**map 和 lambda 函数**

1. **map(function, iterable, ...)**

Python 内置的函数，它接收一个函数 f 和一个 list，把函数 f 依次作用在 list 的每个元素上，得到一个新的 list 并返回。

* 计算列表各个元素的平方

def square(x):

return x \* x

print(list(map(square, [1,2,3,4,5])))

[1, 4, 9, 16, 25]

1. **lambda 函数**

操作(参数)，也称为匿名函数。如果你不想在程序中对一个函数使用两次，可以用 lambda 表达式，作用和普通函数一样。

* 用一行就能解决问题的函数
  + add = lambda x, y: x + y

print(add(5, 6))

11

* + numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]

print(list(map(lambda x: x\*x, numbers)))

[1, 4, 9, 16, 25, 36, 49, 64, 81]

* key = lambda 元素: 元素[字段索引]

a = [(2, 56), (3, 12), (6, 10), (9, 13)]

a.sort(key=lambda x: x[1]) # 按照 x[1]（第二维）进行列表排序。

print(a)

[(6, 10), (3, 12), (9, 13), (2, 56)]