7.6 定义匿名或内联函数¶

问题¶

你想为 `sort()` 操作创建一个很短的回调函数，但又不想用 `def` 去写一个单行函数， 而是希望通过某个快捷方式以内联方式来创建这个函数。

解决方案¶

当一些函数很简单，仅仅只是计算一个表达式的值的时候，就可以使用lambda表达式来代替了。比如：

```
>>> add = lambda x, y: x + y
>>> add(2,3)
5
>>> add('hello', 'world')
'helloworld'
>>>
```

这里使用的lambda表达式跟下面的效果是一样的：

```
>>> def add(x, y):
...     return x + y
...
>>> add(2,3)
5
>>>
```

lambda表达式典型的使用场景是排序或数据reduce等：

```
>>> names = ['David Beazley', 'Brian Jones',
...          'Raymond Hettinger', 'Ned Batchelder']
>>> sorted(names, key=lambda name: name.split()[-1].lower())
['Ned Batchelder', 'David Beazley', 'Raymond Hettinger', 'Brian Jones']
>>>
```

讨论¶

尽管lambda表达式允许你定义简单函数，但是它的使用是有限制的。 你只能指定单个表达式，它的值就是最后的返回值。也就是说不能包含其他的语言特性了， 包括多个语句、条件表达式、迭代以及异常处理等等。

你可以不使用lambda表达式就能编写大部分python代码。 但是，当有人编写大量计算表达式值的短小函数或者需要用户提供回调函数的程序的时候， 你就会看到lambda表达式的身影了。