

1. 在python2中map（）函数返回的是一个列表，但在python3中他返回的是一个迭代数（iteration）

2. map()的原型是map(function, iterable, ...), 它的返回结果是一个列表。

参数 function: 传的是一个函数名，可以是python内置的，也可以是自定义的。
就像上面的匿名函数lambda

参数 iterable: 传的是一个可以迭代的对象，例如列表，元组，字符串这样的。

*这个函数的意思就是将function应用于iterable的每一个元素，结果以列表的形式返回。

注意到没有，iterable后面还有省略号，意思就是可以传很多个iterable。

如果有额外的iterable参数，并行的从这些参数中取元素，并调用function;

如果一个iterable参数比另外的iterable参数要短，将以None扩展该参数元素。

*详例见：map-test

```
>>>def square(x):        # 计算平方数
...    return x ** 2
```

```
>>> map(square, [1,2,3,4,5]) # 计算列表各个元素的平方
[1, 4, 9, 16, 25]
```

```
>>> map(lambda x: x ** 2, [1, 2, 3, 4, 5]) # 使用 lambda 匿名函数
[1, 4, 9, 16, 25]
```

提供了两个列表，对相同位置的列表数据进行相加

```
>>> map(lambda x, y: x + y, [1, 3, 5, 7, 9], [2, 4, 6, 8, 10])
[3, 7, 11, 15, 19]
```