

1 map()函数的简介以及语法：

map是python **内置函数**，会根据提供的函数对指定的序列做映射。

map()函数的格式是：

```
1 | map(function,iterable,...)
```

第一个参数接受一个函数名，后面的参数接受一个或多个可迭代的序列，返回的是一个 **集合**。

把函数依次作用在list中的每一个元素上，得到一个新的list并返回。注意，map不改变原list，而是返回一个新list。

2 map()函数实例：

```
1 | del square(x):  
2 |     return x ** 2  
3 |  
4 | map(square, [1,2,3,4,5])  
5 |  
6 | # 结果如下:  
7 | [1,4,9,16,25]
```

通过使用lambda匿名函数的方法使用map()函数：

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

通过lambda函数使返回值是一个元组：

```
1 | map(lambda x, y : (x**y,x+y), [2,4,6], [3,2,1])  
2 |  
3 | # 结果如下  
4 | [(8,5),(16,6),(6,7)]
```

当不传入function时，map()就等同于zip()，将多个列表相同位置的元素归并到一个元组：

```
1 | map(None, [2,4,6], [3,2,1])  
2 |
```

```
3 | # 结果如下
4 | [(2,3),(4,2),(6,1)]
```

通过map还可以实现类型转换

将元组转换为list :

```
1 | map(int,(1,2,3))
2 |
3 | # 结果如下:
4 | [1,2,3]
```

将字符串转换为list :

```
1 | map(int,'1234')
2 |
3 | # 结果如下:
4 | [1,2,3,4]
```

提取字典中的key , 并将结果放在一个list中 :

```
1 | map(int,{1:2,2:3,3:4})
2 |
3 | # 结果如下
4 | [1,2,3]
```

文章知识点与官方知识档案匹配，可进一步学习相关知识
