1. 数据库：
   1. MySQL（pymysql模块）：
      1. db = pymysql.connect(“主机名/IP”, ”用户名”, ”密码”, ”数据库名称”)
      2. db.commit() 提交数据
      3. db.rollback() 回滚数据
      4. db.close()
   2. cursor（游标对象）:
      1. cursor = db.cursor([flag]) 创建游标对象，默认为普通游标
         1. flag：
            1. pymysql.cursors.DictCursor 字典游标
            2. pymysql.cursors.SS[Dict]Cursor 流式[字典]游标（类似于迭代器）

流式游标一般使用fetchone得到结果后用while输出。

使用流式游标时，如果数据量很大，导致游标一直处在循环遍历状态，这时数据库连接（db）是被占用的，不能再被用于执行其他sql，如果要执行其他sql那就必须再创建一个数据库连接，游标占用数据库连接的时长是有限制的，如果流式游标一直在遍历，60秒后数据库连接会断开，不过可以在创建数据库连接时传入参数init\_command=("SET NET\_WRITE\_TIMEOUT=XX")来设置这个超时时间。

* + 1. cursor.execute(“SQL语句”) 执行SQL语句
    2. data = cursor.fetchone() 获取单条数据，返回元组
       1. fetchall() fetchmany(n) 获取n条数据，返回大元组包小元组
    3. cursor.executemany(“SQL语句”，参数序列) 插入多条数据
  1. SQL语句：
     1. DROP TABLE IF EXISTS 表名; 如果存在表，删除
     2. CREATE TABLE 表名 (
        1. id int(8) NOT NULL AUTO\_INCREMENT PRIMARY KEY,
        2. 字段名 类型(长度，小数点位数) 约束 [自动编号],
     3. )ENGINE=MyISAM AUTO\_INCREMENT=1 DEFAULT CHARSER=utf8;
        1. ‘’’engine指定管理方式，auto指定自动编号起始数，默认编码为utf-8
        2. NOT NULL：该字段永不为空 DEFAULT NULL：该字段默认为空
        3. DEFAULT 0：该字段默认为0‘’’
     4. insert into books(字段1，字段2…) values (‘%s’,%s…); 字符型需加‘ ’
        1. 用%s作占位符，防止SQL注入。
     5. select 返回列 from 表名 [where 条件] [order by 列名 排序方式];
        1. 返回列：返回数据所包含的列，用逗号分隔，无需加引号，\*表示返回所有，列名后可使用as关键词
           1. 可对列名进行四则运算，会创建新列，并对原数据运算后放入其中。
           2. 使用连接符 || 和as可合并结果：

select name || price as result from bill ;

* + - 1. 条件：返回满足条件的数据。
         1. 可使用所有条件运算符(使用=而非==), AND, BETWEEN, OR, IN, NOT LIKE(等价于后者), !=

price BETWEEN 50 and 80 price IN (50,51,52)

* + - * 1. LIKE：格式搜索 ’\_’占一位，’%’占多位，escape’#’定义转义字符为#

name LIKE ‘\_嘉\_是%’ ;

* + - 1. 排序方式：ASC：从小到大（默认） DESC：从大到小