**ajax 和jsonp相关知识**

**一、ajax**

AJAX即“**A***synchronous* **J***avascript* ***A****nd* **X***ML*”（异步JavaScript和XML），是指一种创建交互式[网页](http://baike.baidu.com/view/828.htm)应用的网页开发技术。

AJAX = 异步 [JavaScript](http://baike.baidu.com/view/16168.htm)和[XML](http://baike.baidu.com/view/63.htm)（[标准通用标记语言](http://baike.baidu.com/view/5286041.htm)的子集）。

AJAX 是一种用于创建快速动态网页的技术。

通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行局部更新。

**原生JS:**

//1.打开浏览器

var xhr = null;

//异常捕获处理

try{xhr = new XMLHttpRequest();

}catch(e){xhr = ActiveXObject('Microsoft.XMLHTTP');}

if (method=='get'&& data) {url += '?'+data;}

//2.在地址栏输入地址

xhr.open(method, url, true); //true-异步 false-同步

//3. 提交

if (method=='get') {xhr.send();

}else{

xhr.setRequestHeader('content-type', 'application/x-www-form-urlencoded');

xhr.send(data);}

//4. 等待服务返回内容

xhr.onreadystatechange = function(){

if(xhr.readyState == 4){

if(xhr.status == 200){success && success(xhr.responseText); //返回内容(字符串)}else{alert('出错了,err:'+xhr.status);}

};

};

**jQuery ：**

$.ajax({

type: "GET",

url: "test.json",

data: {content:$("#content").val()},

dataType: "json",

success: function(data){}

});

**async** --默认值: true。默认设置下，所有请求均为异步请求。如果需要发送同步请求，请将此选项设置为 false。注意，同步请求将锁住浏览器，用户其它操作必须等待请求完成才可以执行。

**data --**发送到服务器的数据。将自动转换为请求字符串格式。GET 请求中将附加在 URL 后。查看 processData 选项说明以禁止此自动转换。必须为 Key/Value 格式。如果为数组，jQuery 将自动为不同值对应同一个名称。如 {foo:["bar1", "bar2"]} 转换为 '&foo=bar1&foo=bar2'。

**dataType --**预期服务器返回的数据类型。可用值:

"xml": 返回 XML 文档，可用 jQuery 处理。

"html": 返回纯文本 HTML 信息；包含的 script 标签会在插入 dom 时执行。"script": 返回纯文本 JavaScript 代码。不会自动缓存结果。除非设置"cache" 参数。注意：在远程请求时(不在同一个域下)，所有 POST 请求都将转为 GET 请求。（因为将使用 DOM 的 script标签来加载）

"json": 返回 JSON 数据 。

"jsonp": JSONP 格式。使用 JSONP 形式调用函数时，如 "myurl?callback=?" jQuery 将自动替换 ? 为正确的函数名，以执行回调函数。

"text": 返回纯文本字符串

**jsonp --**在一个 jsonp 请求中重写回调函数的名字。这个值用来替代在 "callback=?" 这种 GET 或 POST 请求中 URL 参数里的 "callback" 部分，比如 {jsonp:'onJsonPLoad'} 会导致将 "onJsonPLoad=?" 传给服务器。

**success --**类型：Function请求成功后的回调函数。参数：由服务器返回，并根据 dataType 参数进行处理后的数据；描述状态的字符串。这是一个 Ajax 事件。

**type --**类型：String默认值: "GET"。请求方式 ("POST" 或 "GET")， 默认为 "GET"。注意：其它 HTTP 请求方法，如 PUT 和 DELETE 也可以使用，但仅部分浏览器支持。

**url**--类型：String默认值: 当前页地址。发送请求的地址。

1. **jsonp**
2. json 和jsonp 的区别

JSON是一种基于文本的数据交换方式，或者叫做数据描述格式，你是否该选用他首先肯定要关注它所拥有的优点。

JSON的优点：

1、基于纯文本，跨平台传递极其简单；

2、Javascript原生支持，后台语言几乎全部支持；

3、轻量级数据格式，占用字符数量极少，特别适合互联网传递；

4、可读性较强，虽然比不上XML那么一目了然但在合理的依次缩进之后还是很容易识别的；

5、容易编写和解析，当然前提是你要知道数据结构；

JSON的缺点当然也有，但在作者看来实在是无关紧要的东西，所以不再单独说明。

格式：

var person = {"Name": "Bob","Age": 32,"Company": "IBM","Engineer": true}

什么是JSONP

　　先说说JSONP是怎么产生的：

　　其实网上关于JSONP的讲解有很多，但却千篇一律，而且云里雾里，对于很多刚接触的人来讲理解起来有些困难，小可不才，试着用自己的方式来阐释一下这个问题，看看是否有帮助。

　　1、一个众所周知的问题，Ajax直接请求普通文件存在跨域无权限访问的问题，甭管你是静态页面、动态网页、web服务、WCF，只要是跨域请求，一律不准；

　　2、不过我们又发现，Web页面上调用js文件时则不受是否跨域的影响（不仅如此，我们还发现凡是拥有”src”这个属性的标签都拥有跨域的能力，比如<script>、<img>、<iframe>）；

　　3、于是可以判断，当前阶段如果想通过纯web端（ActiveX控件、服务端代理、属于未来的HTML5之Websocket等方式不算）跨域访问数据就只有一种可能，那就是在远程服务器上设法把数据装进js格式的文件里，供客户端调用和进一步处理；

　　4、恰巧我们已经知道有一种叫做JSON的纯字符数据格式可以简洁的描述复杂数据，更妙的是JSON还被js原生支持，所以在客户端几乎可以随心所欲的处理这种格式的数据；

　　5、这样子解决方案就呼之欲出了，web客户端通过与调用脚本一模一样的方式，来调用跨域服务器上动态生成的js格式文件（一般以JSON为后缀），显而易见，服务器之所以要动态生成JSON文件，目的就在于把客户端需要的数据装入进去。

　　6、客户端在对JSON文件调用成功之后，也就获得了自己所需的数据，剩下的就是按照自己需求进行处理和展现了，这种获取远程数据的方式看起来非常像AJAX，但其实并不一样。

　　7、为了便于客户端使用数据，逐渐形成了一种非正式传输协议，人们把它称作JSONP，该协议的一个要点就是允许用户传递一个callback参数给服务端，然后服务端返回数据时会将这个callback参数作为函数名来包裹住JSON数据，这样客户端就可以随意定制自己的函数来自动处理返回数据了。

如果对于callback参数如何使用还有些模糊的话，我们后面会有具体的实例来讲解。

1、ajax和jsonp这两种技术在调用方式上“看起来”很像，目的也一样，都是请求一个url，然后把服务器返回的数据进行处理，因此jquery和ext等框架都把jsonp作为ajax的一种形式进行了封装；

　　2、但ajax和jsonp其实本质上是不同的东西。ajax的核心是通过XmlHttpRequest获取非本页内容，而jsonp的核心则是动态添加<script>标签来调用服务器提供的js脚本。

　　3、所以说，其实ajax与jsonp的区别不在于是否跨域，ajax通过服务端代理一样可以实现跨域，jsonp本身也不排斥同域的数据的获取。

　　4、还有就是，jsonp是一种方式或者说非强制性协议，如同ajax一样，它也不一定非要用json格式来传递数据，如果你愿意，字符串都行，只不过这样不利于用jsonp提供公开服务。

总而言之，jsonp不是ajax的一个特例，哪怕jquery等巨头把jsonp封装进了ajax，也不能改变这一点！

jsonp跨域解决方案，其他的跨域方式？

http://www.cnblogs.com/2050/p/3191744.html