

概述

我们先看一下什么是json。

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。

JSON 语法规则

在javascript语言中，一切都是对象。因此，任何支持的类型都可以通过json来表示，例如字符串、数字、对象、数组等。但是对象和数组是比较特殊且常用的两种类型：

- 对象表示为键值对
- 数据由逗号分隔
- 花括号保存对象
- 方括号保存数组

随便看一个json串示例

```
{
  "people": [
    {
      "firstName": "Brett",
      "lastName": "McLaughlin"
    },
    {
      "firstName": "Jason",
      "lastName": "Hunter"
    }
  ]
}
```

对于这个示例，这里不做任何说明，请自行看懂，看不懂就多看几次。

python json解析模块

在Python中，提供了一个标准的json解析模块，所以不需要安装可以直接使用，对于其他第三方json解析库，请自行去找和学习。

怎么使用标准的json解析模块

第一步，导入json模块，如下：

```
import json
```

python json解析最常用的函数:

函数	描述
json.dumps	将Python对象编码成json字符串
json.loads	将已编码的json字符串解码为Python对象

python原始类型与json类型的转化对照表

Python	json
dict	object
list, tuple	array
str, unicode	string
int, long, float	number
True	true
False	false
None	null

这个表对应的转化关系是必须掌握的，一是多练习来记忆，二是死记硬背也要记下来。

看一个基本示例

```
#-*- coding:utf-8 -*-

__author__ = "苦叶子"

import json

if __name__ == "__main__":
    print("python json标准库解析实例")

    # python对象转json对象
    data = [ { 'a' : 1, 'b' : 2, 'c' : 3, 'd' : 4, 'e' : 5 } ]

    json_data = json.dumps(data)

    # 打印出来看下效果
    print("转化前")
    print(type(data))
```

```

print(data)
print("-" * 40)
print("转化后")
print(type(json_data))
print(json_data)

# 将json对象转化成python对象
print()
print("将json对象转化成python对象")
python_data = json.loads(json_data)
print(type(python_data))
print(python_data)

# 请仔细查看结果参照《python原始类型与json类型的转化对照表》对比转化前后的变化

```

对于其他python对象类型与json类型的转换请自行去实践。

格式化输出json

有时为了让json串在console里输出的格式可读性更好，我们需要把json串进行格式化，下面我们看一个实例：

```

#-*- coding:utf-8 -*-

__author__ = "苦叶子"

import json

if __name__ == "__main__":
    print("python json串格式化实例")

    data = [ { 'a' : 1, 'b' : 2, 'c' : 3, 'd' : 4, 'e' : 5 } ]

    json_data = json.dumps(data, sort_keys=True, indent=4, separators=(',', ' '))

    # 打印格式化的json串
    print(json_data)

```

上述的实例演示的都是在内存中进行的，下面我们来看看如何读取文件中的json串或是将内存中的json数据序列化到磁盘中。

从文件加载json格式的内容

将下列json格式的字符串存入到json_data.json文件中。

```

[
  {

```

```
        "a": 1,  
        "b": 2,  
        "c": 3,  
        "d": 4,  
        "e": 5  
    }  
]
```

下面我们演示如何读取json_data.json的内容转化python对象。

```
#-*- coding:utf-8 -*-  
  
__author__ = "苦叶子"  
  
import json  
  
if __name__ == "__main__":  
    print("python 读取json内容文件转化python对象实例")  
  
    fp = open('json_data.json', 'r')  
  
    json_data = json.load(fp)  
    print(type(json_data))  
    print(json_data)  
  
    fp.close()
```

将json对象字符串存文件

下面我们演示如何将python对象转化存json串存入文件。

```
#-*- coding:utf-8 -*-  
  
__author__ = "苦叶子"  
  
import json  
  
if __name__ == "__main__":  
    print("python 写json串实例")  
  
    data = [ { 'a' : 1, 'b' : 2, 'c' : 3, 'd' : 4, 'e' : 5 } ]  
  
    fp = open('json_write.json', 'w')  
  
    # 以可读性格式写入json_write.json文件中  
    json.dump(data, fp, sort_keys=True, indent=4, separators=(',', ' ': ))
```

```
fp.close()
```

小结

请注意json.loads/json.dumps和json.load/json.dump的不同，不带s的函数需要和文件结合。

扫一扫关注微信公众号：

