JSON 简介

* [JSON 教程](http://www.w3school.com.cn/json/index.asp)
* [JSON 语法](http://www.w3school.com.cn/json/json_syntax.asp)

**亲自试一试 - 实例**

通过我们的编辑器，您可以在线编辑 JavaScript 代码，然后通过点击一个按钮来查看结果：

<html>

<body>

<h2>在 JavaScript 中创建 JSON 对象</h2>

<p>

Name: <span id="jname"></span><br />

Age: <span id="jage"></span><br />

Address: <span id="jstreet"></span><br />

Phone: <span id="jphone"></span><br />

</p>

<script type="text/javascript">

var JSONObject= {

"name":"Bill Gates",

"street":"Fifth Avenue New York 666",

"age":56,

"phone":"555 1234567"};

document.getElementById("jname").innerHTML=JSONObject.name

document.getElementById("jage").innerHTML=JSONObject.age

document.getElementById("jstreet").innerHTML=JSONObject.street

document.getElementById("jphone").innerHTML=JSONObject.phone

</script>

</body>

</html>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=json_create)

**类似 XML**

* **JSON 是纯文本**
* **JSON 具有“自我描述性”（人类可读）**
* **JSON 具有层级结构（值中存在值）**
* **JSON 可通过 JavaScript 进行解析**
* **JSON 数据可使用 AJAX 进行传输**

**相比 XML 的不同之处**

* **没有结束标签**
* **更短**
* **读写的速度更快**
* **能够使用内建的 JavaScript eval() 方法进行解析**
* **使用数组**
* **不使用保留字**

**为什么使用 JSON？**

对于 AJAX 应用程序来说，JSON 比 XML 更快更易使用：

**使用 XML**

* **读取 XML 文档**
* **使用 XML DOM 来循环遍历文档**
* **读取值并存储在变量中**

**使用 JSON**

* **读取 JSON 字符串**
* **用 eval() 处理 JSON 字符串**

**JSON 语法是 JavaScript 语法的子集。**

**JSON 语法规则**

JSON 语法

**JSON 语法是 JavaScript 语法的子集。**

* **数据在名称/值对中**
* **数据由逗号分隔**
* **花括号保存对象**
* **方括号保存数组**

**JSON 名称/值对**

JSON 数据的书写格式是：名称/值对。

名称/值对包括字段名称（在双引号中），后面写一个冒号，然后是值：

"firstName" : "John"

这很容易理解，等价于这条 JavaScript 语句：

firstName = "John"

**JSON 值**

JSON 值可以是：

* 数字（整数或浮点数）
* 字符串（在双引号中）
* 逻辑值（true 或 false）
* 数组（在方括号中）
* 对象（在花括号中）
* null

**JSON 对象**

JSON 对象在花括号中书写：

对象可以包含多个名称/值对：

{ "firstName":"John" , "lastName":"Doe" }

这一点也容易理解，与这条 JavaScript 语句等价：

firstName = "John"

lastName = "Doe"

**JSON 数组**

JSON 数组在方括号中书写：

数组可包含多个对象：

{

"employees": [

{ "firstName":"John" , "lastName":"Doe" },

{ "firstName":"Anna" , "lastName":"Smith" },

{ "firstName":"Peter" , "lastName":"Jones" }

]

}

在上面的例子中，对象 "employees" 是包含三个对象的数组。每个对象代表一条关于某人（有姓和名）的记录。

**JSON 使用 JavaScript 语法**

因为 JSON 使用 JavaScript 语法，所以无需额外的软件就能处理 JavaScript 中的 JSON。

通过 JavaScript，您可以创建一个对象数组，并像这样进行赋值：

**例子**

var employees = [

{ "firstName":"Bill" , "lastName":"Gates" },

{ "firstName":"George" , "lastName":"Bush" },

{ "firstName":"Thomas" , "lastName": "Carter" }

];

可以像这样访问 JavaScript 对象数组中的第一项：

employees[0].lastName;

返回的内容是：

Gates

可以像这样修改数据：

employees[0].lastName = "Jobs";

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=json_objectarray)

在下面的章节，您将学到如何把 JSON 文本转换为 JavaScript 对象。

**JSON 文件**

* JSON 文件的文件类型是 ".json"
* JSON 文本的 MIME 类型是 "application/json"

JSON 使用

* [JSON 语法](http://www.w3school.com.cn/json/json_syntax.asp)
* [JSON 教程](http://www.w3school.com.cn/json/index.asp)

**把 JSON 文本转换为 JavaScript 对象**

JSON 最常见的用法之一，是从 web 服务器上读取 JSON 数据（作为文件或作为 HttpRequest），将 JSON 数据转换为 JavaScript 对象，然后在网页中使用该数据。

为了更简单地为您讲解，我们使用字符串作为输入进行演示（而不是文件）。

**JSON 实例 - 来自字符串的对象**

创建包含 JSON 语法的 JavaScript 字符串：

var txt = '{ "employees" : [' +

'{ "firstName":"Bill" , "lastName":"Gates" },' +

'{ "firstName":"George" , "lastName":"Bush" },' +

'{ "firstName":"Thomas" , "lastName":"Carter" } ]}';

由于 JSON 语法是 JavaScript 语法的子集，JavaScript 函数 eval() 可用于将 JSON 文本转换为 JavaScript 对象。

eval() 函数使用的是 JavaScript 编译器，可解析 JSON 文本，然后生成 JavaScript 对象。必须把文本包围在括号中，这样才能避免语法错误：

var obj = eval ("(" + txt + ")");

在网页中使用 JavaScript 对象：

**例子**

<p>

First Name: <span id="fname"></span><br />

Last Name: <span id="lname"></span><br />

</p>

<script type="text/javascript">

document.getElementById("fname").innerHTML = obj.employees[1].firstName

document.getElementById("lname").innerHTML = obj.employees[1].lastName

</script>

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=json_eval)

**JSON 解析器**

**提示：**eval() 函数可编译并执行任何 JavaScript 代码。这隐藏了一个潜在的安全问题。

使用 JSON 解析器将 JSON 转换为 JavaScript 对象是更安全的做法。JSON 解析器只能识别 JSON 文本，而不会编译脚本。

在浏览器中，这提供了原生的 JSON 支持，而且 JSON 解析器的速度更快。

较新的浏览器和最新的 ECMAScript (JavaScript) 标准中均包含了原生的对 JSON 的支持。

|  |  |
| --- | --- |
| **Web 浏览器支持** | **Web 软件支持** |
| * Firefox (Mozilla) 3.5 * Internet Explorer 8 * Chrome * Opera 10 * Safari 4 | * jQuery * Yahoo UI * Prototype * Dojo * ECMAScript 1.5 |

[亲自试一试](http://www.w3school.com.cn/tiy/t.asp?f=json_parse)

对于较老的浏览器，可使用 JavaScript 库：<https://github.com/douglascrockford/JSON-js>

JSON 格式最初是[由 Douglas Crockford 制定的](http://developer.yahoo.com/yui/theater/video.php?v=crockford-json)。