

Python程序设计

第六讲 函数与模块化 lambda表达式



张 华
WHU

lambda表达式

lambda表达式

lambda arg1[,arg2,...]:<expression>

- 用来声明匿名函数，也就是没有名字的临时使用的小函数，尤其适合需要一个函数作为另一个函数参数的场合。
- lambda**表达式只可以包含一个表达式，该表达式的计算结果可以看作是函数的返回值，不允许包含复合语句，但在表达式中可以调用其他函数。

```
>>> sf = lambda x,y : x+y
>>> type(sf)
<class 'function'>
>>> sf(12, 34)
46
```

```
>>> for v in filter(lambda x:x>0, [1,-2,3,-4,5]):
...     print(v, end=' ')
...
1 3 5
```

lambda表达式举例

更多lambda表达式及其应用示例

```
>>> L = [(lambda x: x**2),  
          (lambda x: x**3),  
          (lambda x: x**4)]  
>>> print(L[0](2), L[1](2), L[2](2))  
4 8 16  
  
>>> D = {'f1': (lambda: 2+3),  
          'f2': (lambda: 2*3),  
          'f3': (lambda: 2**3)}  
>>> print(D['f1'](), D['f2'](), D['f3']())  
5 6 8  
  
>>> L = [1, 2, 3, 4, 5]  
>>> print(list(map(lambda x: x+10, L)))  
[11, 12, 13, 14, 15]  
  
>>> L  
[1, 2, 3, 4, 5]
```

#模拟向量运算

lambda表达式举例

更多lambda表达式及其应用示例

```
>>> def demo(n):  
    return n*n  
  
>>> demo(5)  
25  
  
>>> a_list = [1,2,3,4,5]  
>>> list(map(lambda x: demo(x), a_list))    #在lambda表达式中调用函数  
[1, 4, 9, 16, 25]
```

lambda表达式举例

更多lambda表达式及其应用示例

```
>>> data = list(range(20))          #创建列表
>>> data
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

>>> import random
>>> random.shuffle(data)             #打乱顺序
>>> data
[4, 3, 11, 13, 12, 15, 9, 2, 10, 6, 19, 18, 14, 8, 0, 7, 5, 17, 1, 16]

>>> data.sort(key=lambda x: x)       #和不指定规则效果一样
>>> data
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

>>> data.sort(key=lambda x: len(str(x))) #按转换成字符串以后的长度排序
>>> data
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]

>>> data.sort(key=lambda x: len(str(x)), reverse=True) #降序排序
>>> data
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

lambda表达式举例

更多lambda表达式及其应用示例

```
>>> import random
>>> x = [[random.randint(1,10) for j in range(5)] for i in
range(5)]
```

#使用列表推导式创建列表
#包含5个子列表的列表
#每个子列表中包含5个1到10之间的随机数

```
>>> for item in x:
    print(item)
```

```
[5, 6, 8, 7, 4]
[1, 5, 3, 9, 4]
[9, 6, 10, 7, 6]
[8, 2, 7, 1, 6]
[1, 7, 5, 3, 5]
```

lambda表达式举例

更多lambda表达式及其应用示例

```
>>> y = sorted(x, key=lambda item: (item[1], item[4]))  
                                     #按子列表中第2个元素升序、第5个元素升序排序  
  
>>> for item in y:  
    print(item)  
  
[8, 2, 7, 1, 6]  
[1, 5, 3, 9, 4]  
[5, 6, 8, 7, 4]  
[9, 6, 10, 7, 6]  
[1, 7, 5, 3, 5]
```