**python对象与json字符串的相互转化**

在json字符串和python对象的相互转化中，会进行下列变换。

会将python中字典的写法，转化为js中对象的写法。（没有区别）。

会将python中列表的写法，转化为js中array的写法。（没有区别）

会将python中unicode的写法，转化为js中string的写法。（与编码有关）

会将python中的int，long的写法，转化为js中number的写法。（没有区别）

会将python中的True的写法，转化为js中true的写法。（写法有区别）

会将python中的False的写法，转化为js中false的写法。（写法有区别）

会将python中的None的写法，转化为js中null的写法。（写法有区别）

反过来也是对应转换。

js中类型 Python中类型

object dict

array list

string unicode

number (int) int, long

number (real) float

true True

false False

null None

python字典所有的键必须为字符串，值可以是任何对象。包括字典、列表、字符串、数值、布尔值以及null。

## json库

json库集成在了标准库中。使用import json即可引入。

## 将 Python 对象编码成 JSON 字符串

json.dumps（） 将 Python 对象编码成 JSON 字符串

json.dumps(obj, skipkeys=False, ensure\_ascii=True,

check\_circular=True, allow\_nan=True,

cls=None, indent=None, separators=None,

encoding="utf-8", default=None,

sort\_keys=False, \*\*kw)

dumps函数包含一些参数，可以使用参数让 JSON 数据格式化输出。

sort\_keys参数：是否对键值排序。sort\_keys=True

separators：分隔符。separators=(‘,’, ‘: ‘) 参数：两个键值对间的分隔符，键和值中间的分隔符的。

indent：缩进字符数。indent=10

encoding：编码类型。encoding=”utf-8”

## 将 Python 对象存储成 JSON文件

json.dump(obj, fp, \*, skipkeys=False, ensure\_ascii=True,

check\_circular=True, allow\_nan=True,

cls=None, indent=None, separators=None,

default=None, sort\_keys=False, \*\*kw)

## 测试用例

import json

myobject = {

'key2': {

'key31': None,

'key32': False,

'key33': 13.6,

},

'key1': {

'key22': ['value22\_1', 'value22\_2']

}

}

#将对象或对象数组转化为json字符串

jsonstr =json.dumps(myobject, indent=4,sort\_keys=True)

print(jsonstr)

#python对象或对象数组写入json文件：方法1

with open('data.json', 'w') as f:

json.dump(myobject, f)

#python对象或对象数组写入json文件：方法2

import io

f=io.open('data1.json', 'w', encoding='utf-8')

json.dump(myobject, f, ensure\_ascii=False)

注意：测试发现，使用str(dict)，转化为字符串为单引号，

使用json.dumps转化为字符串为双引号

## -Python读取json文件

json.loads（）将已编码的 JSON 字符串解码为 Python 对象

json.loads(s, \*, encoding=None, cls=None,

object\_hook=None, parse\_float=None,

parse\_int=None, parse\_constant=None,

object\_pairs\_hook=None, \*\*kw)

json.load（） 将已编码的 JSON 文件解码为 Python 对象

json.load(fp, \*, cls=None, object\_hook=None,

parse\_float=None, parse\_int=None,

parse\_constant=None, object\_pairs\_hook=None,

\*\*kw)

## 测试用例。运行环境python3.6

#coding:utf-8

import json

#将json字符串转化为对象或对象数组

jsonstr = '{"key2": {"key31": null, "key32": false, "key33": 13.6}, "key1": {"key22": ["value22\_1", "value22\_2"]}}'

obj = json.loads(jsonstr)

print(obj)

#读取json文件转换成python对象或对象数组：方法1

with open('data1.json', 'r') as f:

obj = json.load(f)

print(obj)

#读取json文件转换成python对象或对象数组：方法2

import io

f = io.open('data1.json', 'r', encoding='utf-8')

json\_str = f.readline() #读取文件内容

obj = json.loads(json\_str) #将json字符串转化为对象

print(obj)