



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

## JSON 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：

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

要实现从 JSON 转换为对象，使用 `JSON.parse()` 方法：

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

## 常用类型

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。

## 基础示例

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，如果需要用数组传值，这时就需要用 JSON 将数组转化为字符串。

### 表示对象

JSON 最常用的格式是对象的 键值对。例如下面这样：

1	<code>{"firstName": "Brett", "lastName": "McLaughlin"}</code>
---	---

### 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 `[]`。

1	<code>{</code>
2	<code></code>
3	<code></code>
4	<code>"people": [</code>

```

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

```

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。

## 实例比较

XML和JSON都使用[结构化方法](#)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：

```

1    <?xml version="1.0" encoding="utf-8"?>
2    <country>
3        <name>中国</name>
4        <province>
5            <name>黑龙江</name>
6            <cities>
7                <city>哈尔滨</city>
8                <city>大庆</city>
9            </cities>
10       </province>
11       <province>
12           <name>广东</name>
13           <cities>
14               <city>广州</city>
15               <city>深圳</city>
16               <city>珠海</city>

```

```

14         </cities>
15     </province>
16     <province>
17         <name>台湾</name>
18         <cities>
19             <city>台北</city>
20             <city>高雄</city>
21         </cities>
22     </province>
23     <province>
24         <name>新疆</name>
25         <cities>
26             <city>乌鲁木齐</city>
27         </cities>
28     </province>
29 </country>
30
31
32

```

用JSON表示如下：

```

1  {
2      "name": "中国",
3      "province": [{
4          "name": "黑龙江",
5          "cities": {
6              "city": ["哈尔滨", "大庆"]
7          }
8      }, {
9          "name": "广东",
10         "cities": {
11             "city": ["广州", "深圳", "珠海"]
12         }
13     }, {
14         "name": "台湾",
15         "cities": {
16             "city": ["台北", "高雄"]
17         }
18     }, {
19         "name": "新疆",
20         "cities": {
21             "city": ["乌鲁木齐"]
22         }
23     }
24 ]

```

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。

WEB技术	
HTML	<a href="#">XHTML</a> · <a href="#">HTML 5</a> · <a href="#">CSS</a> · <a href="#">TCP/IP</a>
XML	<a href="#">XML</a> · <a href="#">XSL</a> · <a href="#">XSLT</a> · <a href="#">XSL-FO</a> · <a href="#">XPath</a> · <a href="#">XPointer</a> · <a href="#">XLink</a> · <a href="#">DTD</a> · <a href="#">XML Schema</a> · <a href="#">DOM</a> · <a href="#">XForms</a> · <a href="#">SOAP</a> · <a href="#">WSDL</a> · <a href="#">RDF</a> · <a href="#">RSS</a> · <a href="#">WAP</a> · <a href="#">Web Services</a>
Web脚本	<a href="#">JavaScript</a> · <a href="#">HTML DOM</a> · <a href="#">DHTML</a> · <a href="#">VBScript</a> · <a href="#">AJAX</a> · <a href="#">jQuery</a> · <b>JSON</b> · <a href="#">E4X</a> · <a href="#">WMLScript</a>
Serv脚本	<a href="#">SQL</a> · <a href="#">ASP</a> · <a href="#">ADO</a> · <a href="#">PHP</a>
.NET	<a href="#">Microsoft.NET</a> · <a href="#">.NET Mobile</a>
多媒体	<a href="#">SMIL</a> · <a href="#">SVG</a>