

JSON.parse()

JSON 通常用于与服务端交换数据。

在接收服务器数据时一般是字符串。

可以使用 JSON.parse() 方法将数据转换为 JavaScript 对象。

语法

JSON.parse(text[, reviver])

参数说明：

text:必需， 一个有效的 JSON 字符串。

reviver: 可选，一个转换结果的函数， 将为对象的每个成员调用此函数。

JSON 解析实例

例如从服务器接收了以下数据：

```
{ "name":"runoob", "alexa":10000, "site":"www.runoob.com" }
```

使用 JSON.parse() 方法处理以上数据，将其转换为 JavaScript 对象：

```
var obj = JSON.parse('{ "name":"runoob", "alexa":10000, "site":"www.runoob.com" }');
```

解析前要确保数据是标准的 JSON 格式，否则会解析出错。

你可以使用我们的在线工具检测：<https://c.runoob.com/front-end/53>。

解析完成后，我们就可以在网页上使用 JSON 数据了：

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<h2>从 JSON 对象中创建 JavaScript 对象</h2>
```

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

```
<script>
```

```
var obj = JSON.parse('{ "name": "runoob", "alex": 10000,  
"site": "www.runoob.com" }');
```

```
document.getElementById("demo").innerHTML = obj.name + ": " +  
obj.site;
```

```
</script>
```

```
</body>
```

```
</html>
```

从服务端接收 JSON 数据

可以使用 AJAX 从服务器请求 JSON 数据，并解析为 JavaScript 对象。

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<h2>使用 XMLHttpRequest 来获取文件内容</h2>
```

```
<p>文件内容是标准的 JSON 格式，可以使用 JSON.parse 方法将其转换为 JavaScript 对象。</p>
```

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

```
<script>
```

```
var xmlhttp = new XMLHttpRequest();
```

```
xmlhttp.onreadystatechange = function() {
```

```
        if (this.readyState == 4 && this.status == 200) {  
            myObj = JSON.parse(this.responseText);  
            document.getElementById("demo").innerHTML =  
myObj.name;  
        }  
    };  
xmlhttp.open("GET", "/try/ajax/json_demo.txt", true);  
xmlhttp.send();
```

```
</script>
```

```
<p>查看 JSON 文件数据 <a href="/try/ajax/json_demo.txt"  
target="_blank">json_demo.txt</a></p>
```

```
</body>
```

```
</html>
```

从服务端接收数组的 JSON 数据

如果从服务端接收的是数组的 JSON 数据, 则 `JSON.parse` 会将其转换为 JavaScript 数组:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<h2>内容是数组</h2>
```

```
<p>内容是数组会转换为 JavaScript 数组。</p>
```

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

```
<script>
```

```
var xmlhttp = new XMLHttpRequest();
```

```
xmlhttp.onreadystatechange = function() {
```

```
    if (this.readyState == 4 && this.status == 200) {
```

```
        myArr = JSON.parse(this.responseText);
```

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

```
myArr[1];
```

```
    }
```

```
};
```

```
xmlhttp.open("GET", "/try/ajax/json_demo_array.txt", true);  
xmlhttp.send();
```

```
</script>
```

```
<p>查看服务端数据 <a href="/try/ajax/json_demo_array.txt"  
target="_blank">json_demo_array.txt</a></p>
```

```
</body>
```

```
</html>
```

异常

解析数据

JSON 不能存储 Date 对象。

如果你需要存储 Date 对象，需要将其转换为字符串。

之后再字符串转换为 Date 对象。

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

<body>

<h2>将字符串转换为 Date 对象。</h2>

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

<script>

```
var text = '{ "name":"Runoob", "initDate":"2013-12-14", "site":"www.runoob.com"}';
```

```
var obj = JSON.parse(text);
```

```
obj.initDate = new Date(obj.initDate);
```

```
document.getElementById("demo").innerHTML = obj.name + "创建日期：" + obj.initDate;
```

</script>

</body>

</html>

可以启用 `JSON.parse` 的第二个参数 `reviver`，一个转换结果的函数，对象的每个成员调用此函数。

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<h2>字符串转换为 Date 对象</h2>
```

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

```
<script>
```

```
var text = '{ "name": "Runoob", "initDate": "2013-12-14",
```

```
"site": "www.runoob.com"}';
```

```
var obj = JSON.parse(text, function (key, value) {
```

```
    if (key == "initDate") {
```

```
        return new Date(value);
```

```
    } else {
```

```
        return value;
```



```
});
```

```
document.getElementById("demo").innerHTML = obj.name + "创建  
日期: " + obj.initDate;
```

```
</script>
```

```
</body>
```

```
</html>
```

解析函数

JSON 不允许包含函数，但你可以将函数作为字符串存储，之后再将字符串转换为函数。

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<h2>字符串转换为函数</h2>
```

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

```
<script>
```

```
var text = '{ "name": "Runoob", "alexa": "function () {return  
10000;}", "site": "www.runoob.com"}';
```

```
var obj = JSON.parse(text);
```

```
obj.alexa = eval("(" + obj.alexa + ")");
```

```
document.getElementById("demo").innerHTML = obj.name + "  
Alexa 排名: " + obj.alexa();
```

```
</script>
```

```
</body>
```

```
</html>
```