JSON是一种数据格式，并不从属于JavaScript，很多编程语言都有针对JSON的解析器和序列化器。

JSON实在JavaScirpt中读写结构化数据的更好的方式，因为可以把JSON直接传给eval(),饿而且不必创建DOM对象。

JSON的语法可以表示以下三种类型的值：

简单值：字符串（必须使用双引号，单引号会导致语法错误）、数值、布尔值和null，但是不支持undefined。

对象：一种复杂数据类型。表示一组无序的键值对，而每个键值对中的值可以是简单值，也可以是复杂数据类型的值。

数组：一种复杂数据类型。表示结构化数据的格式。

JSON不支持变量、函数或对象实例，它就是一种表示结构化数据的格式。

JSON中的对象和JavaScirpt对象字面量的区别：  
JavaScirpt:  
var persion = {

“name”:”Nicholas”, //name和age外的引号可有可无

“age”:29

};

JSON表示上述对象的方式：

{

“name”:”Nicholas”,

“age”:29

}

区别：1.没有声明变量（JSON没有变量的概念）

2.没有末尾的分号；

3.对象的属性名必须加双引号

JSON中属性的值可以是简单值，也可以是复杂类型值。

{

“name”:”Nicholas”,

“age”:29

“school”:{

“name”:”XXcollege”,

“location”:”SH”

}

}

这个例子中出现了两个”name”属性，但是因为他们分属于不同的对象，所以这样完全没问题。

但是！同一个对象中绝对不允许出现两个同名属性！

JSON中的数组

采用的是JavaScirpt中的数组字面量形式，

JavaScirpt:

Var values = [25, “hi”, true]

JSON

[25, “hi”, true]

注意：JSON数组也没有变量和分号，可以把数组和对象结合起来，构成更加复杂的数据集合：  
[

{

“title”:”professional JavaScript”,

“authors”:[

“Nicholas C. Zakas”

],

Edition: 3, //这里为何Edition和Year不用加””百思不得其解。。。可能因为值是数字，所以不用?!

Year:2011

},

{

“title”:”professional Ajax”,

“authors”:[

“Nicholas C. Zakas”,

“Jeremy McPeak”,

“Joe Fawcett”

],

Edition: 2,

Year:2008

}

]

如果我想取得第二本书的书名，只需输入：（默认已经把解析JSON数据结构后得到的对象保存到了变量books中）

Books[2].title

如果用DOM操作会如何：

document.getElementsByTagName(“book”[2]).getAttribute(“title”)

没有对比就没有伤害，DOM实在是太繁琐了。

从此以后，JSON就成了Web服务开发中交换数据的事实标准。

早期的JSON解析器基本上就是使用JavaScript的eval()函数，由于JSON是JavaScirpt语法的子集，因此eval()函数可以解析，解释病房会JavaScirpt对象和数组。

ECMAScirpt5对解析JSON的行为进行规范，定义了全局对象JSON，支持者对象的浏览器有IE8+,Firefox3.5+,Safari4+,Chrome和Opera10.5+,对于较早版本的浏览器，可以用一个shim:http://github.com/douglascrockford/JSON-js，在旧版本的浏览器中，使用eval()对JSON数据结构求值存在风险，因为可能会执行一些恶意代码，Shim是最佳选择。

JSON对象有两个方法：stringify()和parse()，分别用于吧JavaScirpt对象徐刘华为JSON字符串和把JSON字符串解析为原生JavaScript值

例子：

使用JSON.stringify()把一个JavaScirpt对象序列化为一个JSON字符串，然后存进jsonText

var book = {

“title”:”professional JavaScript”,

“authors”:[

“Nicholas C. Zakas”

],

Edition: 3,

Year:2011

};

var jsonText = JSON.stringify(book);

默认情况下，JSON.stringify()输出的JSON字符串不包括任何空格字符或者缩进，因此保存在jsonText中的字符串如下所示：

{“title”:”professional JavaScript”,“authors”:[“Nicholas C. Zakas”],Edition: 3,Year:2011;}

在序列化JavaScript对象时，所有函数及原型成员都会被有意忽略，不体现在结果中。此外，值为undefined的任何属性也会被跳过，结果中最终都是值为有效JSON数据类型的实力属性。

例子：

将JSON字符串直接传递给JSON.parse()就可以得到相应的JavaScript值。创建一个与book类似的对象。

var bookCopy = JSON.parse(jsonText);

注意：虽然book与bookCopy具有相同的属性，但他们两个是独立的、没有任何关系的对象。

如果传给JSON.parse()的字符串不是有效的JSON,该方法会抛出错误。

序列化选项

JSON.stringify()除了要序列化的JavaScirpt对象外，还可以接收另外两个参数，这两个参数用于指定以不同的方式序列化JavaScript对象。第一个参数是一个过滤器，可以使一个数组，也可以是一个函数；第二个参数是一个选项，表示是否在JSON字符串中保留缩进。

过滤结果

如果过滤器参数是数组，那么JSON.stringify()的结果中将只包含数组中列出的属性。

例子：

Var book = {

“title”:”professional JavaScript”,

“authors”:[

“Nicholas C. Zakas”

],

Edition: 3,

Year:2011

};

var jsonText = JSON.stringify(book, [“title”,”edition”]);

返回结果：  
{“title”:”professional JavaScript”,”Edition”: 3}

如果第二个参数是函数，行为会稍有不同。传入的函数接受两个参数，属性（键）名和属性值。属性名只能是字符串，而在值并非键值对结构的值时，键名可以使空字符串：

注意：如果函数反悔了undefined，那么相应的属性会被忽略。

var book = {

“title”:”professional JavaScript”,

“authors”:[

“Nicholas C. Zakas”

],

Edition: 3,

Year:2011

};

var jsonText = JSON.stringify(book, function(key, value){

Switch(key){

case “authors”:

return value,join(“,”);

case “year”:

return 5000;

case “edition”:

return undefined;

Default:

return value;

}

});

这里，函数过滤器根据传入的键来决定结果。如果键为”authors”,就将数组连接为一个字符串；

如果键为year,则将其值设置为5000；如果键为”edition”，通过返回undefined删除该属性。

最后，一定要提供default项，此时返回传入的键，以便其他值都能正常的出现在结果中。

实际上，第一次调用这个函数过滤器，传入的键是一个空字符串，而值就是book对象。序列化后的JSON字符串如下所示：

{“title”:”professional JavaScript”,”author”:”Nicholas C. Zakas”,”year”:5000}

字符串缩进：

第三个参数控制结果中的缩进和空白符，如果这个参数是一个数值，那它表示的是每个级别缩进的空格数。例如，要在每个级别缩进四个空格， 可以这样写代码：

Var book = {

“title”:”professional JavaScript”,

“authors”:[

“Nicholas C. Zakas”

],

Edition: 3,

Year:2011

};

var jsonText = JSON.stringify(book, null, 4);

保存在jsonText中的字符串如下所示：

{

“title”:”professional JavaScript”,

“authors”:[

“Nicholas C. Zakas”

],

“Edition”: 3,

“Year”:2011

}

只要传入的是有效的空格缩进的参数值，结果字符串就会包含换行符，最大缩进格数为10，所有大于10的值都会自动转换为10.

如果缩进参数是一个字符串而不是数值，则这个字符串将在JSON字符串中被用作缩进字符（不再使用空格）。

例子：

var jsonText = JSON.stringify(book,null,” - -”);

{

- -“title”:”professional JavaScript”,

- -“authors”:[

- -“Nicholas C. Zakas”

- -],

- -“Edition”: 3,

- -“Year”:2011

}

缩进字符最长不能超过10个字符长，如果字符长度超过了10个，结果中将只出现前10个字符。