

# python yield

作者: [jaz](#)

原文链接: <https://ld246.com/article/1456458213476>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

python中通过列表生成式，我们可以直接创建一个列表。但是，受到内存限制，列表容量肯定有限的。而且，创建一个包含100万个元素的列表，不仅占用很大的存储空间，如果我们仅仅需要访问前面几个元素，那后面绝大多数元素占用的空间都白白浪费了。

所以，如果列表元素可以按照某种算法推算出来，那我们是否可以在循环的过程中不断推算出后面的元素呢？这样就不必创建完整的list，从而节省大量的空间。在Python中，这种一边循环一边计算的机制，称为生成器（Generator）。

有yield的函数成为生成器。变成生成器的函数执行顺序：

函数是顺序执行，遇到return语句或者最后一行函数语句就返回。而变成generator的函数，在每次调用next()的时候执行，遇到yield语句返回，再次执行时从上次返回的yield语句处继续执行。

```
yield &nbsp;a &nbsp;就是返回a；等待下一次next()。
```

```
def fib(max):
```

```
    a, b = 0, 1
```

```
    while a &lt; max:
```

```
        yield a
```

```
        a, b = b, a + b
```

```
<br />
```