# 笔记位置：

<https://blog.csdn.net/qq_44799924/article/details/115710612>

# HTTP请求过程

HTTP协议是超文本传输协议，可以通过此协议完成web server和客户端浏览器之间的数据传输文本和超文本模块的发送(比如 web 页面中的 HTML, 图片, 文件)。就是将所有的数据打包成二进制，采用json字符串的形式去发送。 并且通常使用 80端口，这就是网站 URL 前面 ‘http’的来源。

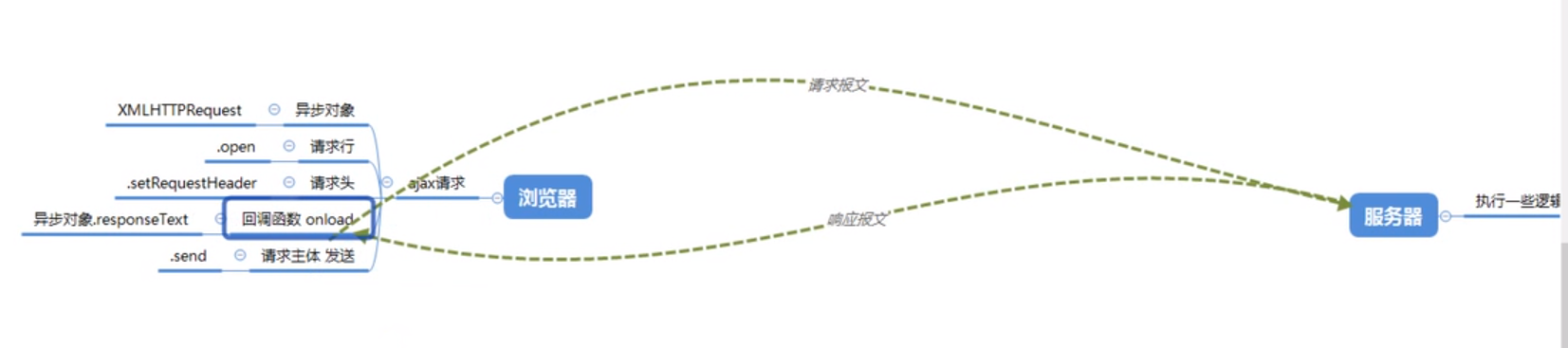
所有的数据传输都采用字符串形式, 这是由于硬件决定的，因为计算机存贮数据只能0和1 存储，即二极管的亮或灭代表了一位bit，1byte有8bit，即8个0或1表示的二极管。所以只能模拟1和0，意味着只能发送字符串，就意味着json的出现。 因为字符串形式可以替代其他所有类型，比如整形和bool类型等，只要加上单引号就变成了字符串。

## 同步请求

同步请求：顺序处理，即当我们向服务器发出一个请求时，在服务器没返回结果给客户端之前，我们要一直处于等待状态直至服务器将结果返回到客户端，我们才能执行下一步操作, 即程序的执行与任务的排列顺序是一致的，同步的。（例如普通的B/S模式就是同步请求，注：B/S模式 也即服务器与浏览器通信主要采用HTTP协议；通信方式为 “请求——响应” ，浏览器发出请求；服务器做出响应。）例如，使用form表单发送请求有一个缺点，点击提交页面就会跳转.

## 异步请求asynchronous request

异步请求：并行处理，当我们向服务器发出一个请求时，在服务器没返回结果之前，我们还是可以执行其他操作。例如AJAX技术就是异步请求。在不刷新整个页面的前提下，可以对部分内容发送请求给服务器，并接收服务器返回的数据内容。例如输入了用户名之后，就知道用户名是否已经被占用。现在大部分网站都比较流行异步请求。那么如果不想页面跳转，又要把数据发送到服务器去，就需要自己人为发送一个请求报文，自己认为设置请求头，请求行，请求主体。



### Ajax核心

Ajax (Asynchronous JavaScript and XML), 是数据交流的一套规则, 是指围绕[由js向服务器发起http请求] 这个功能而开发的一整套完整的做法. 其核心是 XMLHttpRequest（XHR）异步请求对象，是一个JS对象，可以自己人为模拟浏览器设置请求报文，并接收响应报文，XHR在服务器和浏览器之间提供了接口，使浏览器能够以异步(asynchronous)方式和服务器之间传递数据，即不刷新当前页面。

### Xhr对象常用的属性和方法

Text, letter

Description automatically generated

Text, letter

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, receipt, screenshot

Description automatically generated

A picture containing text, screenshot, receipt

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

Text, letter

Description automatically generated

在html页面的script标签中新建一个XHR对象，模拟发送请求报文，然后在得到响应的内容后，对响应内容进行处理。以下为原生的ajax实现（就是用XHR对象的方法和属性模拟浏览器HTTP请求的发送，并对返回结果数据进行处理），比较麻烦。

模拟Get请求发送报文

Text

Description automatically generated

模拟Post请求发送报文

Text

Description automatically generated

企业在开发当中， 前端和后端在交换数据的时候，一般都不会直接返回一个字符串，一般情况下，都会返回XML和JSON格式。什么是XML呢？可扩展的标记语言，也是需要标签将数据包裹，类似于HTML，HTML就是可扩展的标记语言。XML语法上和HTML比较相似，但HTML中的元素是固定的，而XML的标签是可以由用户自定义的。<https://blog.csdn.net/axi295309066/article/details/52760959> （XML）

### Ajax框架的封装（可看视频）

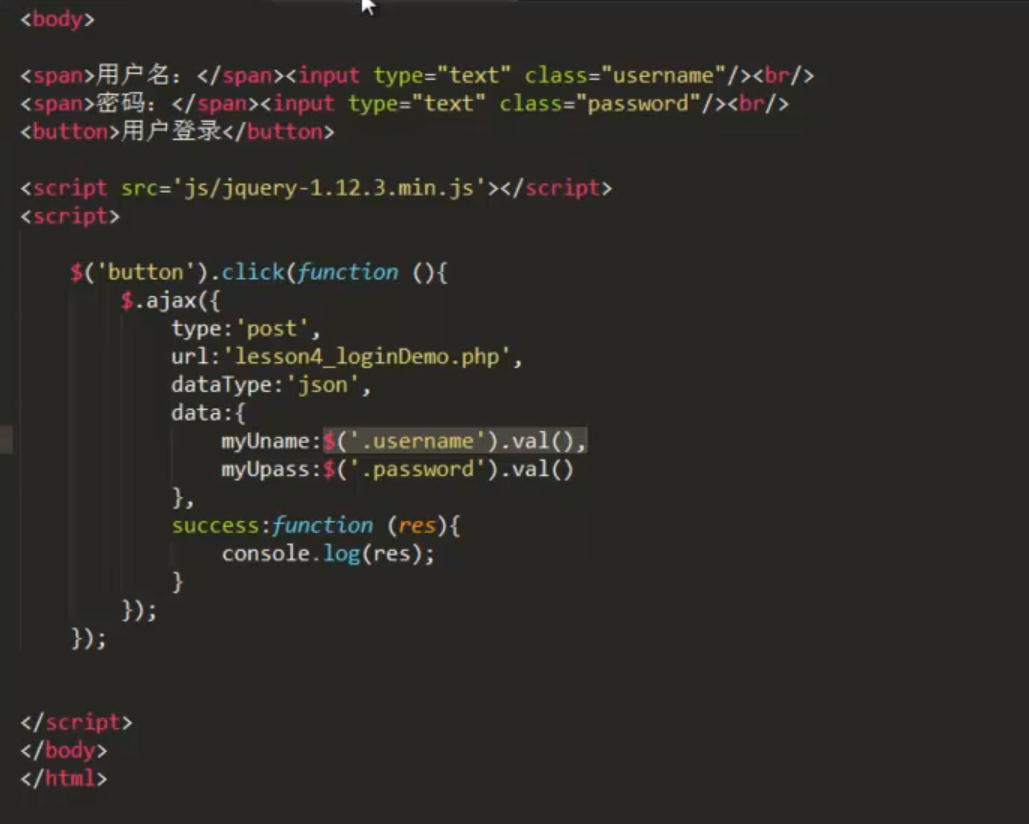
### jQuery中ajax请求案例

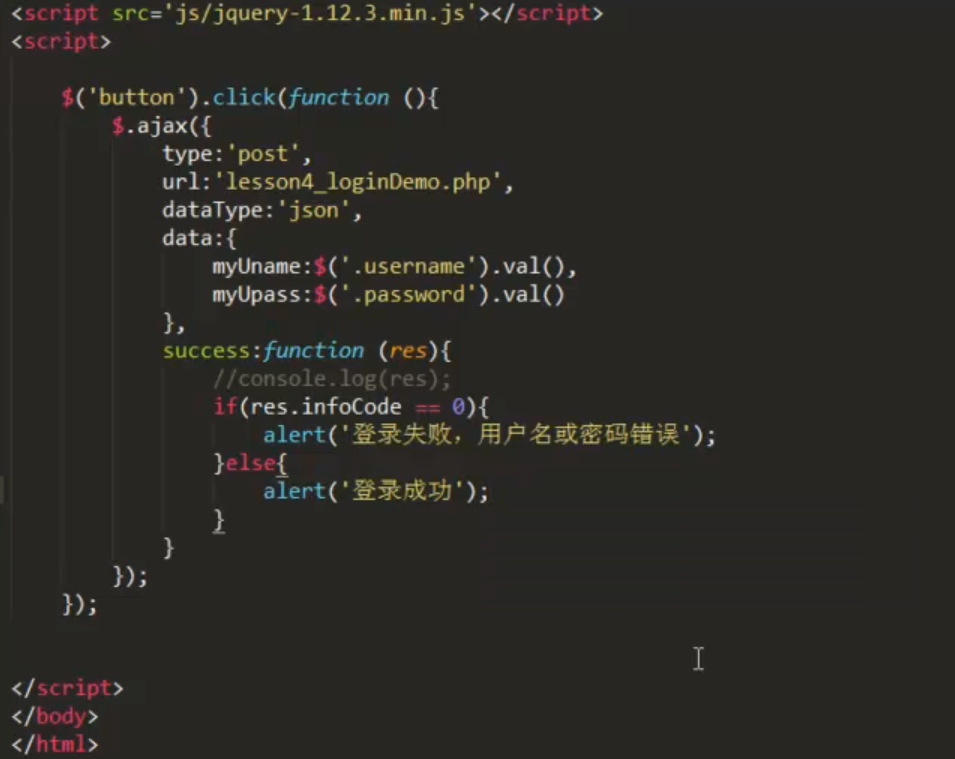
jQuery中使用$.ajax（）方法发送请求：

get无参请求，一般向后台获取数据使用，不会发送给后台任何内容。

get带参请求，一般需要根据属性获取数据时使用，这种情况下参数一般都比较小，将发送数据写在url后面拼接。

post请求：一般用于登录，注册等保密性比较高的位置，或者提交文件。将发送数据写在data字段中。





布局结构：获取DOM对象，然后调用点击事件，触发回调函数，在回调函数中带哦用$.ajax()函数，其参数实质上都是js对象，比如：{name:”simon”,age:”33”}，具体如下图所示：

Text

Description automatically generated with low confidence

当反馈给前端数据的时候，需要使用echo返回真实数据给前端。



Return作为函数返回值或结束语句，是返回数据的指针。也就是数据存放的物理地址。而http请求或接口要求返回的都是真实的数据，需要被渲染在页面上。所以当通过ajax请求数据时，需要echo这些能够在浏览器中打印数据的函数，前端ajax才可以获取和判断。

Graphical user interface, text, application

Description automatically generated

# **jQuery**

## JS与jQuery入口函数区别

 $() 是jQuery的核心函数，会返回一个jQuery对象给我们，对象中保存了找到的DOM元素节点对象。也就是说对象里保存了一个对象。这里要理解好3个对象的概念，即jQuery对象，js对象，DOM对象。

## jQuery对象

· 有0到length-1的属性

· 并且有length属性

伪数组是一个含有length属性的JS对象, {}包裹的，默认key，可以不加引号的。

它是按照索引（数字）的方式存储数据。

它并不具有数组的一些方法，只能能通过Array.prototype.slice转换为真正的数组，并且带有length属性的对象

例如

var obj = {0:"lnj", 1:"33", 2:"male", length: 3}

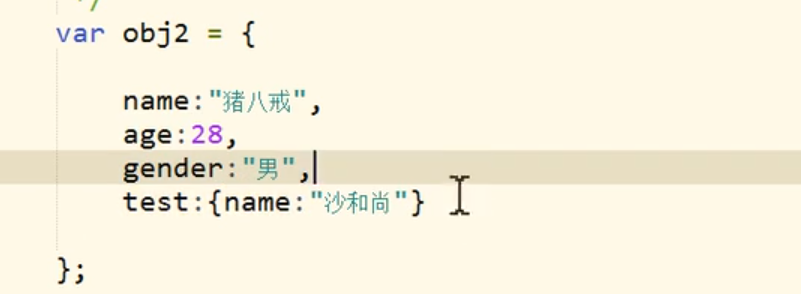
## JS对象（Object）

基本数据类型都是单一的值，值与值之间没有任何的关系，不能成为一个整体。

对象是JS中的引用数据类型

\*\*对象是一种复合数据类型，在对象中可以保存多个不同数据类型的属性\*\*

使用typeof检查一个对象时，会返回object。在对象中保存的值称为属性，属性值可以是任意类型的值，如果其属性是函数的话，我们一般称作为方法，也可以是一个对象。就像在塑料袋里面的塑料袋里找东西。



## JSON字符串

**javascript object notation，是js对象的一种特殊字符串表示方法，将js对象转换为一种特殊的字符串格式，这样的字符串可以被任意的后台语言所识别，并且可以转换为任意后台语言中的对象，例如在PHP中，可利用**json\_decode将json对象转换成php对象**，jSON在开发中主要用来数据的交互。**

**JSON字符串和JS对象的格式一样，只不过JSON字符串中的属性名必须加双引号，JS对象的键名可加单引号，也可是单引号，也可以不加引号。其他和JS语法完全一致。所以当外面没有单引号包裹的时候，就是一个JS对象，如果key使用双引号包裹的时候，既是一个js对象，也是json对象。但没有双引号包裹的时候，肯定是一个JS对象。**

**JSON分类：对象{}和数组[], Json中允许的值：字符串，数值，布尔值，null，对象，数组，不可以是JS函数，因为JS中的函数是无法被其他后台语言识别的。**

|  |
| --- |
| **var obj = {a: 'Hello', b: 'World'}; //这是一个JS对象，注意键名也是可以使用引号包裹的** |

二者主要差别：**JSON字符串中的属性名必须加双引号**

|  |  |
| --- | --- |
| **1** | **var json = '{"a": "Hello", "b": "World"}'; //这是一个 JSON 字符串，本质是一个字符串，属性名必须加双引号。其实里面包含了json对象，除了键名必须双引号，其他和js对象相同。**  **var arr = ‘[1,2,3,“hello”,true]’;这是一个json数组** |

## DOM元素节点对象

将网页中每一个部分都转换为一个DOM节点对象，就可以用面向对象的方式操纵网页了。 在网页中浏览器已经为我们提供了\*\*document对象\*\*，

## forEach()函数

The Array.forEach() method calls a function for each element in an array.数组里的每个元素都执行一遍回调函数或者是匿名函数。

Text

Description automatically generated

## 匿名函数？

此外，我们可以直接在另一个函数内定义一个函数，而不是调用它，比如：

setTimeout(function() {

    console.log("This message is shown after 3 seconds");

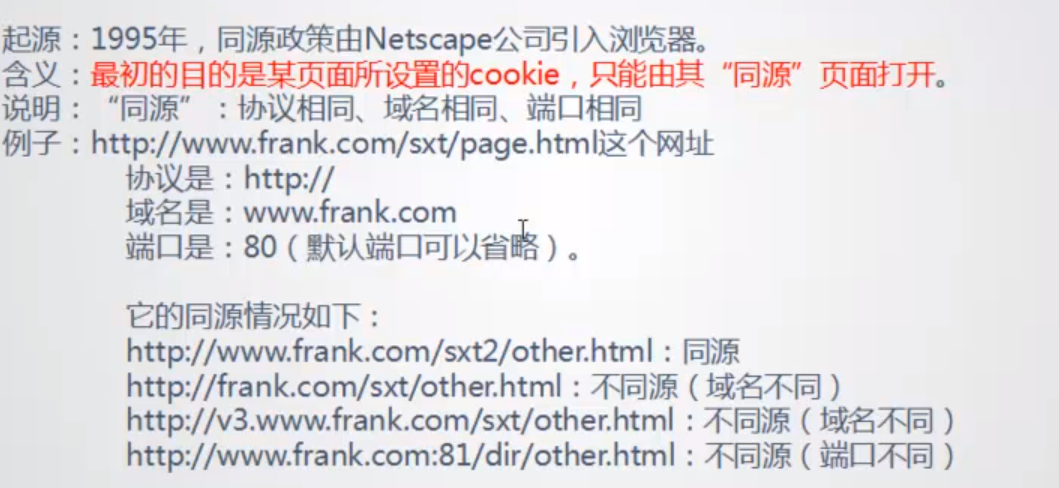
}, 3000);

我们可以看到，这里的回调函数没有名称。在 JavaScript 里，一个没有名称的函数叫作“匿名函数”。这个例子和上面的例子执行的任务一样。

setTimeout() 是设定一个指定等候时间 (单位是千分之一秒, millisecond), 时间到了, 浏览器就会执行一个指定的代码或者函数。1s=1000ms

## 浏览器同源政策

同源政策的目的，是为了保证用户信息的安全，防止恶意的网站窃取数据。同源政策目的，只有同一个服务器上的文件之间才能互相交互，多个服务器之间的文件禁止互相交互。但是在有些情况下合理的用途也会因同源政策而受到影响。因此就有了跨域的由来。使用script标签引入文件的方式，来规避同源政策，实现跨域访问。



## 属性节点

对象身上保存的变量就是属性。属性和属性节点有什么区别呢，任何对象都有属性, 但是只有DOM对象才有属性节点。

节点是构成网页的最基本的单元，网页中的每一个部分都可以称为是一个节点，虽然都是节点，但是节点的类型却是不同的。

常用的节点

文档节点 （Document），代表整个网页

元素节点（Element），代表网页中的标签

属性节点（Attribute），代表标签中的属性

文本节点（Text），代表网页中的文本内容

## Jquery中巧用Ajax的beforeSend方法

防止重复数据

在实际项目开发中，提交表单时常常由于网络或者其原因，用户点击提交按钮误认为自己没有操作成功，进而会重复提交按钮操作次数，如果页面前端代码没有做一些相应的处理，通常会导致多条同样的数据插入数据库，导致脏数据的增加。要避免这种现象，在$.ajax请求中的beforeSend方法中把提交按钮禁用掉，等到Ajax请求执行完毕，在恢复按钮的可用状态。

模拟Toast效果

<https://m.xp.cn/b.php/105257.html>

## disabled属性

只要在表單元素(Button、Input、Optgroup、Option、Select 及 TextArea)的屬性中加入 disabled 屬性的話，該元素就會被停用了。

//两种方法设置disabled属性  
$('#areaSelect').attr("disabled",true);  
$('#areaSelect').attr("disabled","disabled");  
//三种方法移除disabled属性  
$('#areaSelect').attr("disabled",false);  
$('#areaSelect').removeAttr("disabled");  
$('#areaSelect').attr("disabled","");

## 事件委托delegate

 有的时候元素是新增的（出生的），即在页面生成的时候，还没有出来，比如上传附件，没有办法对没有生成的元素进行事件加载。所以一般需要将事件委托给父元素，当子事件出生的时候，也会携带此事件。如下所示：会给class属性等于up\_delete的元素添加点击事件，当点击时执行回调匿名函数。不过其将事件添加给了document元素，因为document对象是一直存在的。如果直接添加给up\_delete元素的话，新增的元素就不会拥有此事件。143上附件上传删除就是应用的这个方法。

$(document).on('click','.up\_delete',function(){

});

## 事件冒泡propagation

因为事件从内部元素“冒泡”到所有父级，就像在水里的气泡一样（同一事件在父级标签中也启动，尽管父标签中没有定义相关事件）。但有时候我们只希望事件在子元素中触发，即不会在父级元素中事件触发，那我们就需要阻止事件冒泡。

event.stopPropagation();

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_event_stoppropagation>

## 默认行为

因为a标签和提交按钮等都存在着默认行为，在没有定义事件的时候，其依然可以进行跳转，所以要组织默认行为。

## $(this)

$('#search\_result tbody tr').live('dblclick', function(e){  
 *//$(this) 表示绑定事件发生的当前对象* var id = $(this).find('.id').text();  
 }

## 绑定事件的四种方法

<https://www.cnblogs.com/guorange/p/7244762.html>

## $.get()函数

## **jQuery.get( url [, data ] [, success ] [, dataType ] )**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $.ajax({  url: url,  data: data,  success: success,  dataType: json  }); |

以上两个是相等的。

## HTML代码/文本值相关操作

.html()用为读取和修改元素的HTML标签，相当于 innerHTML

.text()用来读取或修改元素的纯文本内容，相当于 innerTEXT

.val()用来读取或修改表单元素的value值，只有有value属性的才能取到值

.html(),.text(),.val()三种方法都是用来读取选定元素的内容；只不过.html()是用来读取元素的HTML内容（包括其Html标签），.text()用来读取元素的纯文本内容，包括其后代元素，.val()是用来读取表单元素有"value"值属性的元素内容。其中.html()和.text()方法不能使用在表单元素上,而.val()只能使用在表单元素上；另外.html()方法使用在多个元素上时，只读取第一个元素；.val()方法和.html()相同，如果其应用在多个元素上时，只能读取第一个表单元素的"value"值，但是.text()和他们不一样，如果.text()应用在多个元素上时，将会读取所有选中元素的文本内容。

## Target返回触发该事件的节点，如生成事件的元素，文档或窗口。

IE下,event对象有srcElement属性,但是没有target属性;

Firefox下,event对象有target属性,但是没有srcElement属性.但他们的作用是相当的，所以可以使用以下解决办法

var target = e.target || e.srcElement;

## 获取checkbox的值，当点击checkbox的时候，事件发生。

$('input:checkbox').change( function() {

console.log("here i am: " + this.name + " value = " + $(this).val() );

}

);