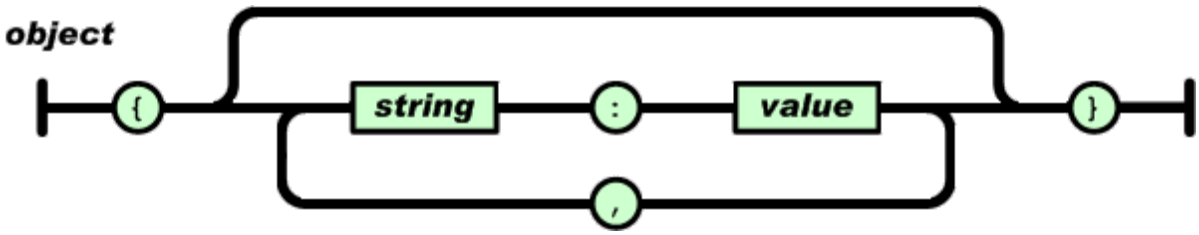
JSON（JavaScript Object Notation）是一种轻量型的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的 JS 规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

对象格式

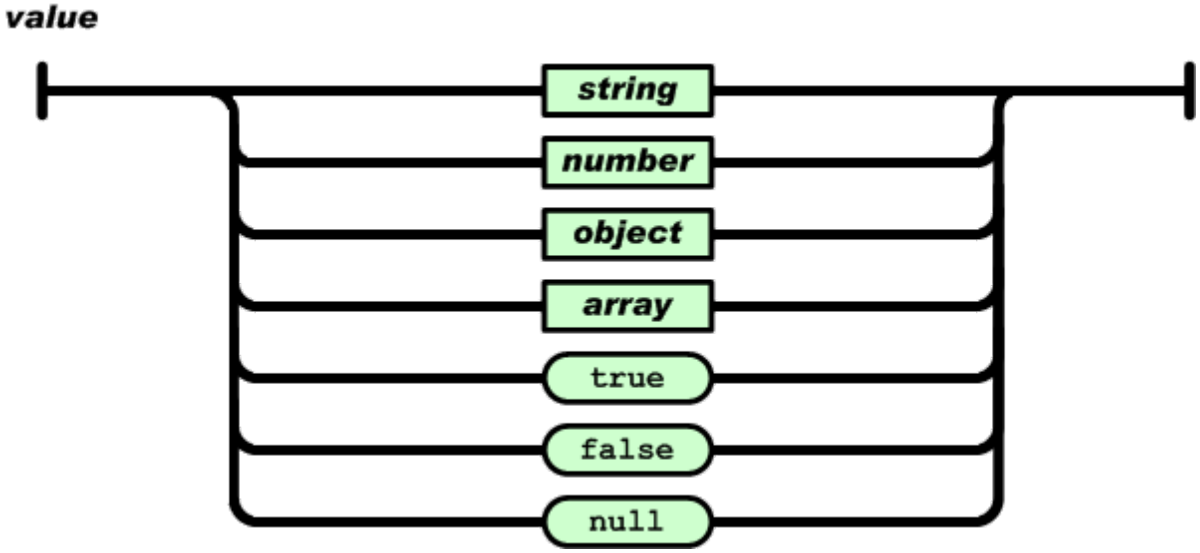
对象以左大括号（{）开始，以右大括号结束，对象中是一系列的name/value对，name和value以冒号（:）分隔，每一对name/value之间以逗号（,）分隔。如下图所示：



示例 1：对象格式的 JSON 数据（此处的代码没有加引号）

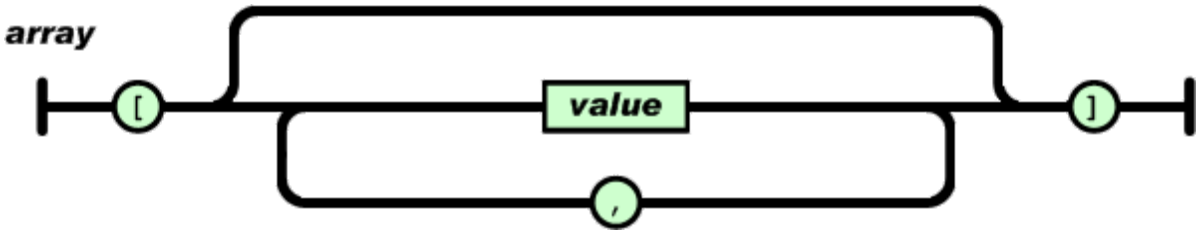
```
{
  "name": "张三",
  "age": 20,
  "gender": "男",
  "hobbies": ["跑步", "打球"]
}
```

对象的属性名必须使用双引包裹起来，属性值只能是下面 7 种类型的数据。



数组格式

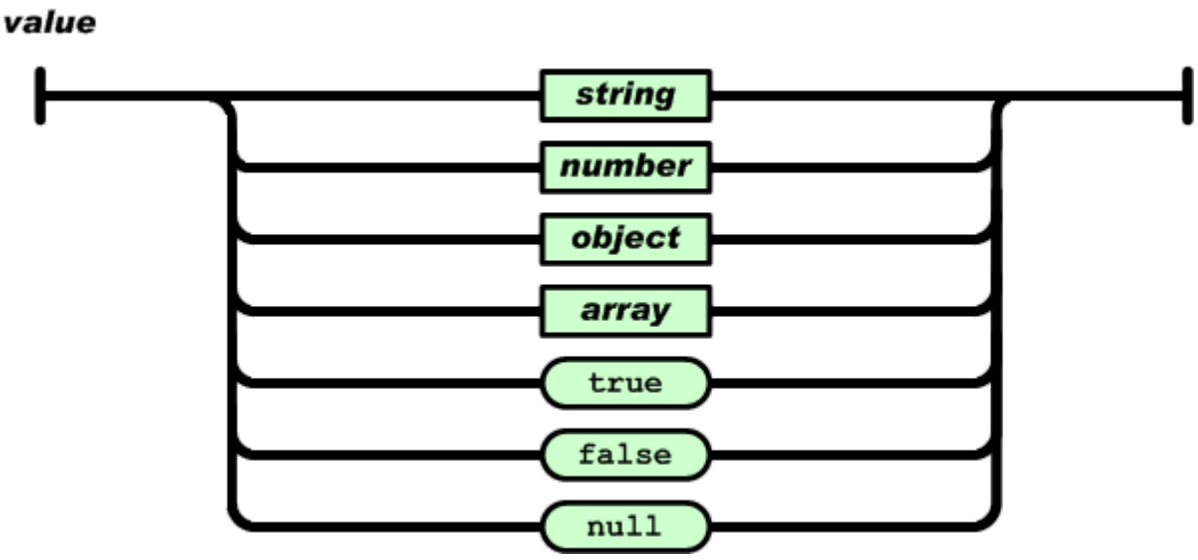
数组以左中括号（[）开始，以右中括号（]）结束，数组中是一系列有序的value值，value值之间以逗号（,）分隔。



示例 2：数组格式的 JSON 数据（此处的代码没有加引号）

```json
[
  {
    "name": "张三",
    "age": 20,
    "gender": "男",
    "hobbies": ["跑步", "打球"]
  },
  {
    "name": "李四",
    "age": 18,
    "gender": "女",
    "hobbies": ["唱歌", "跳舞"]
  }
]
```

数组中的成员也只能是下面 7 种类型的数据。



JSON数据与 JS 对象之间的转换

JS 提供一个 JSON 对象，专门用于处理 JSON 格式的数据。它是一个已经实例化好的本地对象，具有两个方法，parse() 用于将 JSON 格式的字符串转换成 Object 或 Array 对象；stringify() 方法正好相反，用于将 Object 或 Array 对象转换成 JSON 格式的字符串。

JSON数据转 JS 对象

使用 JSON.parse(string) 方法将JSON格式的字符串解析成一个JavaScript的对象（Object或Array）。

示例 3：将 JSON 格式的字符串转换成 Object 对象。

```
var str = '{"name":"张三","age":20}';
JSON.parse(str);  // {name: '张三', age: 20}
```

示例 4：将 JSON 格式的字符串转换成 Array 对象。

```
var str = '["red","green","blue"]';
JSON.parse(str);  // ['red', 'green', 'blue']
```

JS 对象转 JSON 数据

使用 JSON.stringify(Object/Array) 方法将JavaScript对象解析成JSON格式的字符串。

示例 5：将 Object 对象转换成 JSON 格式的字符串

```
var obj = {name: '张三', age: 20};

JSON.stringify(obj);  // '{"name":"张三","age":20}'
```

示例 6：将数组对象转换成 JSON 格式的字符串

```
var arr = ['red', 'green', 'blue'];
JSON.stringifty(arr);  // '["red","green","blue"]'
```

对象的属性值和数组中的成员必须符合 JSON 对 Value 的取值要求，否则会出现如下两种情况。

如果对象中的某个属性的值不合符要求，则该属性会被忽略。

示例 7：obj 对象中的 run 方法会被忽略

```
var obj = {name: '张三', age: 20, run: function ()};
JSON.stringify(obj);  // '{"name":"张三","age":20}'
```

如果数组中的某个成员不符合要求，则该成员会被转换成 null。

示例 8：arr 对象中的成员 undefined 会被转换成 null

```
var arr = ['red', 'green', undefined];
JSON.stringify(arr);  // '["red","green",null]'
```

如果 JSON 数据比较多，可以创建一个以 json 为扩展名的文件，专门用于保存 JSON 格式的数据。

示例 9：创建 example.json 文件，然后在文件中写入 JSON 格式的数据。

```
[
  {
    "name": "张三",
    "age": 20,
    "gender": "男",
    "hobbies": ["跑步", "打球"]
  },
  {
    "name": "李四",
    "age": 18,
    "gender": "女",
    "hobbies": ["唱歌", "跳舞"]
  }
]
```

注意：json 文件中的数据不需要使用引号包裹。