**一、Generator的含义**

意义：提供异步编程解决方案

原理：半协程调度(Generator执行权需调用者返还)

Generator函数遇yield出栈并不销毁，且保存程序计数器状态；遇next时再压入栈

**二、基本用法**

**1 Generator用法(协程)**

协程定义

def g(): # 定义一个Generator函数()

// some code

a = yield 'hello' # 返回表达式的值，并释放执行权

return 'ending' # 结束调度，抛出StopIteration: ending异常

协程调度

g() 构建一个协程实例

.send(val) 使上一yield返回；并执行下一yield表达式

val参数 传给上一yield执行返回值(即a的)，首次需None

send返回值 获取下一yield表达式的值

.throw(Exception, message) 使协程上一yield抛出异常；并执行到下一yield

.close() 使协程上一yield终止(无返回值)