Ajax

# Json数据

**在介绍ajax之前先介绍一种弄数据格式json**

## 什么是json数据

**json是一种与语言无关的数据交换的格式，仅仅是一种有特定格式的数据**

作用：

使用ajax进行前后台数据交换 js User List

移动端与服务端的数据交换

**用于传输数据**

## Json数据格式

json有两种格式：

1. **对象格式：{"key1":obj,"key2":obj,"key3":obj...}**
2. **数组/集合格式：[obj,obj,obj...]**

### 对象格式

|  |
| --- |
| **public** **class** User {    **private** String name ;  **private** Integer age ;  **private** String password ;  **private** String address ; |

**这样一个简单的user对象转换成json格式的数据 :**

**{"name":"zhangsan","age":28,"password":"123456","address":"上海"}**

### 数组/集合格式的json数据 (List<User>)

**[{"name":"zhangsan","age":28,"password":"123456","address":"上海"},**

**{"name":"lisi","age":23,"password":"abcdef","address":"北京"},**

**{"name":"wangwu","age":21,"password":"654321","address":"广州"}**

**]**

**补充**：

**对象格式和数组格式可以互相嵌套**

**json的key是字符串 jaon的value是Object**

## json数据格式展示：

**json是js的原生内容，也就意味着js可以直接取出json对象中的数据**

### Json1

|  |
| --- |
| //创建一个json数据  **var** J\_json = {"name":"张三","age":18,"address":"北京"};  alert(J\_json.name); //张三  alert(J\_json.age);//18 |

### Json2

|  |
| --- |
| //创建一个json数据  **var** J\_json = [{"name":"张三","age":18,"address":"北京"},  {"name":"张四","age":19,"address":"南京"}  ] ;  //J\_json相当于java数组数据  alert(J\_json[0].name); //张三  alert(J\_json[1].name);//张四 |

### Json3

|  |
| --- |
| //创建一个json数据  **var** J\_json={  "user":[  {"name":"杨颖1","age":28,"addr":"扬州"},  {"name":"杨过1","age":18,"addr":"紫禁城"},  {"name":"杨洋1","age":10,"addr":"山西"},  ],  "user1":[  {"name":"杨颖2","age":28,"addr":"扬州"},  {"name":"杨过2","age":18,"addr":"紫禁城"},  {"name":"杨洋2","age":10,"addr":"山西"},  ]  };  //J\_json相当于java数组数据  alert(J\_json.user[0].name); //杨颖  alert(J\_json.user[1].name); //杨过  alert(J\_json.user[2].name); //杨洋  alert(J\_json.user1[0].name);  alert(J\_json.user1[1].name);  alert(J\_json.user1[2].name); |

**json的key是字符串 jaon的value是Object（可以是一个对象，一个数组/集合）**

### Json4

这是一个比较系统的json数据，里面几种常见的基本key/value格式 。

|  |
| --- |
| {  "key1":"value1",  "key2":{"firstname":"张","lastname":"三丰","age":100},  "key3":[  {"name":"小双","age":28,"addr":"扬州"},  {"name":"建宁","age":18,"addr":"紫禁城"},  {"name":"阿珂","age":10,"addr":"山西"},  ]  }; |

## 解析json数据

### Eval()函数js原生函数

|  |
| --- |
| $(**function**(){  //创建一个json数据  **var** str = '{ "name": "熊大", "sex": "公" }';  //将json格式的字符串转换成json对象  **var** J\_json = eval('('+str+')');  // 操作json对象  alert(J\_json.name)  }); |

### parseJSON()

var obj = str.parseJSON(); //由JSON字符串转换为JSON对象

|  |
| --- |
| //创建一个json数据  **var** str = '{ "name": "熊大", "sex": "公" }';  //将json格式的字符串转换成json对象  **var** json = str.parseJSON();  // 操作json对象  alert(json.name) |

**注：**

**找不到parseJSON()方法，则说明您的json包版本太低。**

### JSON.parse();

var obj = JSON.parse(str); //由JSON字符串转换为JSON对象

|  |
| --- |
| $(**function**(){  //创建一个json数据  **var** str = '{ "name": "熊大", "sex": "公" }';  //将json格式的字符串转换成json对象  **var** json = JSON.parse(str);  // 操作json对象  alert(json.name)  }); |

### 将json对象转换成json串

var last=obj.toJSONString(); //将JSON对象转化为JSON字符

**或者**

var last=JSON.stringify(obj); //将JSON对象转化为JSON字符

|  |
| --- |
| $(**function**(){  //创建一个json数据  **var** str = '{ "name": "熊大", "sex": "公" }';  //将json格式的字符串转换成json对象  **var** json = JSON.parse(str);  //將json对象转换成字符串  // var str1 = json.toJSONString();  **var** str1 = JSON.stringify(json);  // 打印json串  alert(str1);  }); |

### Json的转换插件

将java的对象或集合转成json形式字符串

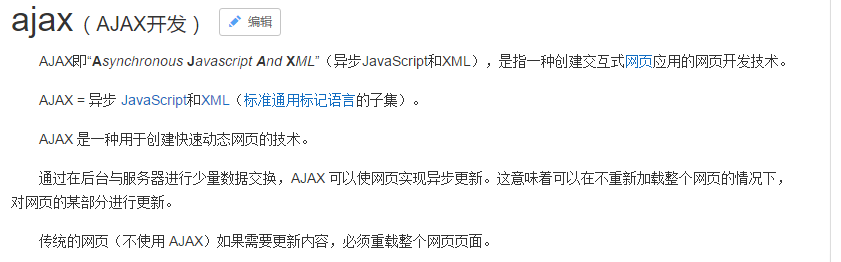
json的转换插件是通过java的一些工具，直接将java对象或集合转换成json字符串。

常用的json转换工具有如下几种：

1. jsonlib
2. Gson：google
3. fastjson：阿里巴巴

# Ajax

## 什么是ajax



## 什么是同步，什么是异步

同步现象：客户端发送请求到服务器端，当服务器返回响应之前，客户端都处于等待 卡死状态

异步现象：客户端发送请求到服务器端，无论服务器是否返回响应，客户端都可以随 意做其他事情，不会被卡死

## Ajax的运行原理

页面发起请求，会将请求发送给浏览器内核中的Ajax引擎，Ajax引擎会提交请求到 服务器端，在这段时间里，客户端可以任意进行任意操作，直到服务器端将数据返回 给Ajax引擎后，会触发你设置的事件，从而执行自定义的js逻辑代码完成某种页面功能。

## js原生的Ajax技术

js原生的Ajax其实就是围绕浏览器内内置的Ajax引擎对象进行学习的，要使用js原生的Ajax完成异步操作，**原生的js ajax了解即可** 。

**开发常用的jQuery中封装的ajax 。**

js原生的Ajax其实就是围绕浏览器内内置的Ajax引擎对象进行学习的，要使用js

原 生的Ajax完成异步操作，有如下几个步骤：

1. 创建Ajax引擎对象
2. 为Ajax引擎对象绑定监听（监听服务器已将数据响应给引擎）
3. 绑定提交地址
4. 发送请求
5. 接受响应数据



注意：如果是post提交

在发送请求之前设置一个头

xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");

总结：

所用异步访问都是ajax引擎

## jQuery的ajax（重点）

j**Query对js原生的ajax进行了封装，封装后的ajax的操作方法更简洁，功能更强大。**

### $.get(url, [data], [callback], [type])

其中：

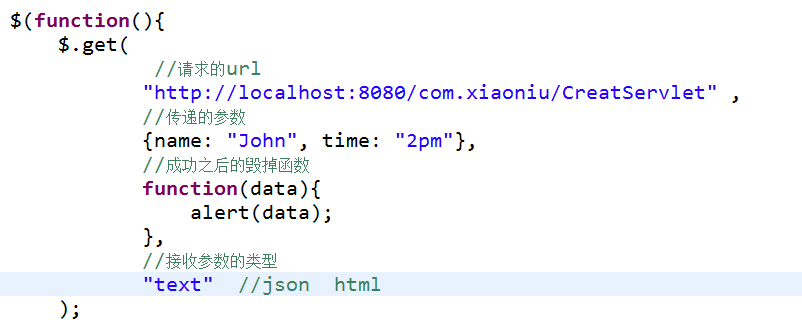
url：代表请求的服务器端地址

data：代表请求服务器端的数据（可以是key=value形式也可以是json格式）

callback：表示服务器端成功响应所触发的函数（只有正常成功返回才执行）

type：表示服务器端返回的数据类型（jquery会根据指定的类型自动类型转换）

常用的返回类型：text、json、html等



### $.post(url, [data], [callback], [type])

|  |
| --- |
| $(**function**(){  $.post(  //请求的url  "http://localhost:8080/com.xiaoniu/CreatServlet" ,  //传递的参数  {name: "John", time: "2pm"},  //成功之后的毁掉函数  **function**(data){  alert(data);  },  //接收参数的类型  "text" //json html  );    }); |

### $.ajax( { option1:value1,option2:value2... } )

常用的option有如下：

async：是否异步，默认是true代表异步

data：发送到服务器的参数，建议使用json格式

dataType：服务器端返回的数据类型，常用text和json

success：成功响应执行的函数，对应的类型是function类型

type：请求方式，POST/GET

url：请求服务器端地址

