
咨询电话: 010-61943026

远程课程咨询: 327712287

Pythoner.cn 技术交流一群: 321318523

Pythoner.cn 技术交流二群: 194102256

Pythoner.cn

***** 永久存储模块 *****

pickle 和 marshal 模块

#Python 提供了许多可以实现最小化永久性储存的模块. 其中的一组(marshal 和 pickle)可
#以用来转换并储存 Python 对象. 该过程将比基本类型复杂的对象转换为一个二进制数据
集合,

#这样就可以把数据集合保存起来或通过网络发送, 然后再重新把数据集合恢复原来的对象
格式.

#这个过程也被称为数据的扁平化, 数据的序列化, 或者数据的顺序化. 另外一些模块

#(dbhash/bsddb, dbm, gdbm, dumbdbm 等)以及它们的"管理器"(anydbm)只提供了 Python
字

#字符串的永久性储存. 而最后一个模块(shelve) 则两种功能都具备.

#marshal 和 pickle 模块都可以对 Python 对象进行储存转换. 这些模块本身

#并没有提供"永久性储存"的功能, 因为它们没有为对象提供名称空间, 也没有提供对永久
性储存对

#象的并发写入访问(concurrent write access). 它们只能储存转换 Python 对象, 为保存和传
输

#提供方便. 数据储存是有次序的(对象的储存和传输是一个接一个进行的). marshal 和
pickle 模

#块的区别在于 marshal 只能处理简单的 Python 对象(数字, 序列, 映射, 以及代码对象),
而

#pickle 还可以处理递归对象, 被不同地方多次引用的对象, 以及用户定义的类和实例.
pickle 模

#块还有一个增强的版本叫 cPickle, 使用 C 实现了相关的功能.

DBM 风格的模块

#*db* 系列的模块使用传统的 DBM 格式写入数据, Python 提供了 DBM 的多种实现:

#dbhash/bsddb, dbm, gdbm, 以及 dumbdbm 等. 你可以随便按照你的爱好使用, 如果你不
确定

#的话, 那么最好使用 anydbm 模块, 它会自动检测系统上已安装的 DBM 兼容模块, 并选
择"最好"

#的一个. dumbdbm 模块是功能最少的一个, 在没有其它模块可用时, anydbm 才会选择它.
这些模块

#为用户的对象提供了一个命名空间, 这些对象同时具备字典对象和文件对象的特点. 不过
不足之处

#在于它们只能储存字符串, 不能对 Python 对象进行序列化.

shelve 模块

#shelve 模块使用 anydbm 模块寻找

#合适的 DBM 模块, 然后使用 cPickle 来完成对储存转换过程. shelve 模块允许对数据库文件进行

#并发的读访问, 但不允许共享读/写访问. 这也许是我们 Python 标准库里找到的最接近于永久

#性储存的东西了. 可能有一些第三方模块实现了"真正"的永久性储存.

#核心模块: pickle 和 cPickle

#你可以使用 pickle 模块把 Python 对象直接保存到文件里, 而不需要把它们转化为字符串, #也不用底层的文件访问操作把它们写入到一个二进制文件里. pickle 模块会创建一个 Python 语言

#专用的二进制格式, 你不需要考虑任何文件细节, 它会帮你干净利索地完成读写对象操作, 唯一需

#要的只是一个合法的文件句柄.

#pickle 模块中的两个主要函数是 dump() 和 load(). dump() 函数接受一个文件句柄和一个 #数据对象作为参数, 把数据对象以特定格式保存到给定文件里. 当我们使用 load() 函数从文件中

#取出已保存的对象时, pickle 知道如何恢复这些对象到它们本来的格式. 我们建议你看一看 #pickle 和更"聪明"的 shelve 模块, 后者提供了字典式的文件对象访问功能, 进一步减少了程序

#员的工作.

#cPickle 是 pickle 的一个更快的 C 语言编译版本.

咨询电话: 010-61943026

远程课程咨询: 327712287

Pythoner.cn 技术交流一群: 321318523

Pythoner.cn 技术交流二群: 194102256

Pythoner.cn