

阅读: 118875

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 $f(x)=x^2$ 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
```

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。

感觉本站内容不错，读后有收获？




¥ 我要小额赞助，鼓励作者写出更好的教程

还可以分享给朋友

分享 自由的破晓 , eeeeecoey 等4人分享过



珠峰NODE.JS全栈开发



Angular React Gulp
Meteor Vue 微信开发

技术陪伴成长社区 珠峰培训



麦子学院
www.maiziedu.com

百万级python导师亲身指导

保你120天 变身python大牛

有时候，你需要的只是一句点拨

立即咨询

掘金

一个只有高手分享
的技术社区

立即加入

深度学习在线课程

通向无人驾驶的必经之路

[返回函数](#)

[装饰器](#)

评论



那前一个例子就可以这样写了吧

蓝瞳白鸽 created at 7-15 20:16, Last updated at 4天前

```
def count():  
    fs = []  
    for i in range(1,4):  
        def g(j):  
            return lambda:j*j  
        fs.append (g(i))  
    return fs
```



dqsudfik

Created at 10-13 12:30, Last updated at 10-13 12:30

这么写更简洁

```
def count():
    fs = []
    for i in range(1,4):
        fs.append(lambda i=i:i*i)
    return fs
```



霓為衣兮風為馬

Created at 10-14 11:34, Last updated at 10-14 11:34

```
for i in range(1,4):
    fs.append(lambda i=i:i*i)
```

楼上大赞！！一个lambda i=i全搞定



用户5493334320

Created at 5天前, Last updated at 5天前

两个方法不一样的！！

```
def count():
    fs = []
    for i in range(1, 4):
        def g(j):
            return lambda : j * j
        fs.append(g(i))
    return fs
```

和

```
def count4():
    fs = []
    for i in range(1, 4):
        fs.append(lambda i: i * i)
    return fs
```

执行f1, f2, f3 = count()结果不同，请注意。



walkersmile

Created at 4天前, Last updated at 4天前

为什么一个i=i就可以让函数正常运行了，而lambda i:i*i却不可以呢



View Full Discuss

Reply This Topic



本节练习：把匿名函数作为返回值返回和把返回函数，作区分（请多指教：班门弄斧了

^ ^)

武邵华 created at 9-20 23:29, Last updated at 11-22 13:47

```
#!/usr/bin/python
```

```

#!/usr/bin/python
# -*- coding: utf-8 -*-

# 匿名函数lambda使用，上节中学习了“返回函数” 这节课学习了“匿名函数”
# （1）、如果你定义一个有参数的函数，返回函数是一个无参函数，
# 那么将定义的有参函数赋值给一个变量(赋值后变量指针指向函数，这时变量就是函数的别名)时，
# 需要转递参数，调用函数变量就等于执行函数体
# （2）、如果你定义一个无参数的函数，返回函数是一个有参函数，
# 那么将定义无参函数赋值给一个变量(赋值后变量指针指向函数，这时变量就是函数的别名)时，
# 不需要转递参数，调用函数变量时传递参数就等于执行函数体

# 返回函数
def build_return_func1(x, y):
    def g():
        return x**2 + y**2
    return g
# 返回lambda匿名函数
def build_return_lambda1(x, y):
    # 无参数lambda匿名函数
    return lambda: x ** 2 + y ** 2

# 有函数调用

```

[Read More](#)



[wodemy的微博](#)

Created at 9-27 11:33, Last updated at 9-27 11:33

总结得好，概念一下子清晰了好多。



[各人时差](#)

Created at 10-11 11:51, Last updated at 10-11 11:51

有帮助，谢谢



[彼岸 仙人掌](#)

Created at 11-22 13:47, Last updated at 11-22 13:47

@武邵华 解释的很透彻，受教了

[View Full Discuss](#)

[Reply This Topic](#)



[用lambda简化](#)

[IceWeet](#) created at 10-13 17:00, Last updated at 10-13 17:00

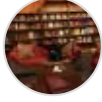
```

def count():
    f = lambda j:lambda :j*j
    fs = []
    for i in range(1, 4):
        fs.append(f(i)) # f(i) 立刻被执行，因此i的当前值被传入f()
    return fs

f1, f2, f3 = count()

```

```
print(f1())
print(f2())
print(f3())
```

[View Full Discuss](#)[Reply This Topic](#)

一个问题？

方道挥斥之 created at 3-18 4:30, Last updated at 9-14 10:36

为什么**lambda**函数作为返回函数的时候不是

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

f=bulid(1,2)

f()

TypeError: <lambda>() missing 2 required positional arguments: 'x' and 'y'
```

反而这个时候 可以这样用了

```
>>> f(2,5)
29
```

求解答！



廖雪峰

Created at 3-18 10:04, Last updated at 3-18 10:04

你的**lambda**函数 `lambda x, y: x * x + y * y`

相当于：

```
def f(x, y):
    return x * x + y * y
```



逝去的9211

Created at 4-2 22:29, Last updated at 4-2 22:29



应该是作用域的问题



Mrtwitter

Created at 4-4 16:27, Last updated at 4-4 16:27

你这样子就可以了，把那个**lambda**的x,y去掉

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

>>> f = build(1, 2)
>>> f()
5
```



奇葩的师弟

Created at 4-6 23:02, Last updated at 4-6 23:02

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

```
>>> f = build(1, 2)
>>> f
<function <lambda> at 0x7fa2c73ed668>
```

当我们调用**build**时，返回的并不是结果，而是**g**函数

```
>>> f(1, 2)
5
```

调用函数**f**时，才真正计算的结果

不知道这样理解对不对,求大神指教.



廖雪峰

Created at 4-7 10:32, Last updated at 4-7 10:32

这样定义：

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

返回的函数在调用时不需要参数，因为 `def g()` 根本就没定义参数：

```
f = build(1, 2)
f()
```

这样定义：

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



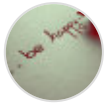
▼ [Read More](#)



[艾菁艾毅](#)

Created at 4-24 15:46, Last updated at 4-24 15:46

是相当于局部变量将全局变量屏蔽了的效果吗？



[北野秋哥](#)

Created at 9-7 14:40, Last updated at 9-7 14:40

老师这么写容易引发误会,**build**函数有两个参数,**a**和**b**,只有传入两个参数调用**build**函数才会返回**lambda**函数,返回的**lambda**函数也需要两个参数来计算最终返回值.前两个参数,和后两个参数是不相干的,都写成**x,y**容易误导初学者!!



[Firefox養殖員](#)

Created at 9-12 16:53, Last updated at 9-12 16:53

楼上的大兄弟说的很好

build函数写成这样，应该就懂了

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

这里的**x,y**没有没实际使用，只是需要两个参数

```
f = build(3, 4)  
f(1, 2)
```

结果是5

```
f = build('a', 'b')  
f(1, 2)
```

结果也是5。

这就是所谓的“闭包”

▼ [Read More](#)



[磊宝仔](#)

Created at 9-14 10:36, Last updated at 9-14 10:36

如果是这样的话

```
def build(x,y):  
    return lambda a,b: aa+bb
```

build函数根本就不需要传参数了，直接

```
def build():  
    return lambda a,b: aa+bb
```

返回一个需要两个参数的匿名函数



 [View Full Discuss](#)

 [Reply This Topic](#)



老师我有个疑惑，求解！

孕Vimmy created at 9-8 15:57, Last updated at 9-9 15:13

```
def f(n):  
    return n * n  
  
lambda n: n * n  
  
def f(n):  
    return lambda : n * n  
  
def f(n):  
    def g():  
        return n * n  
    return g
```

这四个函数都是相等吗？



知名混混

Created at 9-8 19:02, Last updated at 9-8 19:02

不相等啊，你怎么会认为相等呢，， 第一个返回值是int 其余的都是function




孕Vimmy

Created at 9-9 15:13, Last updated at 9-9 15:13

可是上面不是说通过对比可以看出，匿名函数lambda x: x * x实际上就是：

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

 [View Full Discuss](#)

 [Reply This Topic](#)




一个小建议

高小毛君 created at 8-18 11:58, Last updated at 8-18 11:58

这一讲前面用到了匿名函数，初次看的时候还需要翻阅别的资料，如果作者能把这一讲放在前面应该会更好



 [View Full Discuss](#)

 [Reply This Topic](#)



小疑问

演闻先生 created at 2015-10-6 17:13, Last updated at 8-10 18:51

`[lambda x : x % 2 == 0 for x in range(10)]`

的返回结果为什么不是`[True, False, True, False, True, False, True, False, True, False]`



演闻先生

Created at 2015-10-6 17:17, Last updated at 2015-10-6 17:17

自己理解有误，要想得到想要的答案，应该直接写`[x % 2 == 0 for x in range(10)]`；
`lambda x: x % 2 == 0`是一个函数。而列表生成式只需要一个表达式



演闻先生

Created at 2015-10-6 17:20, Last updated at 2015-10-6 17:20

或者这样写：

```
f = lambda x: x % 2 == 0
[f(x) for x in range(10)]
[True, False, True, False, True, False, True, False, True, False]
```



灰手

Created at 4-6 12:43, Last updated at 4-6 12:43

```
list(map(lambda x: x%2==0, range(10)))
```

不是应该这样的么？



滚蛋吧黑眼圈

Created at 5-20 10:53, Last updated at 5-20 10:53

```
f = lambda:[x % 2 == 0 for x in range(10)]
```

返回值`[True, False, True, False, True, False, True, False, True, False]`
这样就可以了，因为冒号后面的相当于返回值，你想要返回的是`list`类型，
`[x % 2 == 0 for x in range(10)]`才是`list`类型



迷宫里的

Created at 6-1 15:18, Last updated at 6-1 15:18



多谢黑眼圈，学到了



海洋why2013

Created at 6-3 1:01, Last updated at 6-3 1:01

```
[lambda x : x % 2 == 0 for x in range(10)]
```

这返回的是函数的list，不是数值



泰山之巅阳

Created at 6-22 10:04, Last updated at 6-22 10:04

那个pingf(x)不也是个函数么，怎么也可以通过“print([pingf(x) for x in range(10)])”生成了想要的数字



作者李四

Created at 6-24 16:56, Last updated at 6-24 16:56

```
list(map(lambda x :x%2==0 ,range(10)))
```

这样的话[True, False, True, False, True, False, True, False, True, False]



顽石47534

Created at 7-26 23:09, Last updated at 7-26 23:09

```
print(list(map(lambda x: x % 2 == 0 ,range(0,10))))
```

或者你可以这样写，前面的演闻先生也说的很明白了，列表生成器只是一个表达式，而map之类的是可以将value带入函数运算的。



是萤火虫

Created at 8-3 13:36, Last updated at 8-3 13:36

```
[lambda x : x % 2 == 0 for x in range(10)]
```


的返回结果为什么不是[True, False, True, False, True, False, True, False, True, False]

我是这么理解的,系统识别问题:

```
x%2==0 for x in range(10)
```

整个被识别成lambda x 的返回值，

因此这个列表生成式生成的是一系列 lambda 函数的列表

 View Full Discuss

 Reply This Topic



求教

想这挺好 created at 6-13 23:09, Last updated at 7-1 10:26

```
def pingf(x):  
    return x*x  
print([pingf(x) for x in range(10)])  
  
print([lambda x :x%2==0 for x in range(10)])
```

上面这两种方式为啥第二个是返回函数第一个却不是呀。



ScriptWang




Created at 7-1 10:26, Last updated at 7-1 10:26

那是因为`lambda x :x%2==0`这句话实际上是指向的一个函数而并非函数调用

```
#你把
[pingf(x) for x in range(10)]
#改成
[pingf for x in range(10)]
#会得到一样的结果，也是因为pingf指向一个函数，而上面pingf(x)是函数调用并且把for迭代出的数字通过x传给了pingf这个函数
```

想要得到一样的结果，只需要将匿名函数进行调用并传入参数：

```
print([ (lambda x :x%2==0) (x) for x in range(10)])
#匿名函数体后面的(x)表示调用该函数并传参，当然这只是一个调用匿名#函数的方式
#或者：
pingf = lambda x :x%2==0
print([ pingf(x) for x in range(10)])
#看懂了吗
```

 View Full Discuss

 Reply This Topic



返回值为函数的问题

习惯孤独al created at 1-18 16:04, Last updated at 4-27 14:42

```
def count():
    fs = []
    for i in range(1, 4):
        def f():
            return i*i
        fs.append(f)
    return fs

f1, f2, f3 = count()
求教一下f1 f2 f3的实现过程，
...
```



习惯孤独al

Created at 1-18 16:06, Last updated at 1-18 16:06

`count()` 返回不是一个列表`[1,4,9]`吗？具体实现过程不是很明白，新手求指教



HowyoungZhou

Created at 1-31 23:43, Last updated at 1-31 23:43



```
fs.append(f)
```

注意这里是将函数`f`添加到了`fs`中，而不是`f`返回的值，可以尝试执行命令：

```
>>> f
<function f at 0x00000283BF7706A8>
>>> count()
[<function count.<locals>.f at 0x00000283BF770840>, <function count.<locals>.f at 0x00000283BF7707B8>, <function count.<locals>.f at 0x00000283BF770598>]
```

可见返回的是一个由三个函数组成的list。
因此当执行代码

```
f1, f2, f3 = count()
```

时，将count包含的三个函数按索引顺序分别赋给f1, f2, f3,而此时i=3，所以函数计算结果全部为9.

```
>>> count()
```

▼ [Read More](#)



[艾菁艾毅](#)

Created at 4-24 15:51, Last updated at 4-24 15:51

在python中一切皆对象，而变量i只有一个，在三次循环中，它指向的对象变化了而已，而函数f1,f2,f3在实际执行的时候i指向的那个地址的实际存放数据已经变为3了，所以三次结果都是一样的。



[任性Monster](#)

Created at 4-27 14:42, Last updated at 4-27 14:42

fs.append(f)这里的f只是指针 不是函数 所以不执行 等到什么时候用什么时候执行 所以在for循环3次之后i变成了3 f总是在所有的执行完了才会去调用 因为它被放在【】中

≡ View Full Discuss

↩ Reply This Topic



[我觉得课程设计不合理](#)

[翁岚敏](#) created at 2015-11-30 21:16, Last updated at 4-12 16:22

既然之前的课程中一直要用到lambda函数 而lambda函数的这节课有没有难点为什么不把它提到前面取

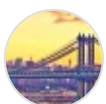


[廖雪峰](#)

Created at 2015-12-1 6:18, Last updated at 2015-12-1 6:18



前面哪用了lambda?



[刚毅木讷823](#)

Created at 2015-12-3 15:44, Last updated at 2015-12-3 15:44

map/reduce用到了



[i11841](#)

Created at 4-12 16:22, Last updated at 4-12 16:22

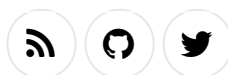
不是不合理，讲一个东西不可避免的用到后面的东西，难道要全都放在第一课吗，这是一个取舍的问题。

 [View Full Discuss](#)

 [Reply This Topic](#)

发表评论

[Sign In to Make a Comment](#)



友情链接: [中华诗词](#) - [阿里云](#) - [SICP](#) - [4closure](#)