**Python Json模块中dumps、loads、dump、load函数介绍**

**1、json.dumps()**

 json.dumps()用于将dict类型的数据转成str，因为如果直接将dict类型的数据写入json文件中会发生报错，因此在将数据写入时需要用到该函数。

**2、json.loads()**

json.loads()用于将str类型的数据转成dict。

**3、json.dump()**

json.dump()用于将dict类型的数据转成str，并写入到json文件中。下面两种方法都可以将数据写入json文件

**4、json.load()**

json.load()用于从json文件中读取数据。

import json  
data = {  
 'name': 'ACME',  
 'shares': 100,  
 'price': 542.23  
}  
json\_str = json.dumps(data)  
data = json.loads(json\_str)

print(json\_str,type(json\_str))   
#{"name": "ACME", "shares": 100, "price": 542.23} <class 'str'>  
print(data,type(data))   
#{'name': 'ACME', 'shares': 100, 'price': 542.23} <class 'dict'>

# pandas.to\_json官方文档阅读

# DataFrame转换成json

#例子  
import pandas as pd  
df = pd.DataFrame([['a', 'b'], ['c', 'd']],  
 index=['row 1', 'row 2'],  
 columns=['col 1', 'col 2'])  
df

# split 将索引总结到索引，列名到列名，数据到数据。将三部分都分开了了

# records 以columns：values的形式输出

# index 以index：{columns：values}...的形式输出

**colums 以columns:{index:values}的形式输出**

**values 直接输出值**

**table形式**

**df.to\_json(orient='split')**  
# '{"columns":["col 1","col 2"],"index":["row 1","row 2"],"data":[["a","b"],["c","d"]]}'  
**df.to\_json(orient='records')**  
# '[{"col 1":"a","col 2":"b"},{"col 1":"c","col 2":"d"}]'  
**df.to\_json(orient='index')**  
# '{"row 1":{"col 1":"a","col 2":"b"},"row 2":{"col 1":"c","col 2":"d"}}'  
**df.to\_json(orient='columns')**  
# '{"col 1":{"row 1":"a","row 2":"c"},"col 2":{"row 1":"b","row 2":"d"}}'  
**df.to\_json(orient='values')**  
# '[["a","b"],["c","d"]]'  
**df.to\_json(orient='table')**  
# '{"schema": {"primaryKey":["index"],"fields":[{"name":"index","type":"string"},{"name":"col 1","type":"string"},{"name":"col 2","type":"string"}],"pandas\_version":"0.20.0"},"data": [{"index":"row 1","col 1":"a","col 2":"b"},{"index":"row 2","col 1":"c","col 2":"d"}]}'