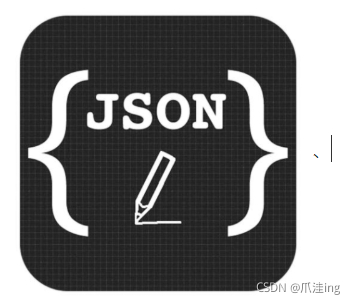
# JSON学习笔记

## JSON简介

* JSON，全称是 JavaScript Object Notation，即 JavaScript对象标记法
* json的读音J-S-O-N，也可以读[ˈdʒeɪs(ə)n]
* JSON是一种轻量级的数据交换格式
* 它基于ECMAScript的一个子集
* ECMAScript（European Computer Manufacturers Association, 欧洲计算机协会制定的js规范）
* 采用完全独立于编程语言的文本格式来存储和表示数据，也使用了类似于C语言家族的习惯
* 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。
* 简洁和清晰的层次结构使得JSON成为理想的数据交换语言。
* 大多数编程语言都支持JSON（有些本身就支持，有些可以通过第三方库得到支持），例如 JavaScript、Java、PHP、Python、C++等



## JSON和XML

JSON常被拿来与XML做比较，因为JSON的诞生本来就多多少少要有取代XNL的意思

相比XML，JSON的优势如下:

* 结构简单、紧凑
* JSON格式所占的存储空间更小
* 可读性高：JSON 的结构有利于程序员阅读
* 没有结束标签,长度更短,读写更快
* 能够直接被JavaScript解释器解析
* 可以使用数组

### 两者比较

* JSON：

1. {
2. "name":"兮动人",
3. "age":22,
4. "fruits":["apple","pear","grape"]
5. }

* XML：

1. <root>
2. <name>兮动人</name>
3. <age>22</age>
4. <fruits>apple</fruits>
5. <fruits>pear</fruits>
6. <fruits>grape</fruits>
7. </root>

### JSON 的不足

* 只有一种数字类型，只支持 IEEE-754 双精度浮点格式
* 没有日期类型：在JSON中只能通过日期的字符串（例如：1970-01-01）或者时间戳（例如：1632366361）来表示日期；
* 在JSON中无法添加注释
* 冗长，并不是最简洁的数据交换格式

## JSON的语法规则

JSON在线检查语法：https://www.json.cn/

### JSON建构于两种结构：

* “名称/值”对的集合（A collection of name/value pairs）

不同的语言中，它被理解为对象（object），纪录（record），结构（struct），字典（dictionary），哈希表（hash table），有键列表（keyed list），或者关联数组 （associative array）。

* 值的有序列表（An ordered list of values）

在大部分语言中，它被理解为数组（array）。

### 对象（0bject)

对象用大括号(“{}”）括起来

大括号里是一系列的“名称/值对”

两个并列的数据之间用逗号(“,”）隔开

#### 注意:

* 对象必须包裹在花括号{ }中；
* 最后一个“名称/值对“之后不要加逗号；
* 对象中的内容需要以键/值对的形式编写；
* 键必须是字符串类型，即使用双引号" "将键包裹起来；
* 值可以是任意 JSON 中支持的数据类型（例如字符串、数字、对象、数组、布尔值、null 等）；
* 键和值之间使用冒号进行分隔；
* 不同键/值对之间使用逗号进行分隔，使用英文的逗号（“,”)，不要用中文的逗号(“，”)；
* 对象中的最后一个键/值对末尾不需要添加逗号。

#### 示例

1. {
2. "name": "xdr630",
3. "favorite": "programming"
4. }

### 数组（Array）

数组表示一系列有序的值，用方括号(“[]”)包围起来，并列的值之间用逗号分隔

#### 示例

以下的数组是合法的:

[1,2,"three","four",true,false,null,[1,2],{"name":"兮动人"}]

### 名称/值对（Name/Value）

名称/值对(name/value）组合成数组和对象。

名称是一个字符串，要用双引号括起来，不能用单引号，也不能没有引号

值的类型只有七种:

* 数值(number)（整数或浮点数）；
* 字符串（string)（需要在双引号中定义）；
* 对象（object)（整数或浮点数）；
* 数组（array)（在方括号中定义）；
* 布尔值（true 或 false）；
* null（空）

不能有这之外的类型，例如undefined、函数等

* JSON是一段包裹在花括号中的数据，数据由若干键/值对组成；
* 键和值之间使用冒号分隔；
* 不同的键/值对之间使用逗号分隔；
* 键需要以字符串的形式定义（即使用双引号包裹，注意：不能使用单引号）；
* 值可以是数字、字符串、布尔值、数组、对象、null；
* 键必须是唯一的，不能重复，否则后定义的键/值对会覆盖前面定义的键/值对；
* JSON 中不可以使用八进制或十六进制表示数字。

## JSON注释

JSON有一个非常明显的缺点，JSON不能像编程语言那样添加注释，JSON中的任何内容都会看作数据的一部分。

之所以不允许添加注释，是因为JSON主要是用来存储数据的，过多的注释会对数据的可读性造成影响，同时也会造成数据体积变大，进而影响数据传输、解析的速度。

### 方法一

想要在JSON中添加注释，我们可以在要添加注释键/值对的前面（或后面）添加一个同名的键，并在键名中添加一个特殊字符加以区分，例如@、#、?、\_、/等，然后在与键对应的值中定义注释的内容。

#### 示例

1. {
2. "@name": "网站名称",
3. "name": "C语言中文网",
4. "\_url": "网址",
5. "url": "http://c.biancheng.net/",
6. "course": "JSON 教程",
7. "@charge": "0=收费; 1=免费",
8. "charge": 1,
9. "#list": "教程目录",
10. "list": [
11. "JSON数据类型",
12. "JSON对象",
13. "JSON数组"
14. ],
15. "message": {
16. "code": 0,
17. "message": "OK",
18. "#data": {
19. "#id": "用户ID",
20. "#type": "0=正常; 1=异常",
21. "#name": "姓名",
22. "#createTime": "创建时间(yyyy-MM-dd)"
23. },
24. "data": {
25. "id": "12345",
26. "type": 0,
27. "name": "我的名字",
28. "createTime": "2020-05-20"
29. }
30. }
31. }

### 方法二

除了可以在键名中添加特殊字符外，也可以直接以“comment”、“\_comment” 或 “\_\_comment”作为键名来定义注释，如下例所示：

#### 示例

1. {
2. "\_comment": "C语言中文网（http://c.biancheng.net/）—— 一个在线学习编程的网站",
3. "course": {
4. "name": "JSON 教程",
5. "url": "http://c.biancheng.net/json/index.html"
6. }
7. }

## JSON对比工具

JSON 是 Web 开发领域中最常用的数据传输格式之一，因为 JSON 的可读性较高，对于一些简单的 JSON 数据，我们不需要借助任何工具就可以轻易的读取。但对于复杂的 JSON 数据就需要借助工具才行，本节我们来介绍一些 JSON 对比工具。

要对比 JSON 数据，最简单的方法就是借助在线工具，在线工具即不需要安装，也不需要配置环境就可以拿来使用，在网络中可以找到许多优秀的 JSON 在线对比工具，如下所示：

https://jsoncompare.org/

https://jsoncompare.com/#!/diff/

https://jsondiff.org/

http://jsondiff.com/

http://www.jsondiff.com/

https://extendsclass.com/json-diff.html

https://www.textcompare.org/json/

https://jsoncompare.org/

https://alldiff.in/

https://bestjsonformatter.com/json-diff/

https://www.jsonformatting.com/json-compare/

https://url-decode.com/tool/json-diff/

https://www.sojson.com/jsondiff.html

https://www.json.cn/

http://json365.com/json\_compare

https://jsonlint.com/

## 参考资料

1. 介绍 JSON http://www.json.org/json-zh.html
2. JSON是什么？如何正确理解？ https://blog.csdn.net/weixin\_41819731/article/details/81414968
3. JSON 基本使用 https://blog.csdn.net/qq\_41684621/article/details/113851644
4. JSON教程（非常详细） https://blog.csdn.net/qq\_41854911/article/details/122006847
5. JSON详解 https://blog.csdn.net/m0\_46571920/article/details/121237388

————————————————

版权声明：本文引用以上文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。