

## 4.11 同时迭代多个序列¶

### 问题¶

你想同时迭代多个序列，每次分别从一個序列中取一个元素。

### 解决方案¶

为了同时迭代多个序列，使用 `zip()` 函数。比如：

```
>>> xpts = [1, 5, 4, 2, 10, 7]
>>> ypts = [101, 78, 37, 15, 62, 99]
>>> for x, y in zip(xpts, ypts):
...     print(x,y)
...
1 101
5 78
4 37
2 15
10 62
7 99
>>>
```

`zip(a, b)` 会生成一个可返回元组  $(x, y)$  的迭代器，其中  $x$  来自  $a$ ， $y$  来自  $b$ 。一旦其中某个序列到底结尾，迭代宣告结束。因此迭代长度跟参数中最短序列长度一致。

```
>>> a = [1, 2, 3]
>>> b = ['w', 'x', 'y', 'z']
>>> for i in zip(a,b):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
>>>
```

如果这个不是你想要的效果，那么还可以使用 `itertools.zip_longest()` 函数来代替。比如：

```
>>> from itertools import zip_longest
>>> for i in zip_longest(a,b):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
(None, 'z')

>>> for i in zip_longest(a, b, fillvalue=0):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
(0, 'z')
>>>
```

### 讨论¶

当你想成对处理数据的时候 `zip()` 函数是很有用的。比如，假设你头列表和一个值列表，就像下面这样：

```
headers = ['name', 'shares', 'price']
values = ['ACME', 100, 490.1]
```

使用 `zip()` 可以让你将它们打包并生成一个字典：

```
s = dict(zip(headers,values))
```

或者你也可以像下面这样产生输出：

```
for name, val in zip(headers, values):  
    print(name, '=', val)
```

虽然不常见，但是 `zip()` 可以接受多于两个的序列的参数。这时候所生成的结果元组中元素个数跟输入序列个数一样。比如；

```
>>> a = [1, 2, 3]  
>>> b = [10, 11, 12]  
>>> c = ['x','y','z']  
>>> for i in zip(a, b, c):  
...     print(i)  
...  
(1, 10, 'x')  
(2, 11, 'y')  
(3, 12, 'z')  
>>>
```

最后强调一点就是， `zip()` 会创建一个迭代器来作为结果返回。如果你需要将结对的值存储在列表中，要使用 `list()` 函数。比如：

```
>>> zip(a, b)  
<zip object at 0x1007001b8>  
>>> list(zip(a, b))  
[(1, 10), (2, 11), (3, 12)]  
>>>
```