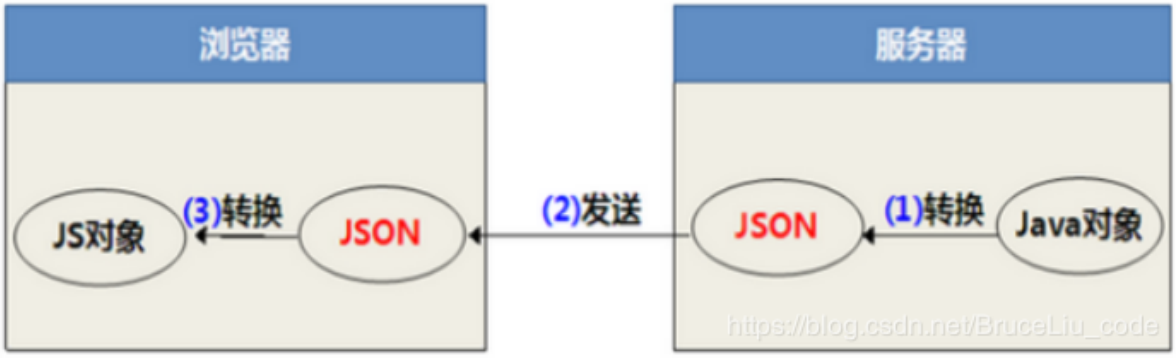
删除水印 万兴PDF

格式是：在JS对象、数组、字符串、数值、逻辑值、null外面套了一对单引号，单引号内部如果要用到引号，就必须是双引号.

例如：单独引号内部是js对象，那么对象中的键必须要用双引号引起来、单引号内部是js字符串，那么这个字符串必须要用双引号引起来,示例---->'{ "name":"lxd","age":100 }' 、 "lxdcshhy" '

1.JSON概述

JSON(JavaScript Object Notation, JS 对象标记) 是一种轻量级的数据交换格式。它基于 ECMAScript (w3c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。



Js中的数组和对象？ 数组：[1,2,3] 对象：{name:"张三", age: 20}

复杂数组：[{name:"张三", age: 20}, {name:"李四", age: 30}, 1] 复杂对象：{total:50, data: [{name:"张三", age: 20}, {name:"李四", age: 30}]}

Json本质：特殊格式的字符串，特殊在字符串的结构，和js中的数组和对象一样一样的！ 数组：[1,2,3] 数组格式的json串："[1,2,3]" 对象：{name: "张", age: 20} 对象格式的json串：'{name: "张", age: 20}' 数组中放对象：[{id:"11",name:"北京"}, {id:"12",name:"天津"}] 数组中放对象的json格式：'[{id:"11",name:"北京"}, {id:"12",name:"天津"}]'

JS代码

```
<script type="text/javascript">

    function show(){
        //1.JS中数组
        var girls=["王昭君","貂蝉","西施","凤姐"];

        //2.JS中对象
        var p={name:"八戒",age:80,address:"高老庄",phone:"1338809090"};


        //3.JS对象数组
        var persons=[
                {name:"八戒",age:80,address:"高老庄",phone:"1338809888"},
                {name:"悟空",age:56,address:"高老庄1",phone:"1338809090"},
```

删除水印 万兴PDF

```
                {name:"沙僧",age:81,address:"高老庄2",phone:"1338809666"},
                {name:"白骨精",age:79,address:"高老庄3",phone:"1338809888"}
            ];

            //alert(persons.length);
            for(var i=0;i<persons.length;i++){
                var p=persons[i];
                //alert(p.name+"==="+p.age+"==="+p.address);
            }

            //4.复杂版的JS对象
            var citys={
                    "name": "中国",
                    "province": [{
                        "name": "黑龙江",
                        "cities": {
                            "city": ["哈尔滨", "大庆"]
                        }
                    }, {
                        "name": "广东",
                        "cities": {
                            "city": ["广州", "深圳", "珠海"]
                        }
                    }, {
                        "name": "台湾",
                        "cities": {
                            "city": ["台北", "高雄"]
                        }
                    }, {
                        "name": "新疆",
                        "cities": {
                            "city": ["乌鲁木齐"]
                        }
                    }]
                };

            alert(citys.province[2].cities.city[0]);
        }
    </script>
```

2.为什么需要Json

删除水印 万兴PDF

ajax就是通过js代码，完成和后台数据的交互，但是，往往后台响应的结果的数据格式都比较复杂，例如List、Map等，这些数据js是不能够识别的，因为js和java是不同的语言！因此需要一种通用的数据格式，即一种通用的“语言”！如xml、json，这两个本质还是字符串！任何语言都支持字符串！

> json是为了完成前台和后台的复杂数据交换的中间桥梁！

Java：List ===> json ===> 响应给ajax ===> 把json转js对象或数组 ===> 通过DOM技术操作页面元素

json和js对象的区别：唯一的区别是json是字符串，不可以直接调用字符串的属性来获取json中的“属性”对应的值，如果想获取到json字符串中的属性的值，需要先将json转换为js对象。

后台的List、Map、User、数组等对象可以转换为json格式的字符串，然后使用js将json字符串转换为js对象，然后操作这些数据，最终达到无刷新更新页面中的部分内容。

3.JSON与Js对象之间的相互转换

3.1.JSON转js对象

删除水印 万兴PDF

3.1.1.使用js中的内建函数eval转换

JSON 实例 - 来自字符串的对象

创建包含 JSON 语法的 JavaScript 字符串：

```
var txt = '{ "employees" : [' +
'{ "firstName":"Bill" , "lastName":"Gates" },' +
'{ "firstName":"George" , "lastName":"Bush" },' +
'{ "firstName":"Thomas" , "lastName":"Carter" } ]}';
```

由于 JSON 语法是 JavaScript 语法的子集，JavaScript 函数 eval() 可用于将 JSON 文本转换为 JavaScript 对象。

eval() 函数使用的是 JavaScript 编译器，可解析 JSON 文本，然后生成 JavaScript 对象。必须把文本包围在括号中，这样才能避免语法错误：

```
var obj = eval ("(" + txt + ")");
```

https://blog.csdn.net/BruceLiu_code

```
> var txt = '{ "employees" : [' +
  '{ "firstName":"Bill" , "lastName":"Gates" },' +
  '{ "firstName":"George" , "lastName":"Bush" },' +
  '{ "firstName":"Thomas" , "lastName":"Carter" } ]}';
< undefined
> var employees = eval("(" + txt + ")");
< undefined
> txt.employees
< undefined
> employees.employees
< [▼Object i , ▼Object i , ▼Object i ]
    firstName: "Bill"      firstName: "George"      firstName: "Thomas"
    lastName: "Gates"      lastName: "Bush"         lastName: "Carter"
  ▶ __proto__: Object    ▶ __proto__: Object      ▶ __proto__: Object
```

万兴PDF https://blog.csdn.net/BruceLiu_code

eval函数，不仅可以将json格式的字符串，转换为js对象，还可以用于一些计算，和js代码的动态拼接

```
> eval("1+2")
< 3
> eval("2" + "*" + "3")
< 6
> eval("alert(1)")
```

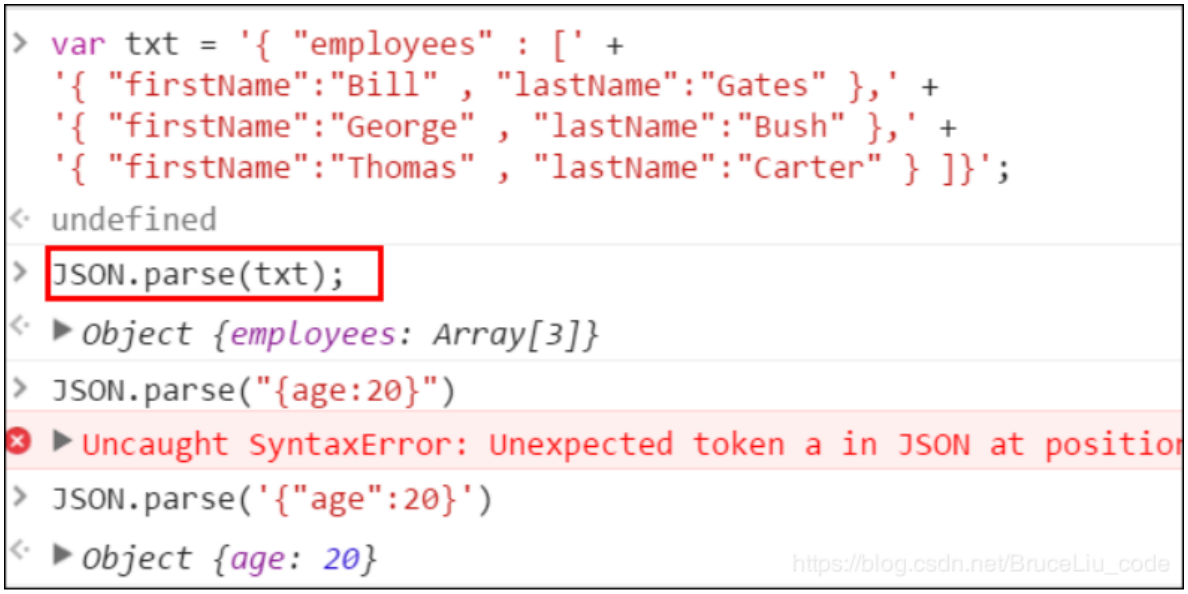
由于eval函数可以解析js代码，并执行，因此存在风险。

3.1.2.使用JSON对象将json字符串转换成js对象

> var txt = "[1,23]"; var arr = JSON.parse(txt); // 将json串转换为js对象 var txt = JSON.stringify(arr); // 将js对象转换为json串，调试的时候，方便查看对象的结构

删除水印 万兴PDF

JSON对象：浏览器自带的一个js对象，对象中有一些方法可供使用



使用JSON对象中的parse方法可以将json字符串转换成js对象，但是JSON对象对json字符串的格式要求比较严格：key值一定要使用双引号引起来，字符串类型的值也必须要使用双引号

但是，IE9以下不支持JSON对象，需要引入json2.js，使JSON对象能够正常使用：

```
<!-- 只有ie浏览器才能够识别下面的html判断 html hack-->
<!--[if lt IE 9]>
    <script type="text/javascript" src="js/json2.js"></script>
<![endif]-->
<script type="text/javascript">
    var txt = '{ "employees" : [' +
    '{ "firstName":"Bill" , "lastName":"Gates" },' +
    '{ "firstName":"George" , "lastName":"Bush" },' +
    '{ "firstName":"Thomas" , "lastName":"Carter" } ]}';
    var obj = JSON.parse(txt);
    alert(obj.employees[0].firstName);
</script>
```

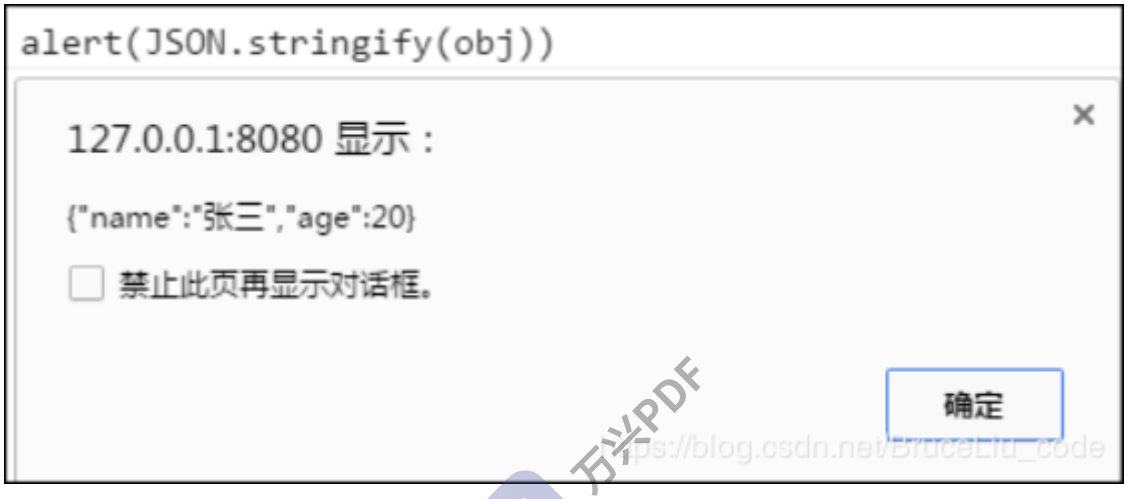
> json字符串==>js对象：JSON.parse("json串") js对象 ===> json字符串：JSON.stringify(js对象);

3.2.js对象转JSON

删除水印 万兴PDF

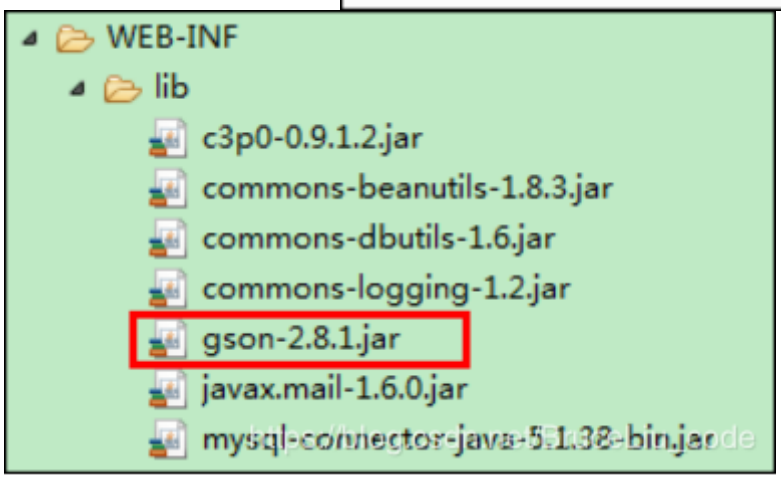
有时我们需要查看对象的结构！一般采用的方式是alert(obj)，但是只能看到下图所示的样子：

将js对象转换成json字符串的方法**JSON.stringify(obj)**，方便查看对象的结构！



4.JSON与Java对象之间的相互转换

- 导入解析JSON的工具包



- 代码示例：

```
1   public static void main(String[] args) {
2       Gson gson = new Gson();
```

删除水印 万兴PDF

```
        User user = new User("脏三", 20);
        List<User> uList = new ArrayList<User>();
        uList.add(new User("李四", 21));
        uList.add(new User("王五", 22));
        Map<String, Object> uMap = new HashMap<String, Object>();
        uMap.put("aa", 11);
        uMap.put("bb", 22);

        // java对象转json
        String json1 = gson.toJson(user);
        String json2 = gson.toJson(uList);
        String json3 = gson.toJson(uMap);
        /*System.out.println(json1); {name:"张三", age:20}
        System.out.println(json2); [{name:"张三", age:20},{name:"张三",
    age:20} ]
        System.out.println(json3);  {aa:11,bb:22}*/

        // json转java对象
        // json转对象
        User user1 = gson.fromJson(json1, User.class);
        // json转集合
        Type type = new TypeToken<List<User>>(){}.getType();
        List<User> uuList = gson.fromJson(json2, type);
        // json转map
        Type type2 = new TypeToken<Map<String, Object>>()
    {}.getType();
        Map<String, Object> map2 = gson.fromJson(json3, type2);
        System.out.println(user1);
        System.out.println(uuList);
        System.out.println(map2);
    }
```

5.异步加载下拉框（案例）