

JSON 对象

对象语法

实例

```
{ "name": "runoob", "alexa": 10000, "site": null }
```

JSON 对象使用在大括号({})中书写。

对象可以包含多个 key/value (键/值) 对。

key 必须是字符串, value 可以是合法的 JSON 数据类型 (字符串, 数字, 对象, 数组, 布尔值或 null)。

key 和 value 中使用冒号(:)分割。

每个 key/value 对使用逗号(,)分割。

访问对象值

你可以使用点号 (.) 来访问对象的值:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

<p>可以使用点号 (.) 来访问 JSON 对象的值: </p>

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

<script>

```
var myObj, x;
```

```
myObj = { "name": "runoob", "alexa": 10000, "site": null };
```

```
x = myObj.name;
```

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

</script>

可以使用中括号（[]）来访问对象的值：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<p>你也可以使用中括号（[]）来访问 JSON 对象的值：</p>

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

```
<script>
```

```
var myObj, x;
```

```
myObj = myObj = { "name": "runoob", "alexa": 10000,  
"site": null };
```

```
x = myObj["name"];
```

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

```
</script>
```

```
</body>
```

```
</html>
```

循环对象

可以使用 for-in 来循环对象的属性：

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

<body>

<p>使用 for-in 来循环对象的属性:</p>

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

<script>

```
var myObj = { "name":"runoob", "alex":10000, "site":null  
};
```

```
for (x in myObj) {
```

```
    document.getElementById("demo").innerHTML += x + "<br>  
";
```

```
}
```

</script>

</body>

</html>

在 for-in 循环对象的属性时，使用中括号（[]）来访问属性的值：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

```
</head>
```

```
<body>
```

<p>在 for-in 循环对象的属性时，使用中括号（[]）来访问属性的值：</p>

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

```
<script>
```

```
var myObj = { "name": "runoob", "alexa": 10000, "site": null };
```

```
for (x in myObj) {
```

```
    document.getElementById("demo").innerHTML += myObj[x] +  
    "<br>";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```

嵌套 JSON 对象

JSON 对象中可以包含另外一个 JSON 对象：

实例

```
myObj = { "name": "runoob", "alexa": 10000, "sites": { "site1": "www.runoob.com", "site2": "m.runoob.com", "site3": "c.runoob.com" } }
```

你可以使用点号(.)或者中括号([])来访问嵌套的 JSON 对象。

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<p>访问 JSON 内嵌对象。</p>
```

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

```
<script>
```

```
myObj = {  
    "name": "runoob",  
    "alexa": 10000,
```

```
    "sites": {  
        "site1": "www.runoob.com",  
        "site2": "m.runoob.com",  
        "site3": "c.runoob.com"  
    }  
}  
  
document.getElementById("demo").innerHTML +=  
myObj.sites.site1 + "<br>";  
  
// 或者  
  
document.getElementById("demo").innerHTML +=  
myObj.sites["site1"];  
  
</script>
```

</body>

</html>

修改值

你可以使用点号(.)来修改 JSON 对象的值:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

```
</head>
```

```
<body>
```

```
<p>修改 JSON 对象的值。</p>
```

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

```
<script>
```

```
var myObj, i, x = "";
```

```
myObj = {
```

```
    "name": "runoob",
```

```
    "alexa": 10000,
```

```
    "sites": {
```

```
        "site1": "www.runoob.com",
```

```
        "site2": "m.runoob.com",
```

```
        "site3": "c.runoob.com"
```

```
    }
```

```
}
```

```
myObj.sites.site1 = "www.google.com";
```

```
for (i in myObj.sites) {
```



```
    x += myObj.sites[i] + "<br>";  
}
```

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

```
</script>
```

```
</body>
```

```
</html>
```

可以使用中括号([])来修改 JSON 对象的值:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<p>使用中括号([])来修改 JSON 对象的值。</p>
```

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

```
<script>

var myObj, i, x = "";

myObj = {

    "name": "runoob",

    "alex": 10000,

    "sites": {

        "site1": "www.runoob.com",

        "site2": "m.runoob.com",

        "site3": "c.runoob.com"

    }

}

myObj.sites["site1"] = "www.google.com";

for (i in myObj.sites) {

    x += myObj.sites[i] + "<br>";

}

document.getElementById("demo").innerHTML = x;

</script>
```

```
</body>
```

```
</html>
```

删除对象属性

可以使用 `delete` 关键字来删除 JSON 对象的属性：

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<p>删除 JSON 对象属性。</p>
```

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

```
<script>
```

```
var myObj, i, x = "";
```

```
myObj = {
```

```
    "name": "runoob",
```

```
    "alexa": 10000,
```

```
    "sites": {
```

```
        "site1": "www.runoob.com",  
        "site2": "m.runoob.com",  
        "site3": "c.runoob.com"  
    }  
}  
  
delete myObj.sites.site1;  
  
for (i in myObj.sites) {  
    x += myObj.sites[i] + "<br>";  
}  
  
document.getElementById("demo").innerHTML = x;
```

</script>

</body>

</html>

可以使用中括号([])来删除 JSON 对象的属性:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

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

```
</head>
```

```
<body>
```

```
<p>删除 JSON 对象属性。</p>
```

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

```
<script>
```

```
var myObj, i, x = "";
```

```
myObj = {
```

```
    "name": "runoob",
```

```
    "alex": 10000,
```

```
    "sites": {
```

```
        "site1": "www.runoob.com",
```

```
        "site2": "m.runoob.com",
```

```
        "site3": "c.runoob.com"
```

```
    }
```

```
}
```

```
delete myObj.sites["site1"];
```

```
for (i in myObj.sites) {
```

```
        x += myObj.sites[i] + "<br>";  
    }  

```

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

```
</script>
```

```
</body>
```

```
</html>
```