

## 5.8 固定大小记录的文件迭代¶

### 问题¶

你想在一个固定长度记录或者数据块的集合上迭代，而不是在一个文件中一行一行的迭代。

### 解决方案¶

通过下面这个小技巧使用 `iter` 和 `functools.partial()` 函数：

```
from functools import partial

RECORD_SIZE = 32

with open('somefile.data', 'rb') as f:
    records = iter(partial(f.read, RECORD_SIZE), b'')
    for r in records:
        ...
```

这个例子中的 `records` 对象是一个可迭代对象，它会不断的产生固定大小的数据块，直到文件末尾。要注意的是如果总记录大小不是块大小的整数倍的话，最后一个返回元素的字节数会比期望值少。

### 讨论¶

`iter()` 函数有一个鲜为人知的特性就是，如果你给它传递一个可调对象和一个标记值，它会创建一个迭代器。这个迭代器会一直调用传入的可调对象直到它返回标记值为止，这时候迭代终止。

在例子中，`functools.partial` 用来创建一个每次被调用时从文件中读取固定数目字节的可调对象。标记值 `b''` 就是当到达文件结尾时的返回值。

最后再提一点，上面的例子中的文件时以二进制模式打开的。如果是读取固定大小的记录，这通常是最普遍的情况。而对于文本文件，一行一行的读取(默认的迭代行为)更普遍点。