# 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。

我们可以使用 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

例如我们从服务器接收了以下数据：

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

我们使用 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" }');

解析完成后，我们就可以在网页上使用 JSON 数据了：

## 实例

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

<script>

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

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

</script>

## 从服务端接收 JSON 数据

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

## 实例

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();

## 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：

## 实例

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();

## 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。

之后再将字符串转换为 Date 对象。

## 实例

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;

我们可以启用 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。

## 实例

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

var obj = JSON.parse(text,

function (key, value) { //在生成值之前调用函数，取函数的返回值生成值

if (key == "initDate") {

return new Date(value);

}

else {

return value;

}

});

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

## 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。

## 实例

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();

eval(string):函数可计算某个字符串，并执行其中的的 JavaScript 代码。

eval("var a=1"); // 声明一个变量a并赋值1。eval("2+3"); // 执行加运算，并返回运算值。eval("mytest()"); // 执行mytest()函数。eval("{b:2}"); // 声明一个对象。