



CSDN学院 IT实战派

数据存储

—— JSON、CSV、Excel、MySQL

讲师：黄老师

1. JSON文件格式处理。
2. CSV文件格式处理。
3. Excel文件处理。
4. MySQL数据库处理。

1. JSON字符串介绍。
2. JSON字符串写入。
3. JSON字符串读取。

JSON(JavaScript Object Notation, JS 对象标记) 是一种轻量级的数据交换格式。它基于 ECMAScript (w3c制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。更多解释请见：

<https://baike.baidu.com/item/JSON/2462549?fr=aladdin>

JSON支持数据格式：

对象（字典）：使用花括号。

数组（列表）：使用方括号。

字符串类型：字符串必须要用双引号，不能用单引号）。

整形、浮点型、布尔类型还有null类型。

多个数据之间使用逗号分开。

注意：json本质上就是一个字符串。

一、dumps函数：

```
books = [{ 'title' : '钢铁是怎样练成的' , 'price' : 9.8},{ 'title' : '红楼梦' , 'price' : 9.9}]  
json_str = json.dumps(books,ensure_ascii=False)  
print(json_str)
```

二、dump函数：

json模块中除了dumps函数，还有一个dump函数，这个函数可以传入一个文件指针，直接将字符串dump到文件中。示例代码如下：

```
with open('a.json','w',ensure_ascii=False) as fp:  
    json.dump(books,fp)
```

因为json在dump的时候，只能存放ascii的字符，因此会将中文进行转义，这时候我们可以使用ensure_ascii=False关闭这个特性。

在Python中。只有基本数据类型才能转换成JSON格式的字符串。也即：int、float、str、list、dict、tuple。

一、loads函数

```
json_str = '[{"title": "钢铁是怎样练成的", "price": 9.8}, {"title": "红楼梦",  
"price": 9.9}]'  
books = json.loads(json_str, encoding= 'utf-8' )  
print(type(books))  
print(books)
```

二、load函数：

load函数需要接收一个文件指针对象：

```
import json  
with open('a.json','r',encoding='utf-8') as fp:  
    json_str = json.load(fp)  
    print(json_str)
```

1. 读取CSV文件。
2. 写入CSV文件。

一、使用csv.reader :

```
import csv
with open( 'stock.csv' , 'r' ) as fp:
    reader = csv.reader(fp)
    titles = next(reader)
    for x in reader:
        print(x)
```

二、使用csv.DictReader :

这样操作，以后获取数据的时候，就要通过下标来获取数据。如果想要在获取数据的时候通过标题来获取。那么可以使用DictReader。示例代码如下：

```
with open('stock.csv','r') as fp:
    reader = csv.DictReader(fp)
    for x in reader:
        print(x['turnoverVol'])
```


写入CSV文件的两种方式

一、使用writerow：写入数据到csv文件，需要创建一个writer对象，主要用到两个方法。一个是writerow，这个是写入一行。一个是writerows，这个是写入多行。示例代码如下：

```
import csv
headers = [ 'name' , 'age' , 'classroom' ]
values = [( 'zhiliao' ,18, '111' ),( 'wena' ,20, '222' ),( 'bbc' ,21, '111' )]
with open( 'test.csv' , 'w' ,newline= '' ) as fp:
    writer = csv.writer(fp)
    writer.writerow(headers)
    writer.writerows(values)
```

二、使用DictWriter：也可以使用字典的方式把数据写入进去。这时候就需要使用DictWriter了。示例代码如下：

```
import csv
headers = ['name','age','classroom']
values = [{"name":'wenn',"age":20,"classroom":'222'},{"name":'abc',"age":30,"classroom":'333'}]
with open('test.csv','w',newline='') as fp:
    writer = csv.DictWriter(fp,headers)
    writer = csv.writer(fp)
    writer.writerow({'name': 'zhiliao', "age":18, "classroom":'111'})
    writer.writerows(values)
```

1. 读取Excel文件。
2. 编辑Excel文件。

在爬虫开发中，我们主要关注Excel文件的读写，不会过多关心Excel中的一些样式。如果想要读写Excel文件，需要借助到两个库xlrd和xlwt，其中xlrd是用于读的，xlwt是用于写的，安装命令如下：

```
pip install xlrd  
pip install xlwt
```

打开Excel文件：

```
xlrd.open_workbook( "abc.xls" )
```

获取Sheet：

一个Excel中可能有多个Sheet，那么可以通过以下方法来获取想要的Sheet信息：

1. sheet_names：获取所有的sheet的名字。
2. sheet_by_index：根据索引获取sheet对象。
3. sheet_by_name：根据名字获取sheet对象。
4. sheets：获取所有的sheet对象。
5. sheet.nrows：这个sheet中的行数。
6. sheet.ncols：这个sheet中的列数。

获取Cell及其属性：

每个Cell代表的是表格中的一格。以下方法可以方便获取想要的cell：

1. `sheet.cell(row,col)`：获取指定行和列的cell对象。
2. `sheet.row_slice(row,start_col,end_col)`：获取指定行的某几列的cell对象。
3. `sheet.col_slice(col,start_row,end_row)`：获取指定列的某几行的cell对象。
4. `sheet.cell_value(row,col)`：获取指定行和列的值。
5. `sheet.row_values(row,start_col,end_col)`：获取指定行的某几列的值。
6. `sheet.col_values(col,start_row,end_row)`：获取指定列的某几行的值。

Cell的数据类型：

Cell的数据类型：

1. xlrd.XL_CELL_TEXT (Text) ：文本类型。
2. xlrd.XL_CELL_NUMBER (Number) ：数值类型。
3. xlrd.XL_CELL_DATE (Date) ：日期时间类型。
4. xlrd.XL_CELL_BOOLEAN (Bool) ：布尔类型。
5. xlrd.XL_CELL_EMPTY ：空白数据类型。

写入Excel步骤如下：

1. 导入xlwt模块。
2. 创建一个Workbook对象。
3. 创建一个Sheet对象。
4. 把数据写入到Sheet下指定行和列中。如果想要在原来workbook对象上添加新的cell，那么需要调用put_cell来添加。
5. 保存成Excel文件。

EDU

CSDN学院 IT实战派

