

数据持久化

本章中描述的模块支持在磁盘上以持久形式存储 Python 数据。[pickle](#) 和 [marshal](#) 模块可以将许多 Python 数据类型转换为字节流，然后从字节中重新创建对象。各种与 DBM 相关的模块支持一系列基于散列的文件格式，这些格式存储字符串到其他字符串的映射。

本章中描述的模块列表是：

- [pickle](#) --- Python 对象序列化
 - 与其他 Python 模块间的关系
 - 与 [marshal](#) 间的关系
 - 与 [json](#) 模块的比较
 - 数据流格式
 - 模块接口
 - 可以被封存/解封的对象
 - 封存类实例
 - 持久化外部对象
 - Dispatch 表
 - 处理有状态的对象
 - 类型，函数和其他对象的自定义归约
 - 外部缓冲区
 - 提供方 API
 - 使用方 API
 - 示例
 - 限制全局变量
 - 性能
 - 例子
 - 命令行接口
- [copyreg](#) --- 注册 [pickle](#) 支持函数
 - 示例
- [shelve](#) --- Python 对象持久化
 - 限制
 - 示例
- [marshal](#) --- 内部 Python 对象序列化
- [dbm](#) --- Unix "数据库" 接口
 - [dbm.sqlite3](#) --- 针对 [dbm](#) 的 SQLite 后端
 - [dbm.gnu](#) --- GNU 数据库管理器
 - [dbm.ndbm](#) --- 新数据库管理器
 - [dbm.dumb](#) --- 便携式 DBM 实现
- [sqlite3](#) --- SQLite 数据库的 DB-API 2.0 接口
 - 教程

- 参考
 - 模块函数
 - 模块常量
 - 连接对象
 - 游标对象
 - Row 对象
 - Blob 对象
 - PrepareProtocol 对象
 - 异常
 - SQLite 与 Python 类型
 - 默认适配器和转换器（已弃用）
 - 命令行接口
- 常用方案指引
 - 如何在 SQL 查询中使用占位符来绑定值
 - 如何将自定义 Python 类型适配到 SQLite 值
 - 如何编写可适配对象
 - 如何注册适配器可调用对象
 - 如何将 SQLite 值转换为自定义 Python 类型
 - 适配器和转换器范例程序
 - 如何使用连接快捷方法
 - 如何使用连接上下文管理器
 - 如何使用 SQLite URI
 - 如何创建并使用行工厂对象
 - 如何处理非 UTF-8 文本编码格式
- 说明
 - 事务控制
 - 通过 `autocommit` 属性进行事务控制
 - 通过 `isolation_level` 属性进行事务控制