PYTHON
编程基础

生成器

生成器

01

当一个列表中包含大量元素时，如果一次性生成这些元素并保存在列表中，将占用大量的内存空间（有的情况下可用内存甚至无法满足存储需求）。

02

对于这个问题，我们可以通过生成器（generator）来解决，即根据需要进行计算并获取列表中某个元素的值。

03

将列表生成表达式中的一对中括号改为一对小括号即可得到生成器，对于生成器对象，也可以像其他可迭代对象一样使用for循环遍历对象中的每一个元素。

例：

```
g=(x*x for x in range(10)) #创建一个生成器对象并赋给g
print('g的类型为：',type(g))
for i in g:
    print(i, end=' ')
```


```
g的类型为：  <class 'generator'>
0 1 4 9 16 25 36 49 64 81
```

生成器

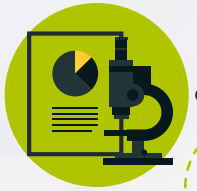
01

如果生成元素的方法比较复杂，不适合用for循环方式实现，我们还可以借助yield关键字利用函数实现生成器的功能。

02

例：实现faclist函数，依次生成1的阶乘、2的阶乘、…、n的阶乘。

例：



```
def faclist(n): #定义函数faclist
    result=1
    for i in range(2,n+2): #i在2至n+1范围内依次取值
        yield result #遇到yield即暂停执行并返回result，下次执行时继续
                     #从此处开始执行
        result*=i #将i乘到result上
for i in faclist(10): #遍历faclist并输出每个元素的值
    print(i, end=' ')
```

1 2 6 24 120 720 5040 40320 362880 3628800