

# JSON 数组

## 数组作为 JSON 对象

### 实例

```
[ "Google", "Runoob", "Taobao" ]
```

JSON 数组在中括号中书写。

JSON 中数组值必须是合法的 JSON 数据类型（字符串, 数字, 对象, 数组, 布尔值或 null）。

JavaScript 中，数组值可以是以上的 JSON 数据类型，也可以是 JavaScript 的表达式，包括函数，日期，及 *undefined*。

## JSON 对象中的数组

对象属性的值可以是一个数组：

### 实例

```
{  
  "name": "网站",  
  "num": 3,  
  "sites": [ "Google", "Runoob", "Taobao" ]  
}
```

我们可以使用索引值来访问数组：

### 实例

```
x = myObj.sites[0];
```

[尝试一下 »](#)

## 循环数组

你可以使用 for-in 来访问数组：

### 实例

```
for (i in myObj.sites) {  
  x += myObj.sites[i] + "<br>";  
}
```

[尝试一下 »](#)

你也可以使用 for 循环：

### 实例

```
for (i = 0; i < myObj.sites.length; i++) {  
  x += myObj.sites[i] + "<br>";  
}
```

[尝试一下 »](#)

## 嵌套 JSON 对象中的数组

JSON 对象中数组可以包含另外一个数组，或者另外一个 JSON 对象：

### 实例

```
myObj = {  
  "name": "网站",  
  "num": 3,  
  "sites": [  
    { "name": "Google", "info": [ "Android", "Google 搜索", "Google 翻译" ] },  
    { "name": "Runoob", "info": [ "菜鸟教程", "菜鸟工具", "菜鸟微信" ] },  
    { "name": "Taobao", "info": [ "淘宝", "网购" ] }  
  ]  
}
```

我们可以使用 for-in 来循环访问每个数组：

### 实例

```
for (i in myObj.sites) {  
  x += "<h1>" + myObj.sites[i].name + "</h1>";  
  for (j in myObj.sites[i].info) {  
    x += myObj.sites[i].info[j] + "<br>";  
  }  
}
```

[尝试一下 »](#)

## 修改数组值

你可以使用索引值来修改数组值：

### 实例

```
myObj.sites[1] = "Github";
```

[尝试一下 »](#)

## 删除数组元素

我们可以使用 **delete** 关键字来删除数组元素：

### 实例

```
delete myObj.sites[1];
```

[尝试一下 »](#)[← JSON 对象](#)[JSON.parse\(\) →](#)

## 2 篇笔记

[写笔记](#)

json数据格式：主要由对象 { } 和数组 [ ] 组成:

其中对象包括键值对（属性:属性值）{key：value}，value 可为 str，num，list，obj。取值使用 objcet.key

{key: value, key2:value2，} 键：值用冒号分开，对间用，连接

数组包含元素：num，str，list，objcet 都可以，利用索引访问 [index]，用 . 连接各个值:

e.g：

```
var stu = {"student":          //stu 对象包含student的key,值为一个数组
[                               //数组的每一个值为一个具体的学生对象
{"name": "Tom","Grade":1, "age":11, "gender": "M"},      //学生对象的键为名字,值为对应属性
{"name": "Jerry", "Grade":1, "age":10, "gender": "M"}      //每个属性对应的是一个key,value
],
"classroom": {"class1": "room1", "class2": "room2"}      //对象的值,嵌套对象
};
```

读取数据:

```
document.write(stu.student[1].name);    // 输出第二个学生名
document.write(stu.student[0].age);     // 输出第一个学生年龄
document.write(stu.classroom.class1);   // 输出 classroom 的 class1 值
document.write(stu["classroom"].class2); // 也可用中括号键访问对象值
```

[尝试一下 »](#)

Tom 1年前 (2017-12-01)



删除数值的元素，数组的大小不变，代码如下：

```
var myObj, i, x = "";
myObj = {
  "name": "网站",
  "num": 3,
  "sites": [ "Google", "Runoob", "Taobao" ]
};
delete myObj.sites[1];

x = "sites.length = " + myObj.sites.length + "<br><br>";
```

```
for (i in myObj.sites) {  
    x += i + " " + myObj.sites[i] + "<br>";  
}  
  
document.getElementById("demo").innerHTML = x;
```

[尝试一下 »](#)

delete 运算符并不是彻底删除元素，而是删除它的值，但仍会保留空间。

运算符 delete 只是将该值置为 undefined，而不会影响数组长度，即将其变为稀疏数组（《JS权威指南》7.5节）。

更多内容可参考：[JS 中彻底删除 JSON 对象组成的数组中的元素](#)

喔喔和奶糖 4个月前 (11-05)