

Python JSON

本章节我们将为大家介绍如何使用 Python 语言来编码和解码 JSON 对象。

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，易于人阅读和编写。

JSON 函数

使用 JSON 函数需要导入 json 库：`import json`。

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

json.dumps

`json.dumps` 用于将 Python 对象编码成 JSON 字符串。

语法

```
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)
```

实例

以下实例将数组编码为 JSON 格式数据：

```
#!/usr/bin/python
import json

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

json = json.dumps(data)
print json
```

以上代码执行结果为：

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

使用参数让 JSON 数据格式化输出：

```
>>> import json
>>> print json.dumps({'a': 'Runoob', 'b': 7}, sort_keys=True, indent=4, separators=(',', ': '))
```

```
{
    "a": "Runoob",
    "b": 7
}
```

python 原始类型向 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

json.loads

json.loads 用于解码 JSON 数据。该函数返回 Python 字段的数据类型。

语法

```
json.loads(s[, encoding[, cls[, object_hook[, parse_float[, parse_int[, parse_constant[, object_pairs_hook[, **kw]]]]]]])
```

实例

以下实例展示了Python 如何解码 JSON 对象：

```
#!/usr/bin/python
import json

jsonData = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

text = json.loads(jsonData)
print text
```

以上代码执行结果为：

```
{u'a': 1, u'c': 3, u'b': 2, u'e': 5, u'd': 4}
```

json 类型转换到 python 的类型对照表：

JSON	Python
object	dict
array	list
string	unicode
number (int)	int, long
number (real)	float
true	True
false	False
null	None

更多内容参考：<https://docs.python.org/2/library/json.html>。

使用第三方库：Demjson

Demjson 是 python 的第三方模块库，可用于编码和解码 JSON 数据，包含了 JSONLint 的格式化及校验功能。

Github 地址：<https://github.com/dmeranda/demjson>

官方地址：<http://deron.meranda.us/python/demjson/>

环境配置

在使用 Demjson 编码或解码 JSON 数据前，我们需要先安装 Demjson 模块。本教程我们会下载 [Demjson](#) 并安装：

```
$ tar -xvzf demjson-2.2.3.tar.gz
$ cd demjson-2.2.3
$ python setup.py install
```

更多安装介绍查看：<http://deron.meranda.us/python/demjson/install>

JSON 函数

函数	描述
encode	将 Python 对象编码成 JSON 字符串
decode	将已编码的 JSON 字符串解码为 Python 对象

encode

Python encode() 函数用于将 Python 对象编码成 JSON 字符串。

语法

```
demjson.encode(self, obj, nest_level=0)
```

实例

以下实例将数组编码为 JSON 格式数据：

```
#!/usr/bin/python
import demjson

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

json = demjson.encode(data)
print json
```

以上代码执行结果为：

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

decode

Python 可以使用 demjson.decode() 函数解码 JSON 数据。该函数返回 Python 字段的数据类型。

语法

```
demjson.decode(self, txt)
```

实例

以下实例展示了Python 如何解码 JSON 对象：

```
#!/usr/bin/python
import demjson

json = '{"a":1,"b":2,"c":3,"d":4,"e":5}';

text = demjson.decode(json)
print text
```

以上代码执行结果为：

```
{u'a': 1, u'c': 3, u'b': 2, u'e': 5, u'd': 4}
```

[← Python IDE](#)

[Python 中文编码 →](#)

 [点我分享笔记](#)