

## JSON.stringify()

JSON 通常用于与服务端交换数据。

在向服务器发送数据时一般是字符串。

我们可以使用 `JSON.stringify()` 方法将 JavaScript 对象转换为字符串。

语法

`JSON.stringify(value[, replacer[, space]])`

参数说明：

**value:**

必需， 一个有效的 JSON 对象。

**replacer:**

可选。用于转换结果的函数或数组。

如果 `replacer` 为函数，则 `JSON.stringify` 将调用该函数，并传入每个成员的键和值。使用返回值而不是原始值。如果此函数返回 `undefined`，则排除成员。根对象的键是一个空字符串：""。

如果 `replacer` 是一个数组，则仅转换该数组中具有键值的成员。成员的转换顺序与键在数组中的顺序一样。当 `value` 参数也为数组时，将忽略 `replacer` 数组。

space:

可选，文本添加缩进、空格和换行符，如果 space 是一个数字，则返回值文本在每个级别缩进指定数目的空格，如果 space 大于 10，则文本缩进 10 个空格。space 有可以使用非数字，如：\t。

### JavaScript 对象转换

例如我们向服务器发送以下数据：

```
var obj = { "name":"runoob", "alexa":10000, "site":"www.runoob.com"};
```

我们使用 JSON.stringify() 方法处理以上数据，将其转换为字符串：

```
var myJSON = JSON.stringify(obj);
```

myJSON 为字符串。

可以将 myJSON 发送到服务器：

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>菜鸟教程(runoob.com)</title>
```

```
</head>
```

```
<body>
```

<h2>将 JavaScript 对象转换为 JSON 字符串</h2>

<p id="demo"></p>

<script>

```
var obj = { "name":"runoob", "alexa":10000,  
"site":"www.runoob.com"};  
  
var myJSON = JSON.stringify(obj);  
  
document.getElementById("demo").innerHTML = myJSON;
```

</script>

</body>

</html>

JavaScript 数组转换

我们也可以将 JavaScript 数组转换为 JSON 字符串：

实例

```
var arr = [ "Google", "Runoob", "Taobao", "Facebook" ]; var  
myJSON = JSON.stringify(arr);
```

myJSON 为字符串。

可以将 myJSON 发送到服务器：

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>菜鸟教程(runoob.com)</title>
```

```
</head>
```

```
<body>
```

```
<h2>将 JavaScript 数组转换为 JSON 对象</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var arr = [ "Google", "Runoob", "Taobao", "Facebook" ];
```

```
var myJSON = JSON.stringify(arr);
```

```
document.getElementById("demo").innerHTML = myJSON;
```

```
</script>
```

</body>

</html>

异常

解析数据

JSON 不能存储 Date 对象。

JSON.stringify() 会将所有日期转换为字符串。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<h2>JSON.stringify 将 Date 对象转换为字符串</h2>

<p id="demo"></p>

<script>

```
var obj = { "name": "Runoob", "initDate": new Date(), "site":  
"www.runoob.com"};
```

```
var myJSON = JSON.stringify(obj);  
document.getElementById("demo").innerHTML = myJSON;
```

```
</script>
```

```
</body>
```

```
</html>
```

之后可以再将字符串转换为 **Date** 对象。

解析函数

**JSON** 不允许包含函数, **JSON.stringify()** 会删除 **JavaScript** 对象的函数, 包括 **key** 和 **value**。

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>菜鸟教程(runoob.com)</title>
```

```
</head>
```

```
<body>
```

```
<h2>JSON.stringify 将删除对象中的函数</h2>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var obj = { "name":"Runoob", "alexa":function () {return 1000;}, "site":"www.runoob.com"};
```

```
var myJSON = JSON.stringify(obj);
```

```
document.getElementById("demo").innerHTML = myJSON;
```

```
</script>
```

```
</body>
```

```
</html>
```

可以在执行 `JSON.stringify()` 函数前将函数转换为字符串来避免  
以上问题的发生：

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>菜鸟教程(runoob.com)</title>
```

```
</head>
```

```
<body>
```

## <h2>JSON.stringify 将移除对象中的函数</h2>

<p>我们可以在执行 JSON.stringify() 函数前将函数转换为字符串来保留函数。</p>

<p id="demo"></p>

<script>

```
var obj = { "name":"Runoob", "alex":function () {return 1000;}, "site":"www.runoob.com"};
obj.alex = obj.alex.toString();
var myJSON = JSON.stringify(obj);
document.getElementById("demo").innerHTML = myJSON;
```

</script>

</body>

</html>