- Python教程
  - Python简介
  - 安装Python
  - 第一个Python程序
  - 使用文本编辑器
  - 运行脚本
  - 输入和输出
  - 字符串和编码
  - 使用list和tuple
  - 条件判断
  - 模式匹配
  - 循环
  - 使用dict和set
  - 再议可变对象
  - 函数
  - 高级特性
  - 函数式编程
  - 模块
  - 面向对象编程
  - 面向对象高级编程
  - 错误、调试和测试
  - IO编程
  - 进程和线程
  - 正则表达式
  - 常用内建模块
  - 常用第三方模块
  - virtualenv
  - 图形界面
  - 网络编程
  - 实战
  - 调试
  - 异步IO

关于作者

微信号：（未设置）

关注

廖雪峰 自己的Python课程

Python商业爬虫全解码

让天下没有爬不到的数据！

Python爬虫 + 数据分析

Java高级架构师

更专业 更权威

课程内专享 · 更新实时跟进

BAT/大厂实战 · 项目经验丰富

抗摩雪峰老师

廖雪峰老师

自己的Java课程

Java高级架构师

更专业 更权威

课程内专享 · 更新实时跟进

BAT/大厂实战 · 项目经验丰富

抗摩雪峰老师

unicef 联合国儿童基金

一个纸杯 一个吸管 一个卫生间

4个小时内 1条手机信息

unicef 联合国儿童基金

一个纸杯 一个吸管 一个卫生间

4个小时内 1条手机信息

Top

序列化

最新 2020/20

如果你使用过Python，并有过把数据保存到文件中，比如，定义一个dict：

```
d = dict(name='Bob', age=20, score=88)
```

可以顺利地保存到文件，比如保存到一个text文件：

```
$ cat dump.txt
{"age": 20, "name": "Bob", "score": 88}
```

现在我们要把这个字符串从文件读入内存，并反序列化成一个dict，这项工作称为解序列（deserialize），或反序列化（unserialize）。

Python的pickle模块实现了序列化和反序列化的功能。

```
>>> import pickle
>>> d = dict(name='Bob', age=20, score=88)
>>> pickle.dumps(d)
b'\x80\x03}q\\x00(X\\x03\\x00\\x00\\x00ageq\\x01K\\x14X\\x05\\x00\\x00\\x00scoreq\\x02K\\xb8X\\x04\\x00\\x00\\x00nameq\\x03X\\x03\\x00\\x00\\x00Bobq\\x04u.'
```

pickle.dumps()方法把任何对象序列化成一个str，然后，就可以把这个str写入文件。或者把这个序列化的str发到网络上进行传输（Pickle）。

```
>>> f = open('dump.txt', 'wb')
>>> pickle.dump(d, f)
>>> f.close
>>>
```

看到文件的内容，以二进制方式打开：

```
$ cat dump.txt
cq
(X
age
q
K
X
score
q
K
X
name
q
X
Bob
q
u.
```

pickle.dumps()方法把任何对象序列化成一个str，然后，就可以把这个str写入文件。或者把这个序列化的str发到网络上进行传输（Pickle）。

```
>>> f = open('dump.txt', 'rb')
>>> d = pickle.load(f)
>>> f.close
>>> d
{'age': 20, 'score': 88, 'name': 'Bob'}
```

由于Python是跨平台的语言，序列化过程只能是Python的内部数据格式，否则其他平台就无法读写pickle，所以，跨进程传输的序列化还是以文本格式的JSON最合适。

JSON表示的对象类型和JavaScript语言的类型非常接近，JSON的null对应的是Python的None。

JavaScript	Python
{}	dict
[]	list
"string"	str
1234.56	int或float
true/false	True/False
null	None

Python内置的json模块提供了非常完整的Python对象到JSON格式的转换。我们先看看如何把Python对象变成一个JSON：

```
>>> import json
>>> d = dict(name='Bob', age=20, score=88)
>>> json.dumps(d)
'{"age": 20, "score": 88, "name": "Bob"}'
```

dumps()方法返回一个str，内容是标准的JSON。类似的，dump()方法可以直接把JSON写入一个file-like Object。

要把JSON反序列化为Python对象，用loads()或者对应的load()方法，前者把一个str反序列化为Python对象，后者从一个file-like Object中读写内容。

```
>>> sx = '{"name": "Bob", "age": 20, "score": 88}'
>>> d = loads(sx)
>>> d
{'name': 'Bob', 'age': 20, 'score': 88}
```

由于JSON标准规定JSON编码是UTF-8，所以我们总是能正确地在网络上传输字符串。

JSONlint

JSONlint是一个在线的JSON格式校验工具，可以帮助检查JSON格式是否正确。

JSONP

Python的pickle可以序列化任意对象，但是，这种序列化的结果只能被Python程序读取，如果需要跨语言的序列化，可参考Google的Protocol Buffer。

练习

请编写一个函数，将任意对象变成dict，以便能调用json.dumps()把该对象序列化为JSON：

```
def person(name, age):
    return {'name': name,
            'age': age}

import json
print(dumps(p, default=person))
```

提交

参考源码

- use_pickle.py
- use_json.py

喜欢我的话记得点赞哦谢谢，请关注我的公众号：

关注微信公众号