项目的/public/images/test.jpg，就可以通过/assets/images/test.jpg这一.....

--quxiaozha

14. Re:来玩Play框架06 用户验证

支持挖煤哥~~~

--quxiaozha

15. Re:“不给力啊，老湿！”：RSA加密与破解

@maanshancss请你仔细阅读了这个文章再来评价。...

--Vamei

推荐排行榜

1. “不给力啊，老湿！”：RSA加密与破解(218)
2. Python快速教程(140)
3. 野蛮生长又五年(91)
4. Java快速教程(88)
5. 协议森林01 邮差与邮局 (网络协议概观)(79)
6. 为什么要写技术博(71)
7. 编程异闻录(54)
8. 博客一年：心理之旅(49)
9. 协议森林08 不放弃 (TCP协议与流通信)(45)
10. Python快速教程 尾声(43)
11. 协议森林(42)
12. Java基础01 从HelloWorld到面向对象(42)
13. Python基础08 面向对象的基本概念(40)
14. 一天能学会的计算机技术(34)
15. 博客第二年，杂谈(33)

Copyright ©2016 Vamei

05370300