JSON 轻量级的数据交换格 规定字符集是UTF-8，字符串必须使用""，Object 的键也必须使用""

JSON最常用的格式是

对象的键值对。

{"firstName": "Brett", "lastName": "McLaughlin"}

表示数组

{ "people":[

{ "firstName": "Brett",

"lastName":"McLaughlin" },

{

"firstName":"Jason",

"lastName":"Hunter"}

]

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

XML格式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

<country>

<name>中国</name>

<province>

<name>黑龙江</name>

<cities>

<city>哈尔滨</city>

<city>大庆</city>

</cities>

</province>

<province>

<name>广东</name>

<cities>

<city>广州</city>

<city>深圳</city>

<city>珠海</city>

</cities>

</province>

</country>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

json格式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{

"name": "中国",

"province": [{

"name": "黑龙江",

"cities": {

"city": ["哈尔滨", "大庆"]

}

}, {

"name": "广东",

"cities": {

"city": ["广州", "深圳", "珠海"]

}

}]

}

整体看下来想递归

xml ：编码的可读性有明显的优势。

json：读起来更像一个数据块，读起来就比较费解了。不过，我们读起来费解的语言，恰恰是适合机器阅读。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

JS 内置两个Json方法

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

对象序列化为Json字符串 ： string = JSON.stringify( {} , [ ] , "")

反序列化成一个JavaScript对象 ： JavaScript对象 = JSON.parse( json.DATA )

修改 JSON 数据

people.musicians[1].lastName = "Rachmaninov";

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

python实现

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

url = "http://fanyi.youdao.com/translate"

//response

r = requests.request("post", url, params=createData("handsome boy"))

d = json.loads(r.text)

#解析 r.text {"type":"EN2ZH\_CN","errorCode":0,"elapsedTime":0,"translateResult":[[{"src":"handsome boy","tgt":"帅哥"}]]}

str = d['translateResult'][0][0]['tgt']

print(str)