**JSON 语法规则**

**在 JS 语言中，一切都是对象**。因此，任何支持的类型都可以通过 JSON 来表示，例如字符串、数字、对象、数组等。但是对象和数组是比较特殊且常用的两种类型：

* 对象表示为键值对
* 数据由逗号分隔
* 花括号保存对象
* 方括号保存数组

**JSONJSON 键/值对**

JSON 键值对是用来保存 JS 对象的一种方式，和 JS 对象的写法也大同小异，键/值对组合中的键名写在前面并用双引号 "" 包裹，使用冒号 : 分隔，然后紧接着值：

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1 | {"firstName": "John"} |

这很容易理解，等价于这条 JavaScript 语句：

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1 | {firstName = "John"} |

**JSONJSON 与 JS 对象的关系**

很多人搞不清楚 JSON 和 Js 对象的关系，甚至连谁是谁都不清楚。其实，可以这么理解：

**JSON 是 JS 对象的字符串表示法，它使用文本表示一个 JS 对象的信息，本质是一个字符串。**

如

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1 | var obj = {a: 'Hello', b: 'World'}; //这是一个对象，注意键名也是可以使用引号包裹的 |

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1 | var json = '{"a": "Hello", "b": "World"}'; //这是一个 JSON 字符串，本质是一个字符串 |

**JSONJSON 和 JS 对象互转**

要实现从对象转换为 JSON 字符串，使用 JSON.stringify() 方法：

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1 | var json = JSON.stringify({a: 'Hello', b: 'World'}); //结果是 '{"a": "Hello", "b": "World"}' |

要实现从 JSON 转换为对象，使用 JSON.parse() 方法：

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1 | var obj = JSON.parse('{"a": "Hello", "b": "World"}'); //结果是 {a: 'Hello', b: 'World'} |

## 常用类型

[编辑](javascript:;)

**在 JS 语言中，一切都是对象**。因此，任何支持的类型都可以通过 JSON 来表示，例如字符串、数字、对象、数组等。但是对象和数组是比较特殊且常用的两种类型。

对象：对象在 JS 中是使用花括号包裹 {} 起来的内容，数据结构为 {key1：value1, key2：value2, ...} 的键值对结构。在面向对象的语言中，key 为对象的属性，value 为对应的值。键名可以使用整数和字符串来表示。值的类型可以是任意类型。

数组：数组在 JS 中是方括号 [] 包裹起来的内容，数据结构为 ["java", "javascript", "vb", ...] 的索引结构。在 JS 中，数组是一种比较特殊的数据类型，它也可以像对象那样使用键值对，但还是索引使用得多。同样，值的类型可以是任意类型。

## JSON基础示例

[编辑](javascript:;)

简单地说[1]  ，JSON 可以将 JavaScript 对象中表示的一组数据转换为字符串，然后就可以在网络或者程序之间轻松地传递这个字符串，并在需要的时候将它还原为各编程语言所支持的数据格式，例如在 PHP 中，可以将 JSON 还原为数组或者一个基本对象。在用到AJAX时，如果需要用到数组传值，这时就需要用JSON将数组转化为字符串。

### JSON表示对象

JSON最常用的格式是对象的 键值对。例如下面这样：

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1 | {"firstName": "Brett", "lastName": "McLaughlin"} |

### JSON表示数组

和普通的 JS 数组一样，JSON 表示数组的方式也是使用方括号 []。

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | {      "people":[      {    "firstName": "Brett",    "lastName":"McLaughlin"    },      {    "firstName":"Jason",    "lastName":"Hunter"      }        ]    } |

这不难理解。在这个示例中，只有一个名为 people的变量，值是包含两个条目的数组，每个条目是一个人的记录，其中包含名和姓。上面的示例演示如何用括号将记录组合成一个值。当然，可以使用相同的语法表示更过多的值（每个值包含多个记录）。

在处理 JSON 格式的数据时，没有需要遵守的预定义的约束。所以，在同样的数据结构中，可以改变表示数据的方式，也可以使用不同方式表示同一事物。

如前面所说，除了对象和数组，你也可以简单地使用字符串或者数字等来存储简单的数据，但这样并没有多大意义。

## JSON和XML的比较

[编辑](javascript:;)

### JSON可读性

JSON和[XML](https://baike.baidu.com/item/XML" \t "_blank)的可读性可谓不相上下，一边是简易的语法，一边是规范的标签形式，很难分出胜负。

### JSON可扩展性

XML天生有很好的扩展性，JSON当然也有，没有什么是XML可以扩展而JSON却不能扩展的。不过JSON在Javascript主场作战，可以存储Javascript复合对象，有着xml不可比拟的优势。

### JSON编码难度

XML有丰富的[编码工具](https://baike.baidu.com/item/%E7%BC%96%E7%A0%81%E5%B7%A5%E5%85%B7" \t "_blank)，比如Dom4j、JDom等，JSON也有提供的工具。无工具的情况下，相信熟练的开发人员一样能很快的写出想要的xml文档和JSON[字符](https://baike.baidu.com/item/%E5%AD%97%E7%AC%A6)串，不过，xml文档要多很多结构上的字符。

### JSON解码难度

[XML](https://baike.baidu.com/item/XML)的解析方式有两种：

一是通过文档模型解析，也就是通过父标签索引出一组标记。例如：xmlData.getElementsByTagName("tagName")，但是这样是要在预先知道文档结构的情况下使用，无法进行通用的封装。

另外一种方法是遍历节点（document 以及 childNodes）。这个可以通过[递归](https://baike.baidu.com/item/%E9%80%92%E5%BD%92" \t "_blank)来实现，不过解析出来的数据仍旧是形式各异，往往也不能满足预先的要求。

凡是这样可扩展的结构数据解析起来一定都很困难。

JSON也同样如此。如果预先知道JSON结构的情况下，使用JSON进行数据传递简直是太美妙了，可以写出很实用美观可读性强的代码。如果你是纯粹的前台开发人员，一定会非常喜欢JSON。但是如果你是一个应用开发人员，就不是那么喜欢了，毕竟xml才是真正的结构化[标记语言](https://baike.baidu.com/item/%E6%A0%87%E8%AE%B0%E8%AF%AD%E8%A8%80" \t "_blank)，用于进行数据传递。

而如果不知道JSON的结构而去解析JSON的话，那简直是噩梦。费时费力不说，代码也会变得冗余拖沓，得到的结果也不尽人意。但是这样也不影响众多前台开发人员选择JSON。因为json.js中的toJSONString()就可以看到JSON的字符串结构。当然不是使用这个字符串，这样仍旧是噩梦。常用JSON的人看到这个字符串之后，就对JSON的结构很明了了，就更容易的操作JSON。

以上是在Javascript中仅对于数据传递的xml与JSON的解析。在Javascript地盘内，JSON毕竟是主场作战，其优势当然要远远优越于xml。如果JSON中存储Javascript复合对象，而且不知道其结构的话，我相信很多程序员也一样是哭着解析JSON的。

除了上述之外，JSON和[XML](https://baike.baidu.com/item/XML" \t "_blank)还有另外一个很大的区别在于有效数据率。JSON作为数据包格式传输的时候具有更高的效率，这是因为JSON不像XML那样需要有严格的闭合标签，这就让有效数据量与总数据包比大大提升，从而减少同等数据流量的情况下，网络的传输压力[2]  。

### JSON实例比较

XML和JSON都使用[结构化方法](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E6%96%B9%E6%B3%95" \t "_blank)来标记数据，下面来做一个简单的比较。

用XML表示中国部分省市数据如下：

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | <?xml version="1.0" encoding="utf-8"?>  <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>      <province>          <name>台湾</name>          <cities>              <city>台北</city>              <city>高雄</city>          </cities>      </province>      <province>          <name>新疆</name>          <cities>              <city>乌鲁木齐</city>          </cities>      </province>  </country> |

用JSON表示如下：

[?](https://baike.baidu.com/item/JSON/2462549?fr=aladdin)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | {      "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"]          }      }, {          "name": "台湾",          "cities": {              "city": ["台北", "高雄"]          }      }, {          "name": "新疆",          "cities": {              "city": ["乌鲁木齐"]          }      }]  } |

可以看到，JSON 简单的语法格式和清晰的层次结构明显要比 XML 容易阅读，并且在数据交换方面，由于 JSON 所使用的字符要比 XML 少得多，可以大大得节约传输数据所占用得带宽。

## JSON校验工具

[编辑](javascript:;)

### JSON前言

JSON格式取代了xml给网络传输带来了很大的便利,但是却没有了xml的一目了然,尤其是json数据很长的时候,我们会陷入繁琐复杂的数据节点查找中。

但是国人的一款在线工具 BeJson 、SoJson在线工具让众多程序员、新接触JSON格式的程序员更快的了解JSON的结构，更快的精确定位JSON格式错误。

### JSON功能

**1 JSON格式化校验**

很多人在得到JSON数据后,一时没有办法判断JSON数据格式是否正确,是否少或多符号而导致程序不能解析,这个功能正好能帮助大家来完成JSON格式的校验。

**2 JSON视图**

想必很多程序员都会遇到当找一个节点的时候,会发现如果直接对着一行行数据无从下手,就算知道哪个位置,还要一个节点一个节点的往下找,万一一不留神又得从头开始找的麻烦事。

有了这个功能,一切JSON数据都会变成视图格式,一目了然,什么对象下有多少[数组](https://baike.baidu.com/item/%E6%95%B0%E7%BB%84" \t "_blank),一个数组下有多少对象。

这个功能非常实用。不光有视图功能还有格式化、压缩、转义、校验功能。总之很强大。

**3 压缩转义**

程序员在写JSON语句[测试用例](https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E7%94%A8%E4%BE%8B" \t "_blank)的时候,很多时候为了方便直接写了个JSON字符串做测试,但是又陷入了无止境的双引号转义的麻烦中。这款功能集压缩、转义于一身,让你在写测试用例的时候,如鱼得水。

**4 JSON在线编辑器**

如果你现在的电脑刚巧没有装你所熟悉的编辑器,如果你想针对拿到的JSON数据的某个节点做数据修改时,这个功能可以满足你的需求。

**5 在线发送JSON数据**

大家都知道,JSON用的最多的还是web项目的开发,那你要测试一个接口是否能准确的接受JSON数据,那你就得写一个页面发送JSON字符串,重复的做着这件事。随着这个功能的横空出世,你可以摆脱写测试页面了,因为这个功能可以将指定的JSON数据发送指定的url,方便吧。

**6 JSON着色**

很多人在写文档时，总希望文档能一目了然,但是面对着白底黑字的JSON数据总是提不起精神没关系,使用这个功能,所有的关键字都[3]  会被着色,数据结构一目了然。

**7 JSON-XML互转**

顾名思义，将JSON格式的数据转化成XML[3]  格式、或者XML格式的数据转化成JSON格式，一切都不是问题。

**8 JSON-VIEW**

JSON查看实用工具，在开发过程中(windows平台中)可以对JSON数据进行格式化和视图显示。

**9 它和xml一样都是一种数据交换格式**

### JSONJSON的最小化

Eclipse RAP的一个提交者也是领导者的Ralf Sternberg，他只用了十个类就把快速的和轻量级的库整合到了一起。显然，使用精益和解析法真正地改善了服务器的性能，因为服务器进程以更高的效率为大量的客户创建了JSON信息。在外部JSON中不存在依赖关系，代码很容易管理，而且也不会占用很多内存。对于你的全部JSON项目来说，这还远远不够，但这确实带来了几件好事。[