## 一、JSON数据类型

JSON格式支持以下数据类型：

|  |  |
| --- | --- |
| 类型 | 描述 |
| 数字型（Number） | JavaScript中的双精度浮点型格式 |
| 字符串型（String） | 双引号包裹的Unicode字符和反斜杠转义字符 |
| 布尔型（Boolean） | true、false |
| 数组（Array） | 有序的值序列 |
| 值（Value） | 可以是字符串，数字，true或false |
| 对象（Object） | 无序的键：值对集合 |
| 空格（Whitespace） | 可用于任意符号对之间 |
| null | 空 |

### 1.数字型

JavaScript中的双精度浮点型格式，取决于实现。

不能使用八进制和十六进制格式。

在数字中不能使用NaN和Infinity。

整形（Integer）：数字1-9,0和正负数

分数（Fraction）：分数，比如.3，.9

指数（Exponent）：指数，比如e，e+，e-­，E，E+，E-

var obj = {mark: 97}

### 2.字符串型

零个或多个双引号包裹的Unicode字符以及反斜杠转义序列。

字符就是只有一个字符的字符串，长度为1。

|  |  |
| --- | --- |
| 类型 | 描述 |
| “ | 双引号 |
| \ | 反斜线 |
| / | 斜线 |
| b | 退格符 |
| f | 换页符 |
| n | 换行符 |
| r | 回车符 |
| t | 水平制表符 |
| u | 四位十六进制数字 |

var obj = {name: “Chance”}

### 3.布尔型

var = {name: “Chance”, marks: 97,distinction: true}

### 4,数组

一个有序的值集合；

使用方括号闭合；

值使用逗号分隔；

数组索引可以以0或1开始；

当键名是连续的整数时应该使用数组。

{

“books”: [

{“language”: “Java”, “edition”: ”second”},

{“language”: “C++”, “lastName”: “fifth”},

{“language”: “C”, “lastName”: “third”}

]

}

### 5.值

包括：

数字（整型和浮点数）

字符串

布尔值

数组

对象

null

var i = 1;

var j = “sachin”;

var k = null;

### 6.对象

一个无序的名/值对集合；

对象使用大括号闭合，以{开始，以}结尾；

每个名称后面都跟随一个冒号，名/值对使用逗号分割；

键名必须是字符串，并且不能同名；

当键名是任意字符串时应该使用对象。

{

“id”: “001A”,

“language”: “Java”,

“price”: 500,

}

### 7.空格

可以在任意一对符号之间插入。可以添加用来让代码更可读。

var i = “ sachin”;

var j = “ saurav”

### 8.null

空类型

var i = null;

## 二、JSON对象

{“name”: “Chance”, “alexa”: 8000, “site”: null}

JSON对象使用在大括号中书写，对象可以包含多个键/值对。

key必须是字符串，值可以是合法的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；

每个键/值对使用逗号分割。

### 创建简单的对象

//创建一个空对象

var jsonObj = {};

//创建一个新对象

var jsonObj = new Object();

### 访问对象的值

//创建一个bookname属性值为字符串，price属性值为数字的对象。可以通过对象.属性访问对象属性。

var jsonObj = {“bookname”:”VB BLACK BOOK”, “price”: 500};

//也可以使用对象[“属性”]访问对象属性

var jsonObj, x;

jsonObj = {“name”:”Chance”, “alexa”: 8000, “site”: null};

x = jsonObj[“name”];

### 循环对象

使用for-in来循环对象的属性：

## 三、JSON模式

## 四、JSON与XML对比