数据库存储

- sqlite
  - 创建数据库

```
import sqlite3
import os
dbpath = 'data.sqlite'
if not os.path.exists(dbpath):
    conn = sqlite3.connect(dbpath)
    c = conn.cursor()
    c.execute('''
    CREATE TABLE persons
    (id INT PRIMARY KEY    NOT NULL,
    name            TEXT   NOT NULL,
    age             INT    NOT NULL,
    address         CHAR(50)    NOT NULL,
    salary          REAL);
    ''')
    conn.commit()
    conn.close()
    print('创建数据库成功')
```

  - 导入sqlite3模块
  - 创建路径和sqlite文件，如果没有则创建
  - conn = sqlite3.connect(dbpath)：连接数据库
  - c = conn.cursor()：初始化游标对象
  - c.execute()：执行sql语句创建数据表和字段
  - conn.commit()：提交数据
  - conn.close()：关闭数据库
  - 数据库查询操作

```
conn = sqlite3.connect(dbpath)
c = conn.cursor()
c.execute('delete from persons')
c.execute("INSERT INTO persons (id,name,age,address,salary)\
      VALUES (1,'Paul',32,'California',20000.00)");
c.execute("INSERT INTO persons (id,name,age,address,salary)\
      VALUES (2,'ALLen',25,'Texas',15000.00)");
c.execute("INSERT INTO persons (id,name,age,address,salary)\
      VALUES (3,'Teddy',23,'Norway',20000.00)");\
c.execute("INSERT INTO persons (id,name,age,address,salary)\
      VALUES (4,'Mark',25,'Rich-Mond',65000.00)");

conn.commit()
print('插入数据库成功')
persons = c.execute("select name,age,address,salary from persons order by age")
result = []
for person in persons:
    value = {}
    value['name'] = person[0]
    value['age'] = person[1]
    value['address'] = person[2]
    result.append(value)
conn.close()
print(type(result))
print(result)
```

- 大概分为五步：1.建立连接，2.获取游标对象，3.执行sql语句，4.提交数据，5.关闭数据库

- pymysql
  - 创建链接

```
from pymysql import *
import json
def connectDB():
    db = connect('127.0.0.1','root','1120','test',charset = 'utf8')
    return db
db = connectDB()
```

    - 返回db对象，将函数赋值给db，方便后面获取游标对象对数据库操作
    - connect('127.0.0.1','root','1120','test',charset = 'utf8')：连接数据库实例化对象，'127.0.0.1'为本机地址，'root'为用户名，'1120'为用户密码，'test'为数据库名称，必须先存在，charset = 'utf8'为字符集
  - 创建数据表

```
def createTable(db):
    cursor = db.cursor()
    sql = '''CREATE TABLE persons
        (id    INT    PRIMARY    KEY    NOT NULL,
        name                      TEXT   NOT NULL,
        age                       INT    NOT NULL,
        address                   CHAR(50),
        salary                    REAL);'''
    try:
        cursor.execute(sql)
        db.commit()
        return True
    except:
        db.rollback()
    return False
```

  - 创建函数

- cursor = db.cursor()：获取游标对象
- 写入sql语句
- cursor.execute(sql)：使用execute函数执行sql语句
- db.commit()：提交数据
- db.rollback()：创建表失败数据回滚，恢复之前的数据
- 插入数据

```
def insertRecords(db):
    cursor = db.cursor()
    try:
        cursor.execute("DELETE FROM persons")
        cursor.execute("INSERT INTO persons (id,name,age,address,salary) \
        VALUES (1,'Paul',32,'California',20000.00)");
        cursor.execute("INSERT INTO persons (id,name,age,address,salary) \
        VALUES (2,'Allen',25,'Texas',15000.00)");
        cursor.execute("INSERT INTO persons (id,name,age,address,salary) \
        VALUES (3,'Teddy',23,'Norway',20000.00)");
        cursor.execute("INSERT INTO persons (id,name,age,address,salary) \
        VALUES (4,'Mark',25,'Rich-Mond',65000.00)");
        db.commit()
        return True
    except Exception as e:
        print(e)
        db.rollback()
```

  - 与上面大同小异，不同的是sql语句，创建字段，关于sql语句可以查阅 旧mysql笔记
- 查询数据

```
def selectRecords(db):
    cursor = db.cursor()
    sql = 'SELECT name,age,salary FROM persons ORDER BY age DESC'
    cursor.execute(sql)
    results = cursor.fetchall()
    print(results)

    fields = ['name','age','salary']
    records = []
    for row in results:
        records.append(dict(zip(fields,row)))
    return json.dumps(records)
```

  - cursor.fetchall()：执行sql查询语句，使用fetchall函数打印记录
  - 使用zip函数双循环遍历数据，组织成字典类型，添加到列表中
  - json.dumps(records)：使用dumps函数转换成json字符串
- 启动函数

```
if createTable(db):
    print('成功创建数据表persons')
else:
    print('persons表已存在')

if insertRecords(db):
    print('成功插入数据')
else:
    print('插入数据失败')
print(selectRecords(db))
db.close()
```

- 注意：在运行这段代码之前，确保有test数据库，并且开启了mysql服务