## **json**

json是 JavaScript Object Notation 的首字母缩写，单词的意思是javascript对象表示法，这里说的json指的是类似于javascript对象的一种数据格式对象，目前这种数据格式比较流行，逐渐替换掉了传统的xml数据格式。

json数据对象类似于JavaScript中的对象，但是它的键对应的值里面是没有函数方法的，值可以是普通变量，不支持undefined，值还可以是数组或者json对象。

与JavaScript对象写法不同的是，json对象的属性名称和字符串值需要用双引号引起来，用单引号或者不用引号会导致读取数据错误。

json格式的数据：

{

"name":"tom",

"age":18

}

json的另外一个数据格式是数组，和javascript中的数组字面量相同。

["tom",18,"programmer"]

还可以是更复杂的数据机构：

{

"name":"jack",

"age":29,

"hobby":["reading","travel","photography"]

"school":{

"name":"Merrimack College",

"location":'North Andover, MA'

}

}

## **ajax与jsonp**

ajax技术的目的是让javascript发送http请求，与后台通信，获取数据和信息。ajax技术的原理是实例化xmlhttp对象，使用此对象与后台通信。ajax通信的过程不会影响后续javascript的执行，从而实现异步。

##### **同步和异步**

现实生活中，同步指的是同时做几件事情，异步指的是做完一件事后再做另外一件事，程序中的同步和异步是把现实生活中的概念对调，也就是程序中的异步指的是现实生活中的同步，程序中的同步指的是现实生活中的异步。

##### **局部刷新和无刷新**

ajax可以实现局部刷新，也叫做无刷新，无刷新指的是整个页面不刷新，只是局部刷新，ajax可以自己发送http请求，不用通过浏览器的地址栏，所以页面整体不会刷新，ajax获取到后台数据，更新页面显示数据的部分，就做到了页面局部刷新。

##### **数据接口**

数据接口是后台程序提供的，它是一个url地址，访问这个地址，会对数据进行增、删、改、查的操作，最终会返回json格式的数据或者操作信息，格式也可以是text、xml等。

##### **$.ajax使用方法**

常用参数：  
1、url 请求地址  
2、type 请求方式，默认是'GET'，常用的还有'POST'  
3、dataType 设置返回的数据格式，常用的是'json'格式，也可以设置为'html'  
4、data 设置发送给服务器的数据  
5、success 设置请求成功后的回调函数  
6、error 设置请求失败后的回调函数  
7、async 设置是否异步，默认值是'true'，表示异步

以前的写法：

$.ajax({

url: '/change\_data',

type: 'GET',

dataType: 'json',

data:{'code':300268}

success:function(dat){

alert(dat.name);

},

error:function(){

alert('服务器超时，请重试！');

}

});

新的写法(推荐)：

$.ajax({

url: '/change\_data',

type: 'GET',

dataType: 'json',

data:{'code':300268}

})

.done(function(dat) {

alert(dat.name);

})

.fail(function() {

alert('服务器超时，请重试！');

});

##### **$.ajax的简写方式**

$.ajax按照请求方式可以简写成$.get或者$.post方式

$.get("/change\_data", {'code':300268},

function(dat){

alert(dat.name);

});

$.post("/change\_data", {'code':300268},

function(dat){

alert(dat.name);

});

##### **课堂练习**

##### **同源策略**

ajax请求的页面或资源只能是同一个域下面的资源，不能是其他域的资源，这是在设计ajax时基于安全的考虑。特征报错提示：

XMLHttpRequest cannot load https://www.baidu.com/. No

'Access-Control-Allow-Origin' header is present on the requested resource.

Origin 'null' is therefore not allowed access.

##### **jsonp**

ajax只能请求同一个域下的数据或资源，有时候需要跨域请求数据，就需要用到jsonp技术，jsonp可以跨域请求数据，它的原理主要是利用了<script>标签可以跨域链接资源的特性。jsonp和ajax原理完全不一样，不过jquery将它们封装成同一个函数。

##### **课堂实例**

获取360搜索关键词联想数据

$(function(){

$('#txt01').keyup(function(){

var sVal = $(this).val();

$.ajax({

url:'https://sug.so.360.cn/suggest?',

type:'get',

dataType:'jsonp',

data: {word: sVal}

})

.done(function(data){

var aData = data.s;

$('.list').empty();

for(var i=0;i<aData.length;i++)

{

var $li = $('<li>'+ aData[i] +'</li>');

$li.appendTo($('.list'));

}

})

})

})

//......

<input type="text" name="" id="txt01">

<ul class="list"></ul>

##### **一般ajax数据接口和jsonp数据接口的区别**

开发返回数据的接口，如果是一般的ajax接口，让接口直接返回json格式的数据字符串就可以了，这种接口的数据是不能跨域请求的，如果要跨域请求数据，需要开发jsonp的接口，开发jsonp的接口，需要获取请求地址中的参数，也就是'callback'键对应的值，然后将这个值和json数据拼装成一个函数调用的形式的字符串返回，就完成了一个jsonp的接口，这个键值对是由$.ajax函数自动产生的。比如：'callback'键对应的值一般是这样的：

jQuery1124018787969015631711\_1522330257607，所以：

一般接口返回的数据形式：'{"iNum":12,'sTr':'abc'}';

jsonp返回的数据形式：'jQuery1124018787969015631711\_1522330257607({"iNum":12,"sTr":"abc"})';

## **数据交互**

vue.js没有集成ajax功能，要使用ajax功能，可以使用vue官方推荐的axios.js库来做ajax的交互。 axios库的下载地址：[https://github.com/axios/axios/releases](https://github.com/axios/axios/releases" \t "H:/%E5%A4%87%E4%BB%BD%E6%96%87%E4%BB%B6/%E8%AF%BE%E7%A8%8B%E8%B5%84%E6%96%99/python%E6%95%99%E5%AD%A6/python%E5%89%8D%E7%AB%AF(%E6%9C%80%E6%96%B0)/%E5%89%8D%E7%AB%AF/%E5%89%8D%E7%AB%AF%E8%AE%B2%E4%B9%89/008day/_blank)

#### **axios完整写法：**

axios({

method: 'post',

url: '/user/12345',

data: {

firstName: 'Fred',

lastName: 'Flintstone'

}

});

axios请求的写法也写成get方式后post方式。

#### **执行get请求**

// 为给定 ID 的 user 创建请求

// then是请求成功时的响应，catch是请求失败时的响应

axios.get('/user?ID=12345')

.then(function (response) {

console.log(response);

})

.catch(function (error) {

console.log(error);

});

// 可选地，上面的请求可以这样做

axios.get('/user', {

params: {

ID: 12345

}

})

.then(function (response) {

console.log(response);

})

.catch(function (error) {

console.log(error);

});

#### **执行post请求**

axios.post('/user', {

firstName: 'Fred',

lastName: 'Flintstone'

})

.then(function (response) {

console.log(response);

})

.catch(function (error) {

console.log(error);

});