## 预习知识

1. 网址的组成：协议+域名+文件夹+文件名+端口+搜索关键字
2. 需要配置两个虚拟主机（zhangsan.com,lisi.com）

## 知识点讲解流程

ajax基础知识 --> 同步与异步 --> 数据格式(xml,json) --> 封装ajax --> jquery中使用ajax --> 跨域 --> 封装jsonp --> 完善ajax函数（支持ajax与jsonp) --> 模板引擎

## 视频播放速度

1.1或者1.2倍

## 难点

1. 同步与异步
2. 封装ajax方法
3. 跨域
4. 模板引擎

视频中并没有提到的知识点（可以补充讲解）

1. JSON.parse对应的JSON.stringify
2. json数据格式(包含 “null”,”[数组]”,”12数字”,”{对象}”)
3. 如果效果出不来如何调试（通过network查看）
4. json view插件
5. postman插件
6. fiddler工具的使用（不建议，操作界面全英文不够友好）

## 01.初识Ajax.avi-06-35

总结：

1. ajax的作用
   1. 获取服务器的数据
2. ajax的效果
   1. 在不刷新整个页面的情况下，通过一个url地址获取服务器的数据，然后进行页面的局部刷新

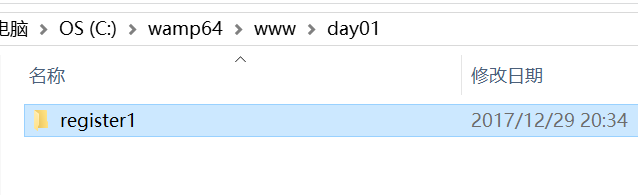
3、小结

六个字：局部 异步刷新（异步这）

Ajax的全称：Asynchronous Javascript And XML（这几个单词不用记，了解即可，只需要记住async代表的是异步 sync代表的是同步即可），就是使用JS代码获取服务器数据

## 02.Ajax的使用\_Ajax使用的基本场景和步骤.avi-20:43

代码见：



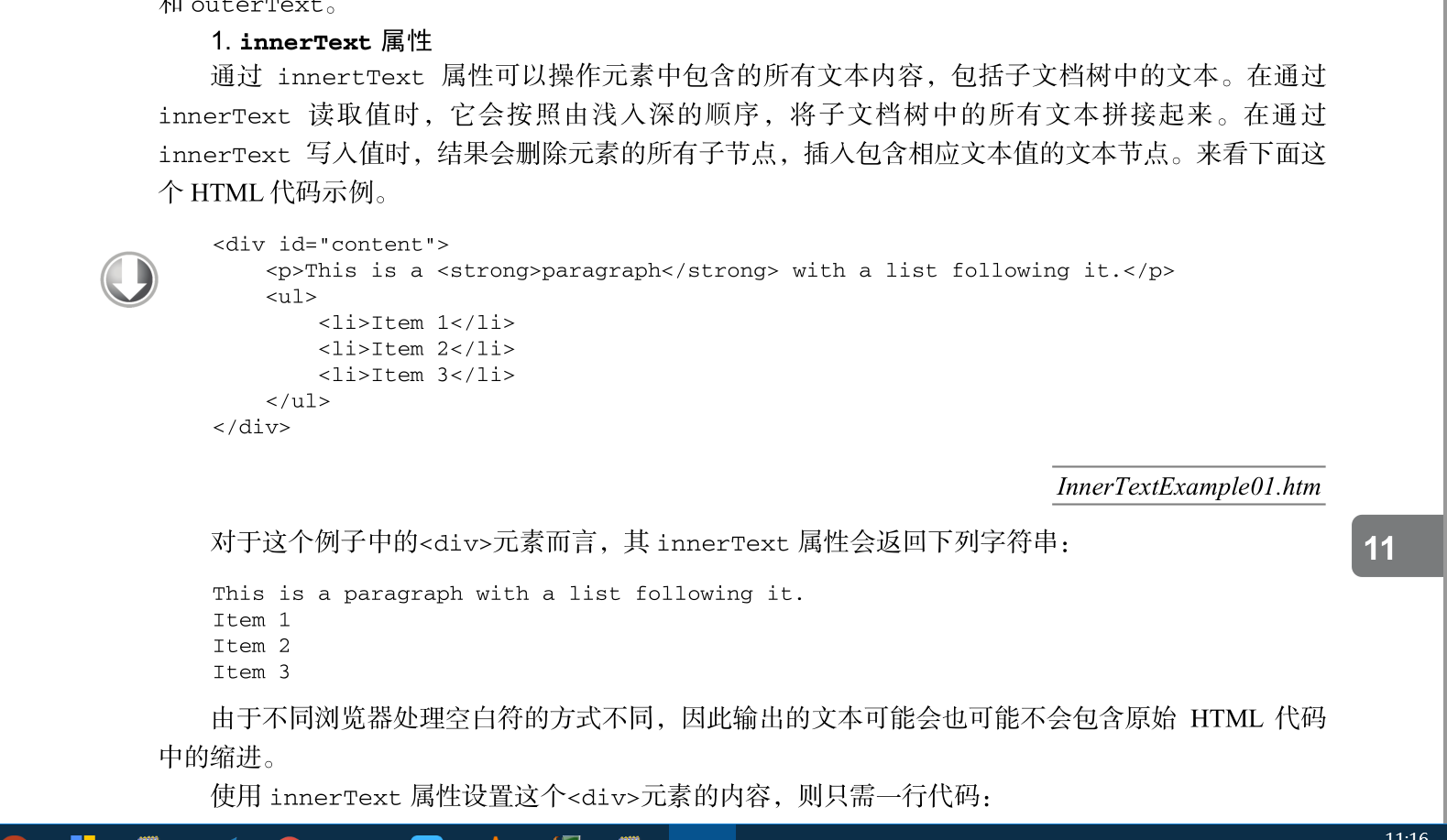
总结：（这块大家先把代码看一遍，不要深究，下一个视频才会对这个代码进行具体的讲解）

1、书写ajax的基本代码

* 1. 创建XMLHttpRequest对象
  2. 准备发送网络请求
  3. 开始发送网络请求
  4. 指定回调函数







参考代码：

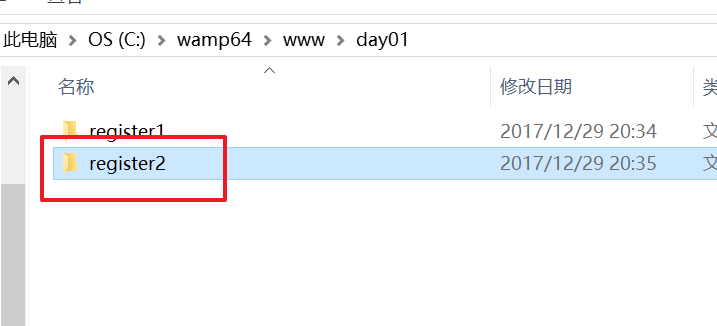
|  |
| --- |
| register.html  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>注册界面</title>  <script>  window.onload = function(){  var btn = document.getElementById('btn');  btn.onclick = function(){  var username = document.getElementById('username').value;  //用js代码进行checkUsername.php这个文件的访问，将username传递给这个文件  var xhr = new XMLHttpRequest();  xhr.open('get','checkUsername.php?username='+ username,true);  xhr.send(null);  xhr.onreadystatechange = function(){  var result = xhr.responseText;  // console.log(result);  document.getElementById('result').innerText = result;  }  }  }  </script>  </head>  <body>  <h1>注册界面</h1>  <form action="register.php" method="get">    用户名：<input type="text" name="username" id="username">  <input type="button" value="验证用户名" id="btn">  <span id="result"></span>  <br>  密码：<input type="password" name="password"><br>  <input type="submit" value="提交">  </form>  </body>  </html> |

|  |
| --- |
| checkUsername.php  <?php  $uname = $\_GET["username"];  //按道理来说，这个地方应该是要查询数据库  if($uname == 'zhangsan') {  echo "username exists";  } else {  echo "username ok";  }  ?> |

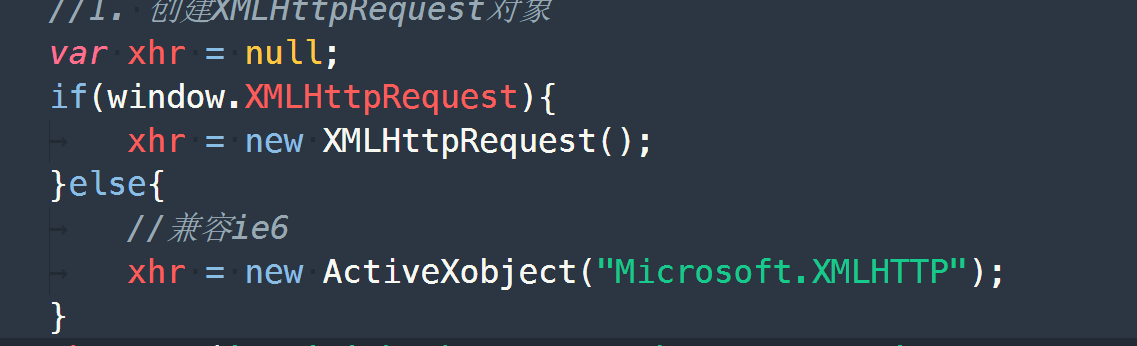
## 03.Ajax的使用\_Ajax中4个步骤的讲解.avi-18:09

总结：

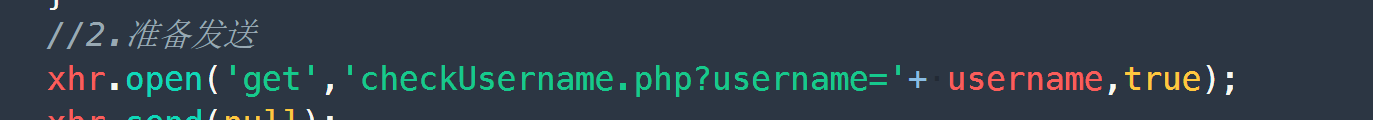
代码见：



步骤一：

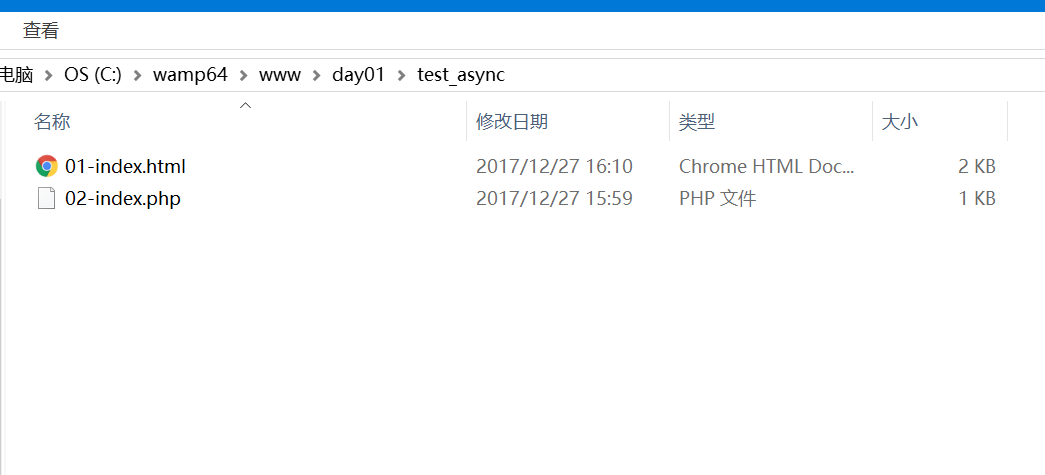


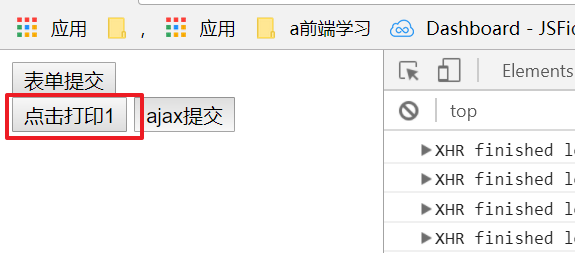
步骤二：



第三个参数可以不写，默认代表的是异步

这块可以测试写的这个代码玩一下感受同步与异步的区别：



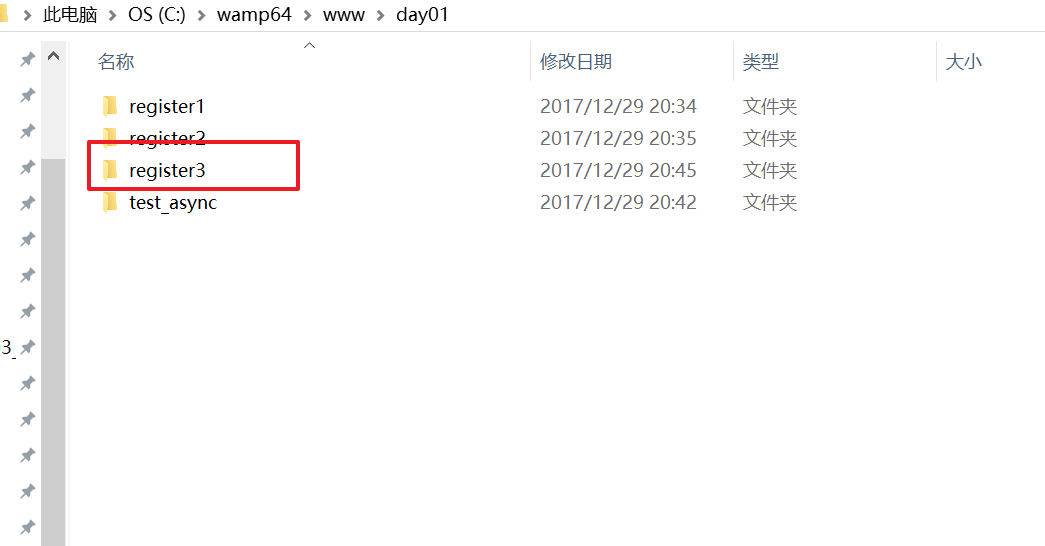


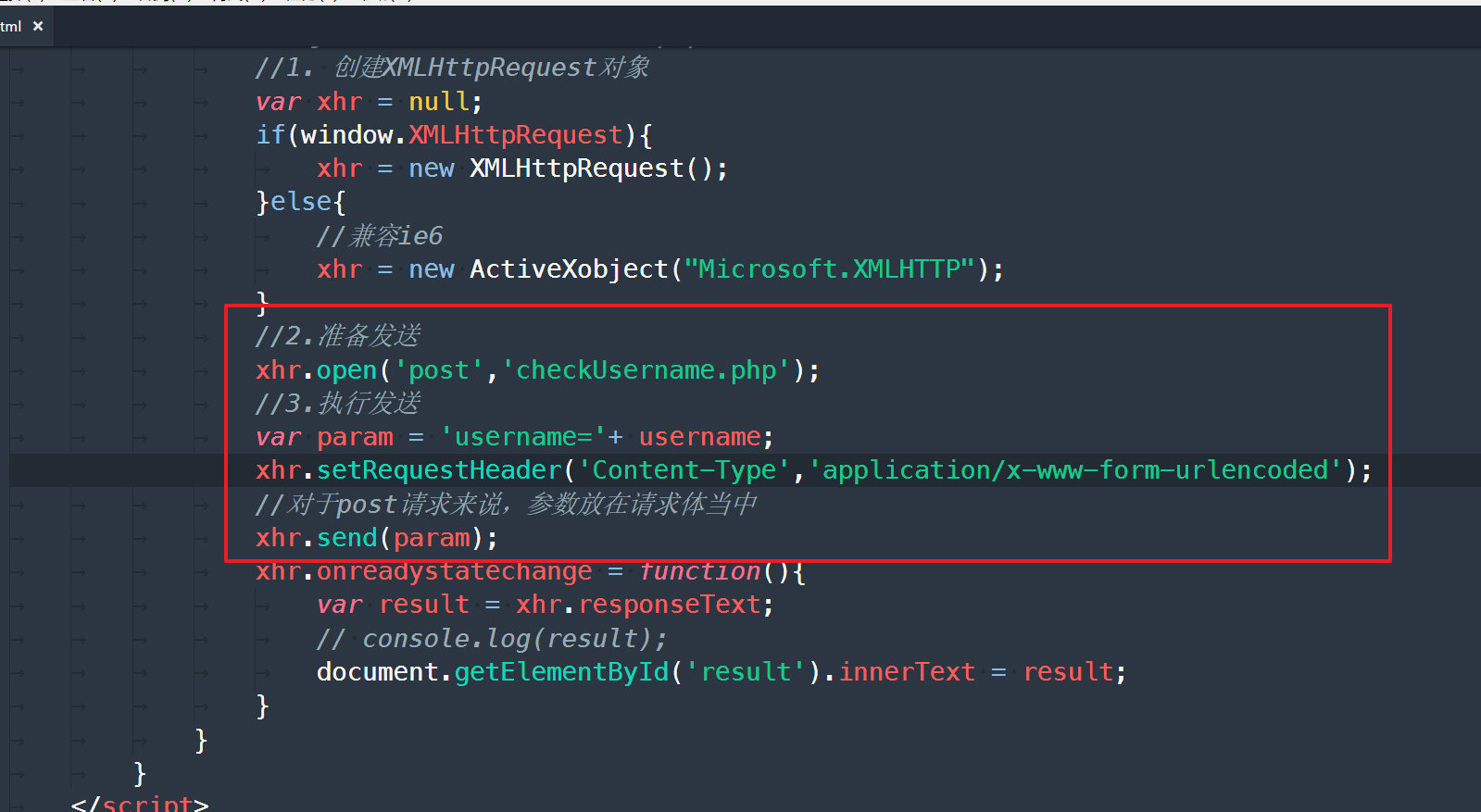
同步的情况下如果点击这个“点击打印1”是没有反应的

步骤三：

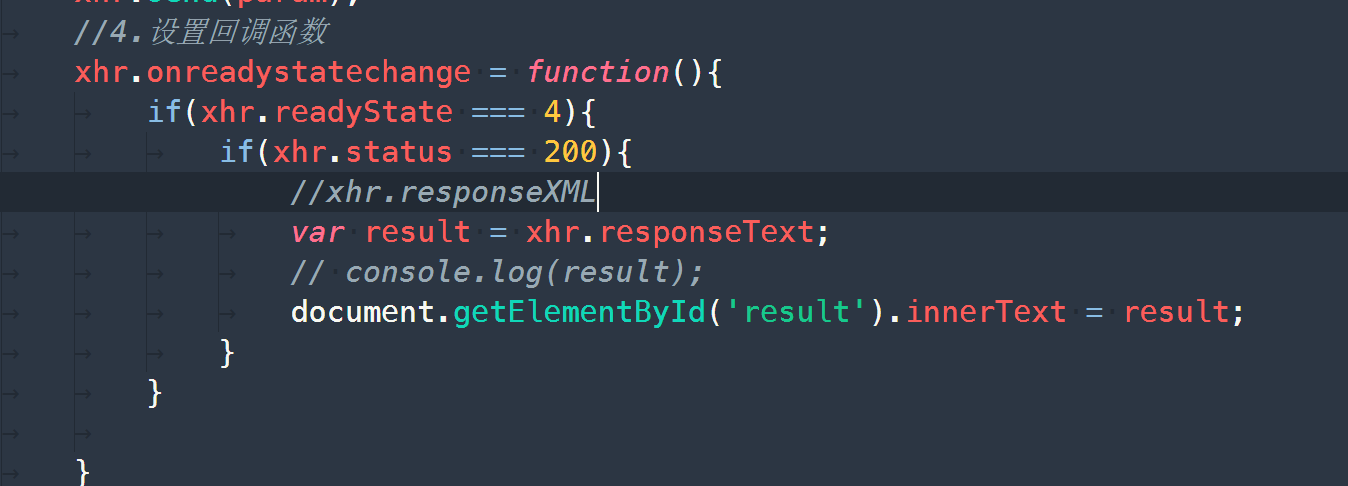


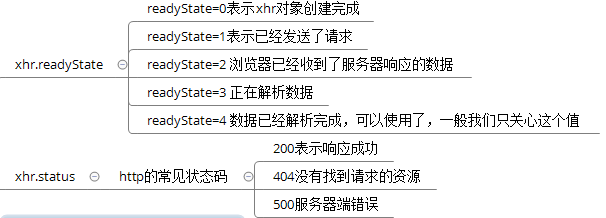
改造代码使用POST方式：





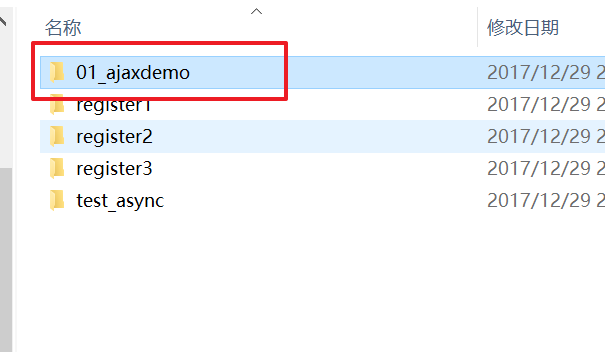
步骤四：





## 04.Ajax的使用\_案例01\_验证用户名唯一性.avi-20:41

代码：

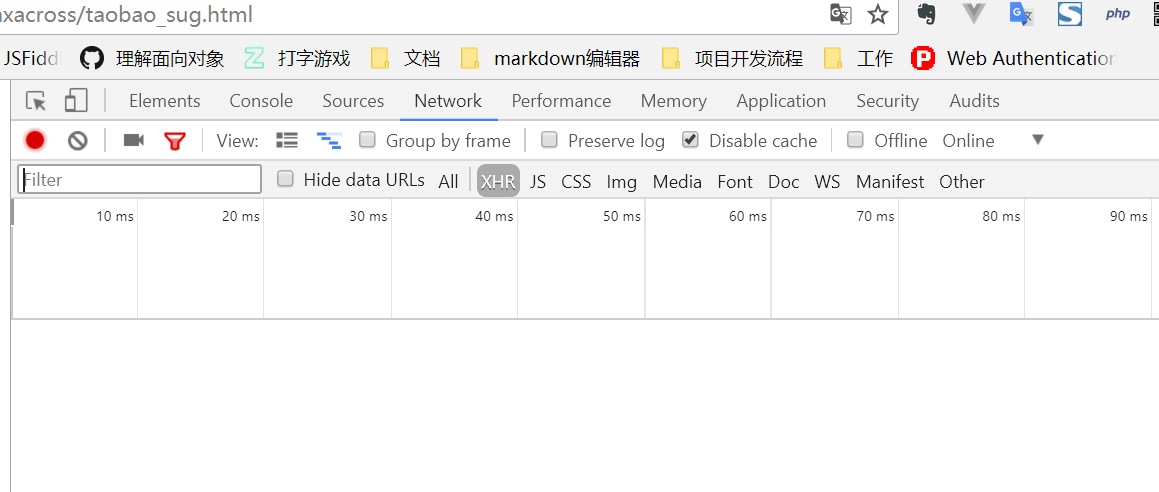


总结：

1. 校验应该是用户输入完信息按tab键的时候进行校验
2. 学会使用接口文档
   1. 不要自己去看php代码
   2. 通过文档来了解请求的地址、请求的方法、请求的参数
   3. 通过文档了解返回数据的含义

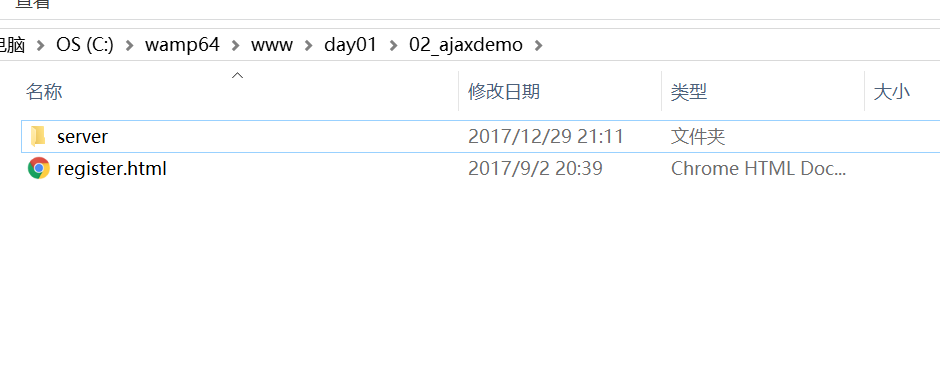
补充：

这块可以讲解一下浏览器的network中的xhr的使用（后面的同源策略直接使用了）



## 05.Ajax的使用\_案例02\_验证邮箱唯一性.avi-8:17

代码如下：

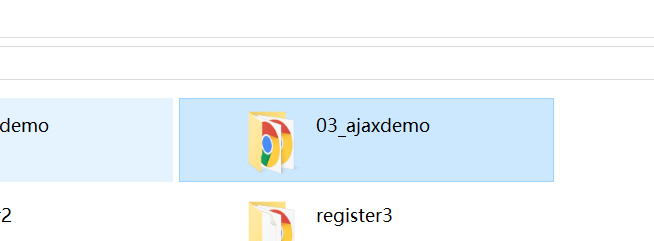


总结：

1. 这块和上面的区别是这里使用的是POST请求方法
   1. xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

## 06.Ajax的使用\_案例03\_验证手机唯一性.avi-19:04

代码如下：

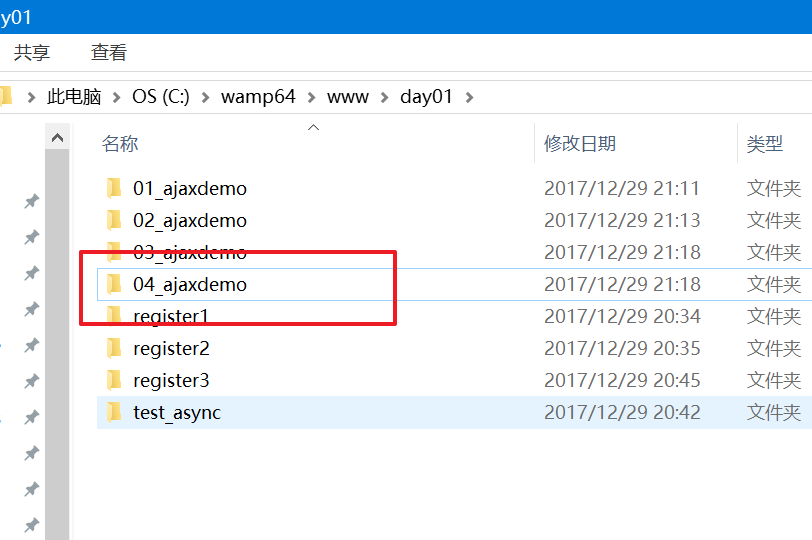


总结：

1. 这块咱们使用了一个叫JSON.parse的方法可以把json类型的数据转换成对象（后面会对json进行讲解）

## 07.Ajax的使用\_同步请求和异步请求.avi-15:18

代码：



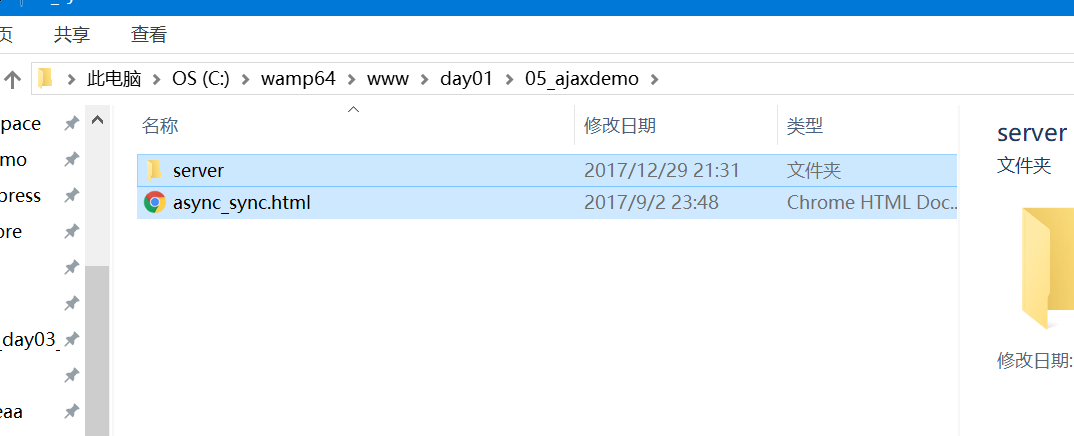
总结：



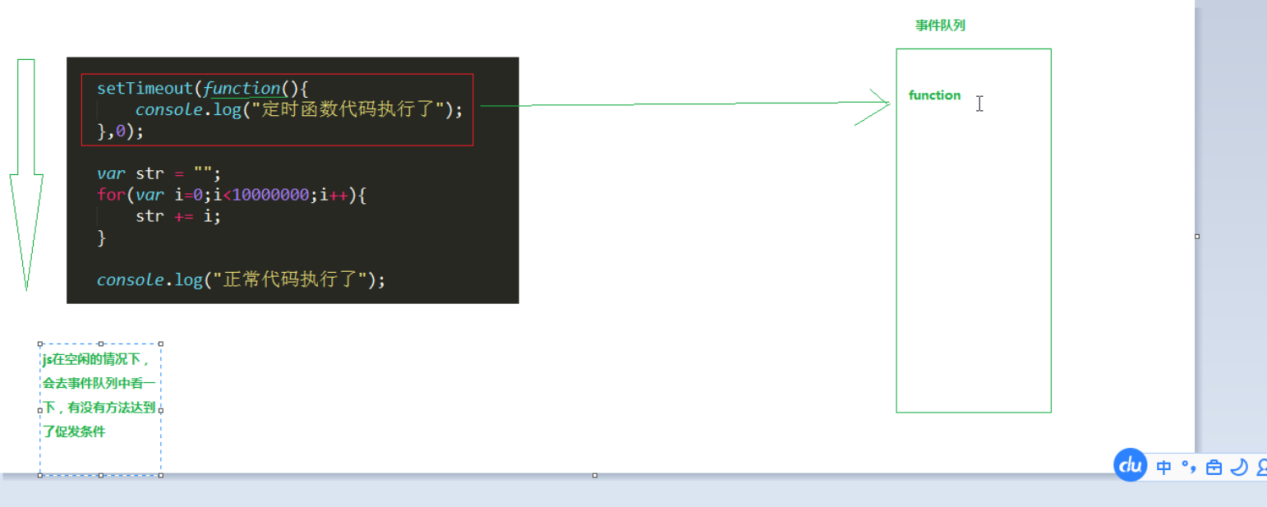
这块要注意：在php中时间的单位是秒，在js中是毫秒

## 08.Ajax的使用\_同步异步的原理.avi-22:17

代码：



总结：



## 09.Ajax的使用\_数据格式的讲解.avi-9:43

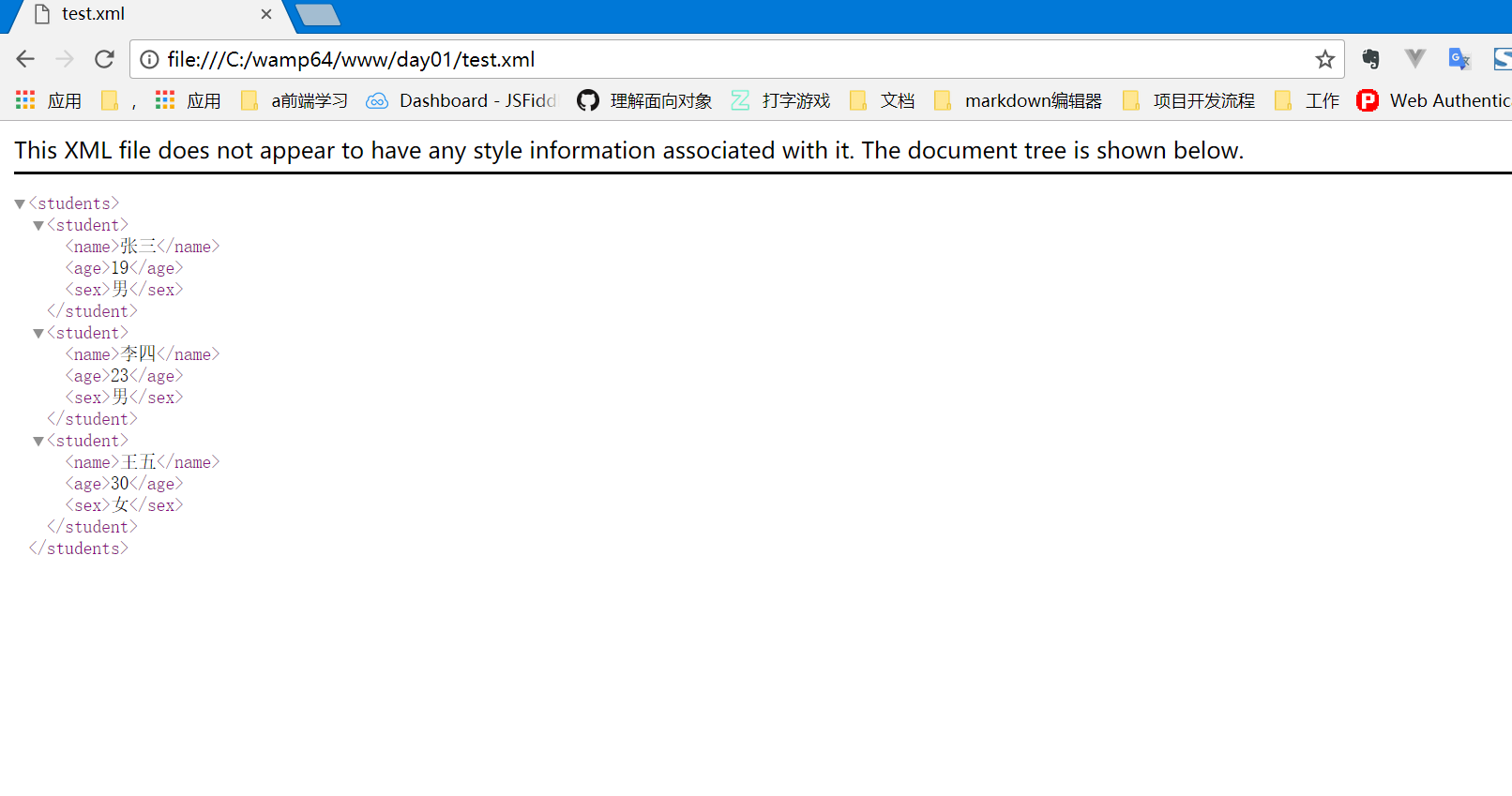
总结：

什么是数据格式

将数据通过一定的规范组织起来，叫做数据格式，例如

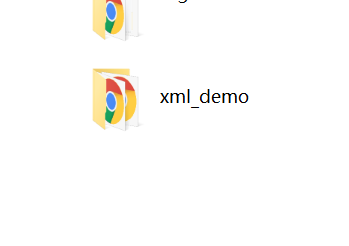
**张三%19%男-李四%23%男-王五%30%女**

这就是一种数据格式，这种数据格式组成的规则不通用。



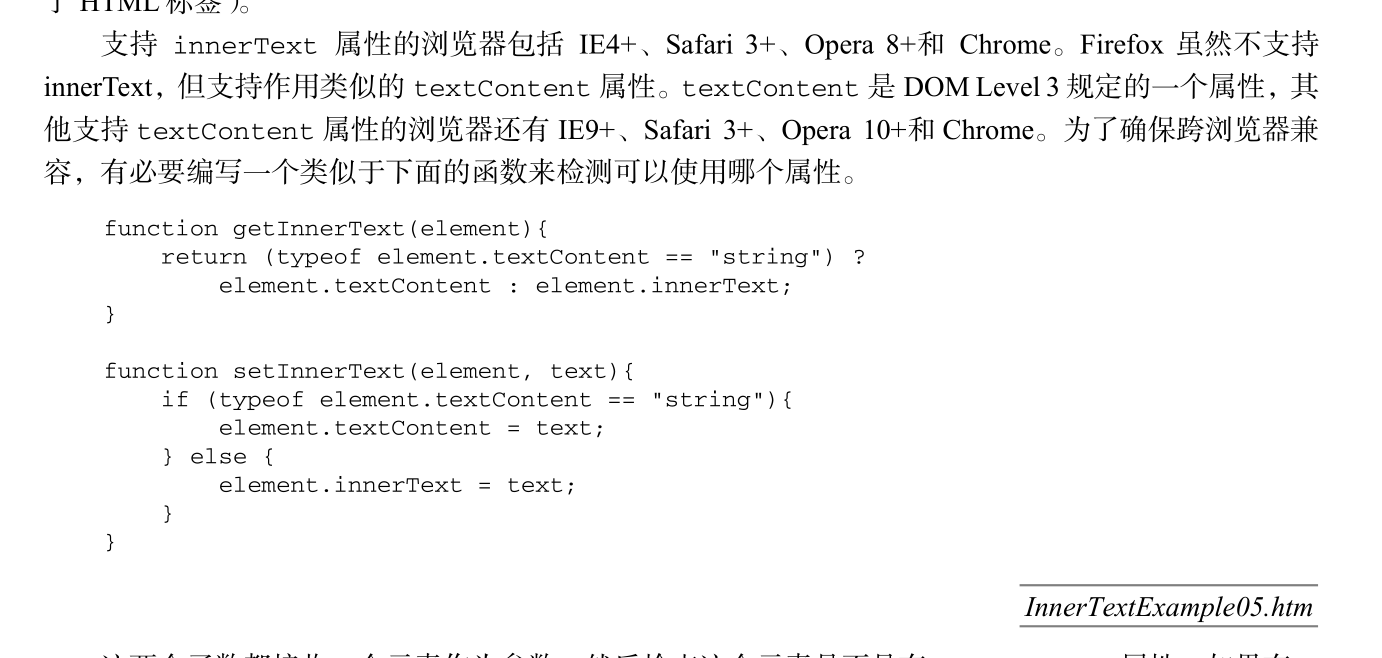
## 10.Ajax的使用\_解析xml数据格式.avi-22:32

代码：



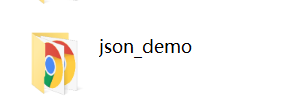
总结：

1、xml使用起来和DOM的操作是类似的



## Ajax的使用\_解析json数据格式.avi-11:22

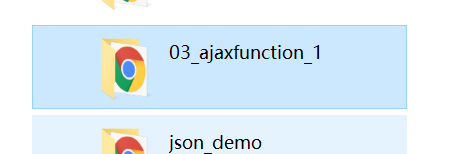
代码：



总结：

## Ajax的封装\_初步封装.avi-15:00

代码：



总结：

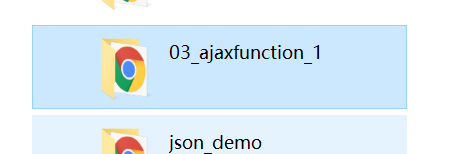
封装的思路：

1. 哪些东西是变的。
2. 哪些东西是不变的。
3. 如何将结果通知调用者。
4. 如何调用方便。

小技巧：如果大家不会封装，可以先把代码写出来，然后再看着代码思考上面的几个要点

## 13.Ajax的封装\_测试.avi-9:46

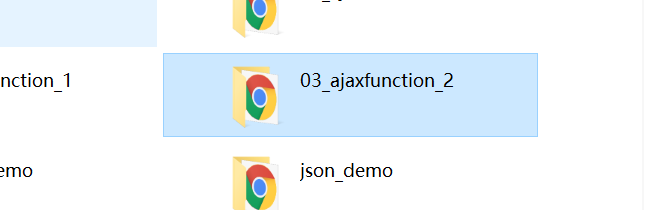
代码：



总结：

## 14.Ajax的封装\_细节完善\_考虑同步异步情况.avi-6:30

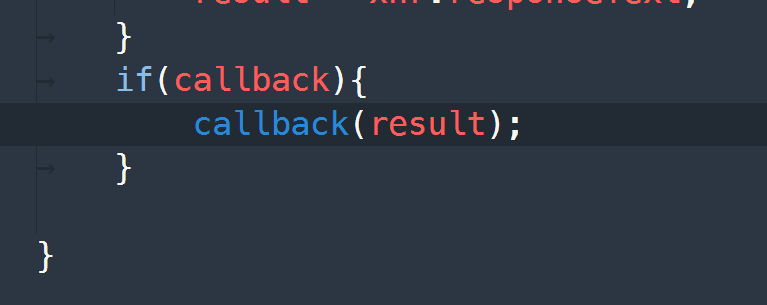
代码：



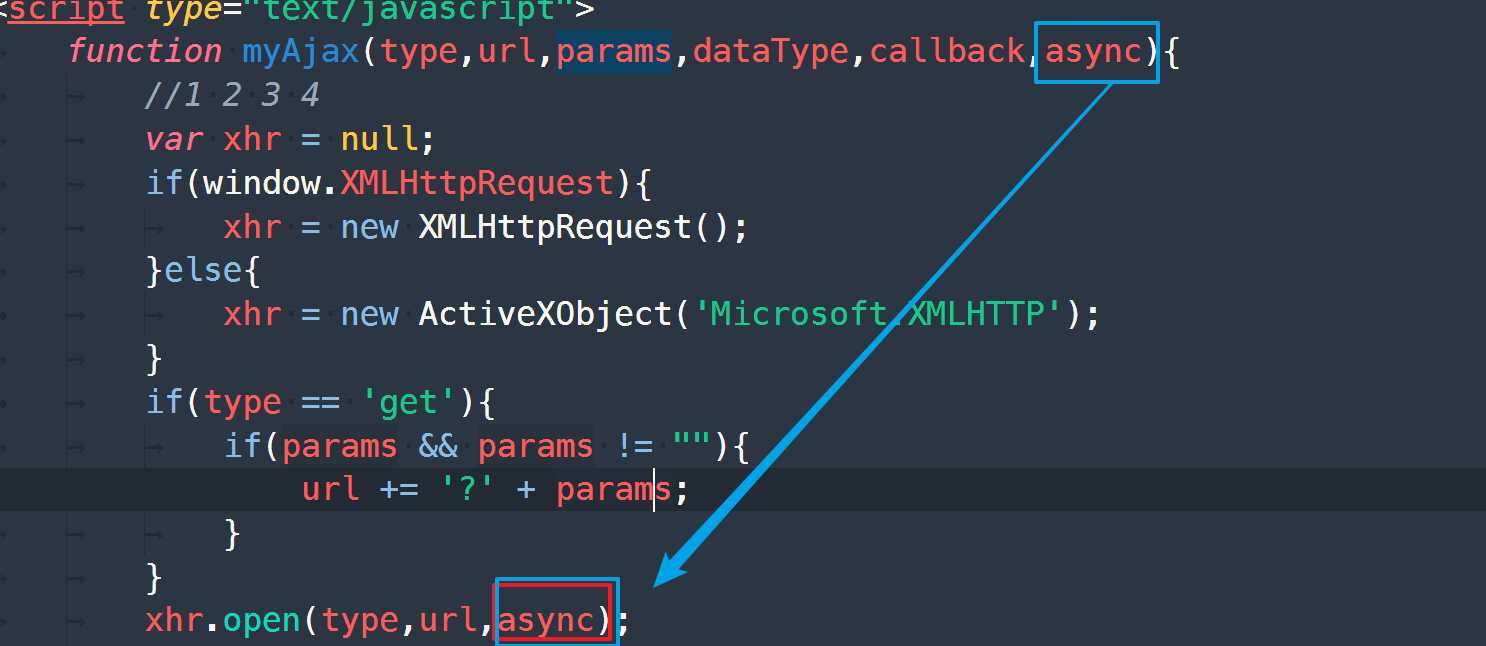
总结：

1. 防止用户输入的参数有问题，完善我们的ajax方法



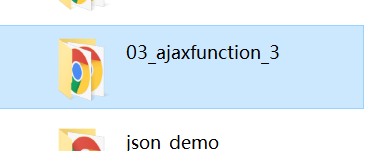


1. 考虑同步和异步两种情况



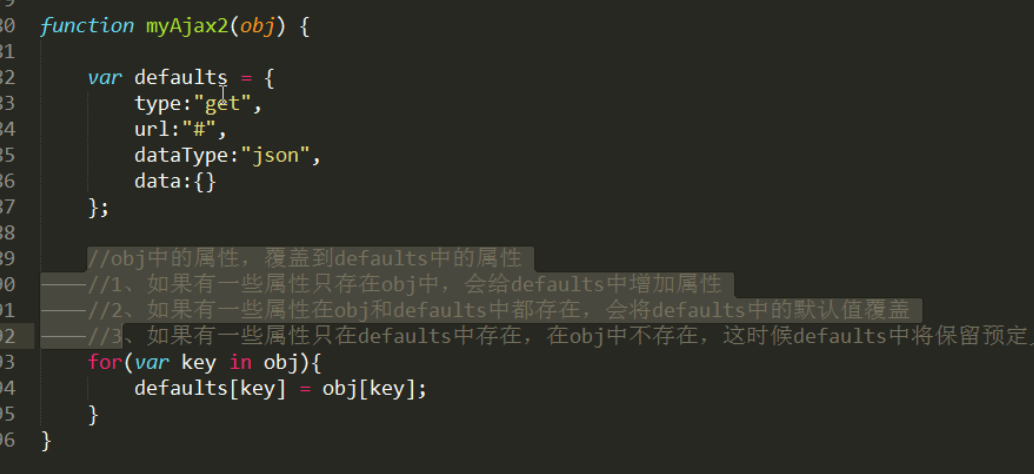
## 15.Ajax的封装\_优化封装方案分析.avi-12：03

代码：



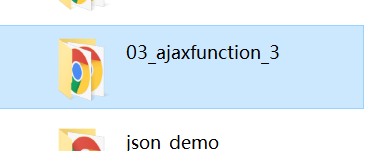
总结：

1. 参数传入太多了，而且顺序是有要求的，写起来不太方便



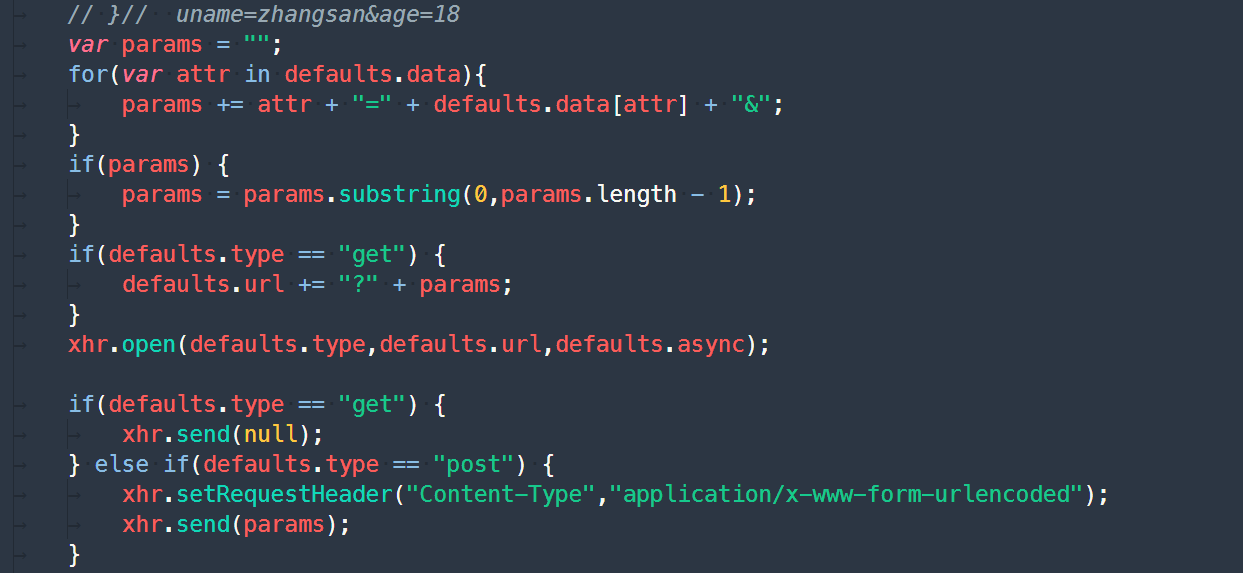
## 16.Ajax的封装\_优化封装方案实现.avi-14:18

代码：



总结：



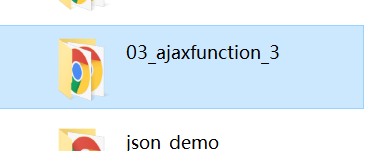






## 17.Ajax的封装\_优化封装方案测试.avi-9:09

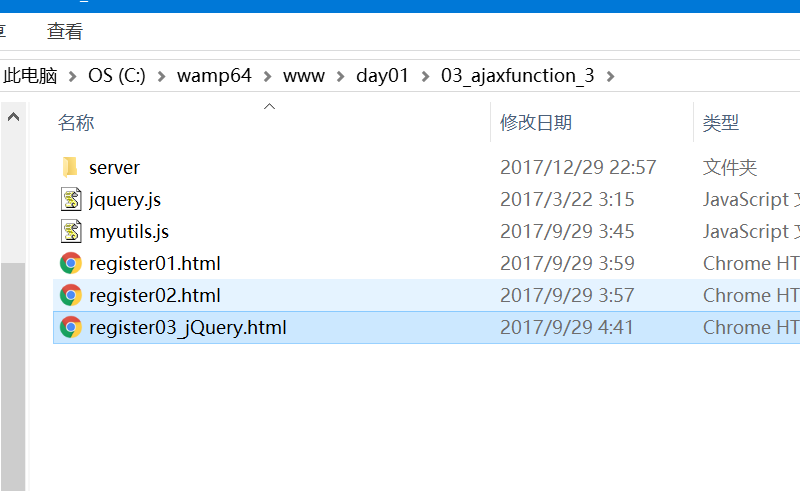
代码：



总结：

## 18.Ajax的封装\_jQuery的使用.avi-10:20

代码：

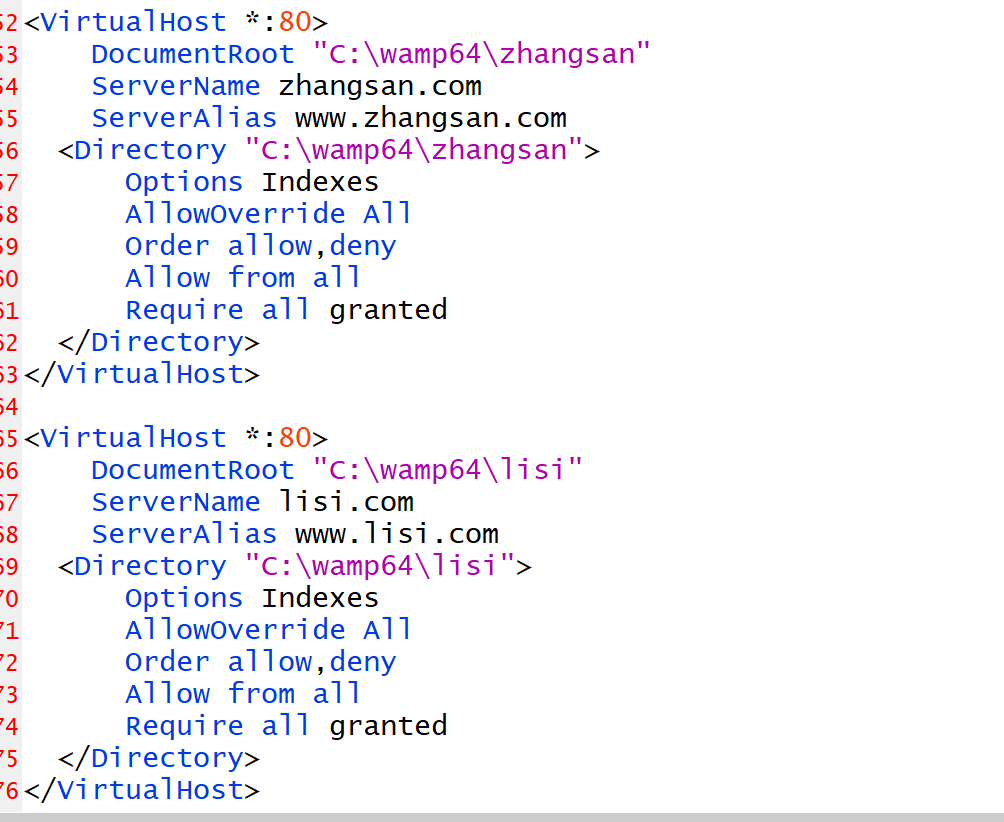


总结：

## 19.跨域\_同源策略的介绍.avi-9:43

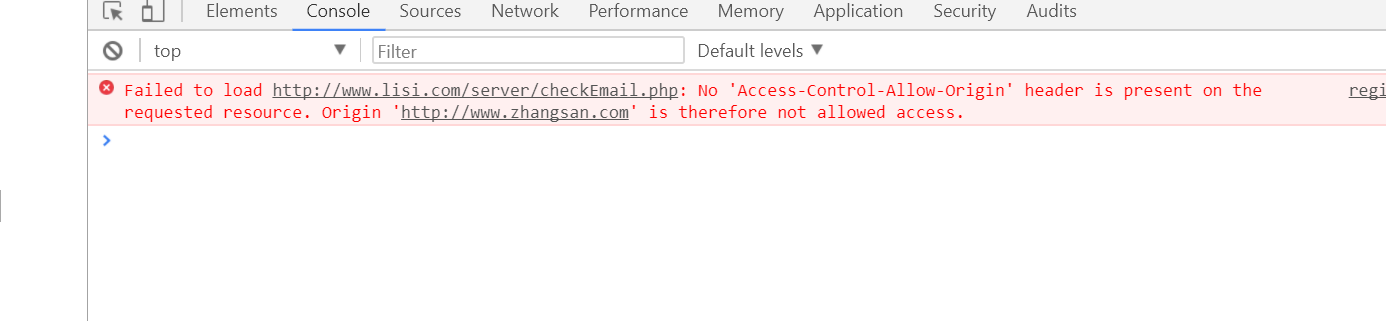
代码：

上面的例子可以不配置虚拟主机，到这一步，必须配置两个虚拟主机



总结：

1. 同源
   1. 协议
   2. 域名
   3. 端口



## 20.跨域\_学习跨域的必要性.avi-3:17

总结：

1. 有些数据我们自己是没有的，必须要跨域去获取，比如淘宝上的商品数据、天气数据，我们必须解决掉同源策略

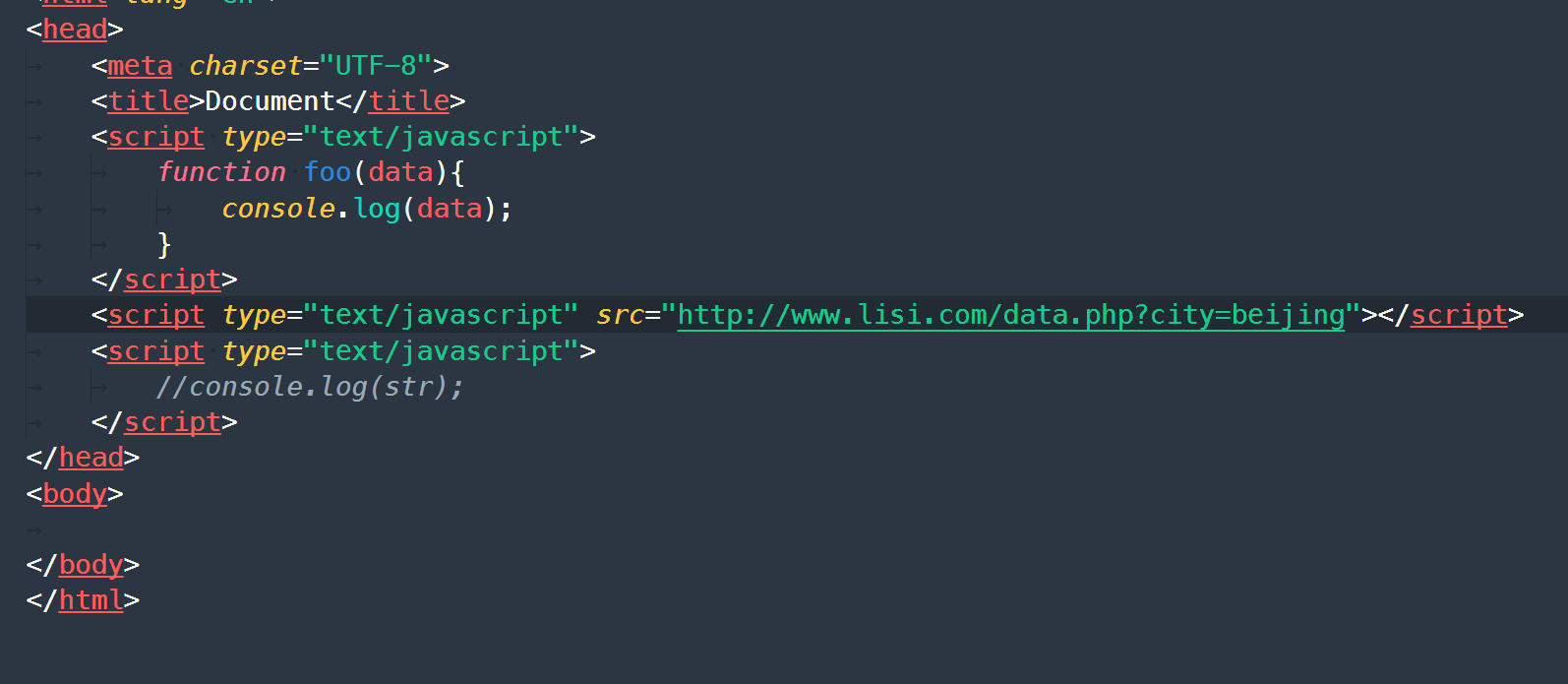
## 21.跨域\_跨域的本质\_引入外部js文件.avi-6:40

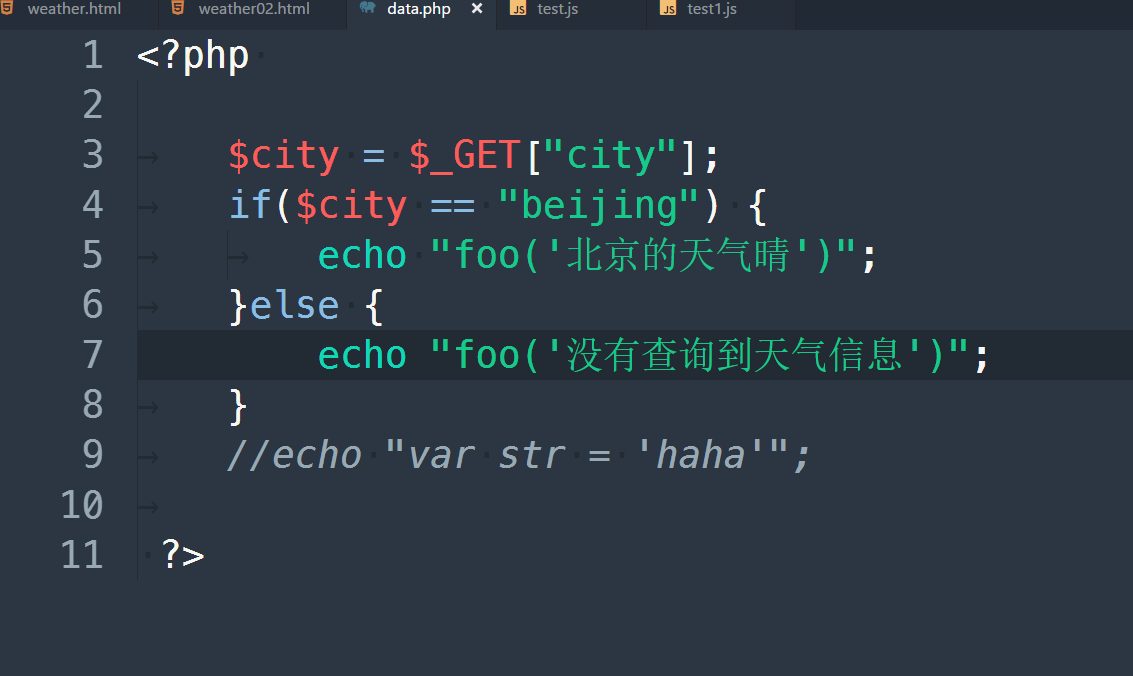
总结：

1. 跨域和ajax没有关系
2. 跨域的本质，其实就是服务器返回了一个方法调用，这个方法是我们事先定义好的，而方法中的参数就是我们想要的数据

## 22.跨域\_跨域的本质\_引入外部php文件.avi-5:54

总结：





过渡：

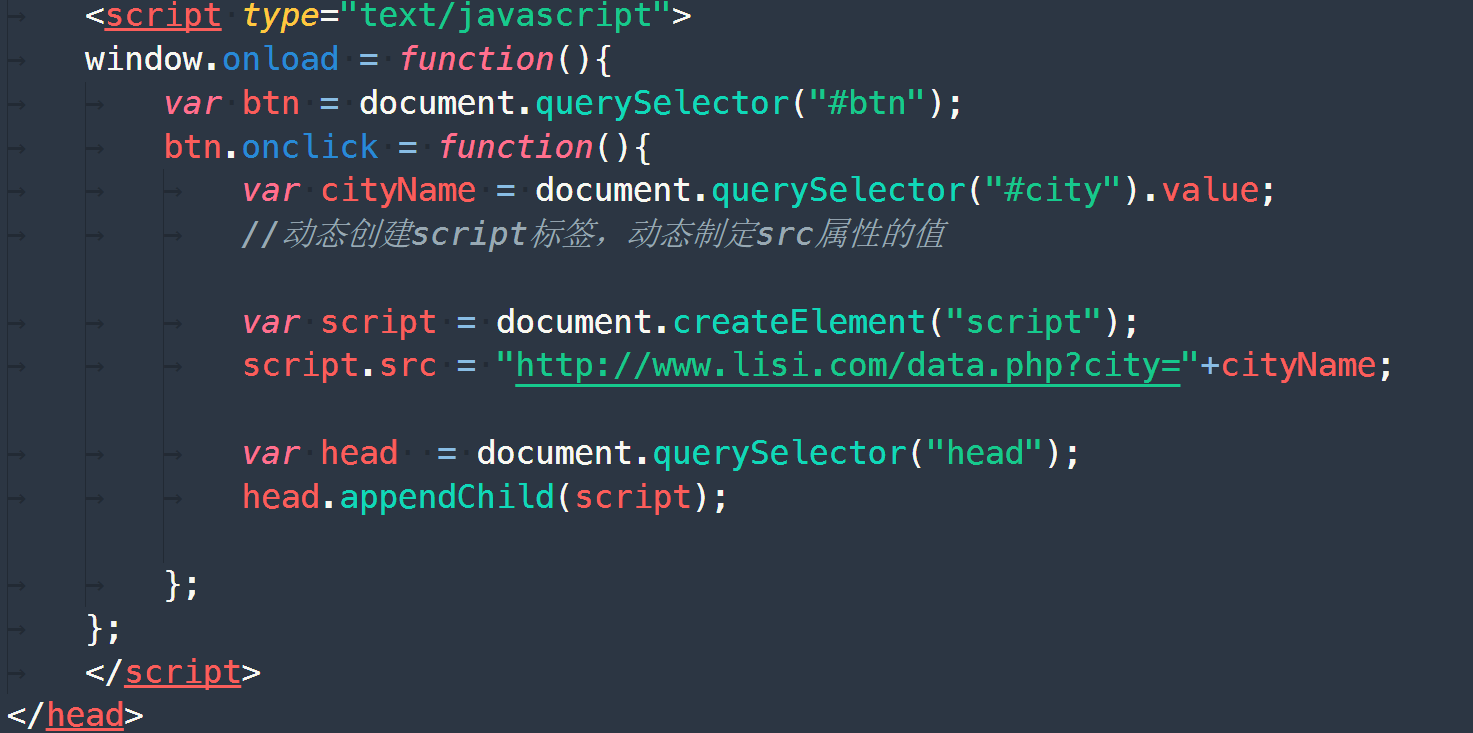
我们上面的script标签是写死的，意味着要的数据是死的，但是如果我们查询的城市事先是不确定的，由用户输入进行查询的，那怎么来做呢？



---> 既然查询的天气是动态的，script标签的地下后面的参数是不确定的 --> script标签是动态创建出来的：

## 23.跨域\_跨域的本质\_动态创建script标签.avi-6:45

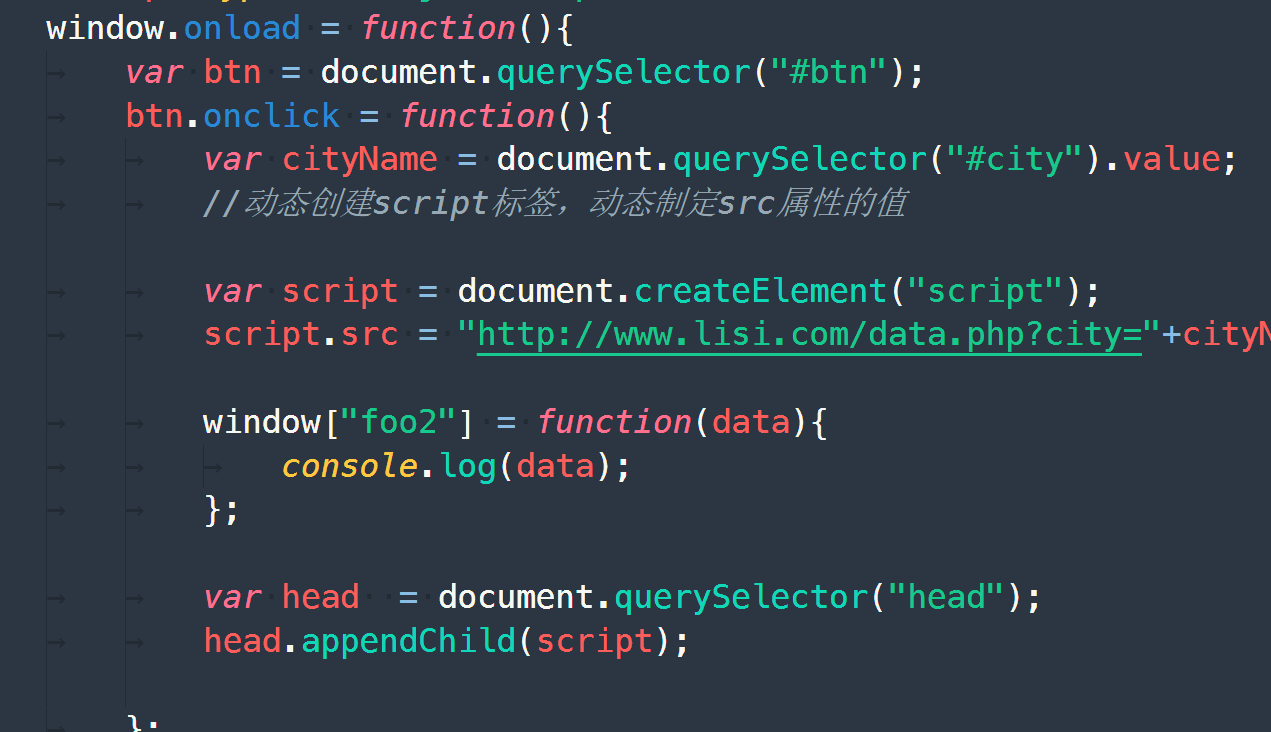
总结：



## 24.跨域\_跨域的本质\_动态指定回调函数名称.avi-7:47

总结：

1. 函数名不能写死
2. 函数创建使用window[‘函数名’]



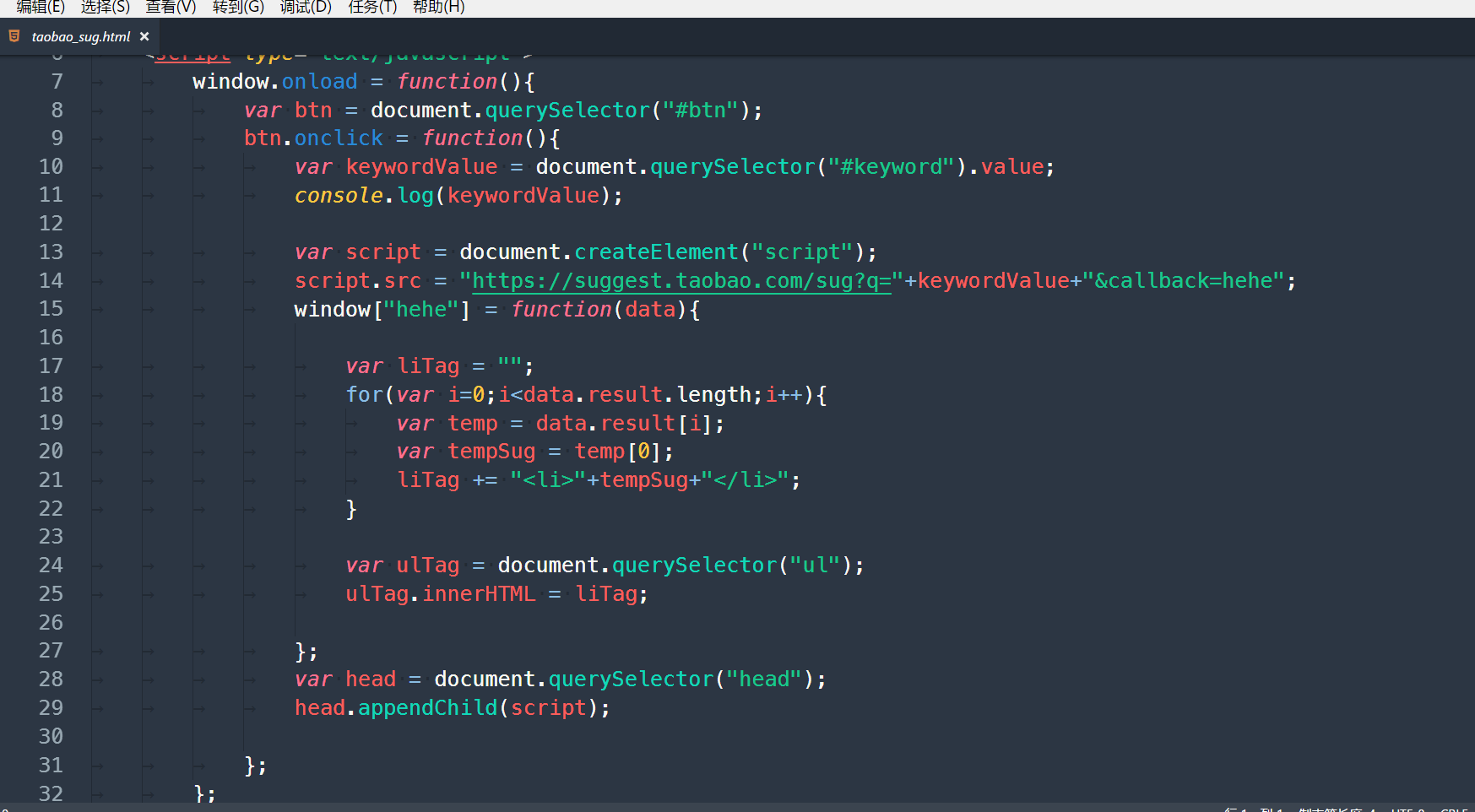
## 25.跨域\_淘宝提示词案例.avi-19:09

总结：

1. 总结解决跨域的思路
   1. 访问外部的js文件不会导致同源策略
   2. 访问php文件实现数据不写死
   3. 动态创建script标签传入动态参数
   4. 由前端决定函数名
   5. 使用window[函数名]来创建函数







## 26.跨域\_百度提示词案例.avi-5:23

总结：

1. 这个例子是让大家意识到，不同的接口文档使用方法是不一样的

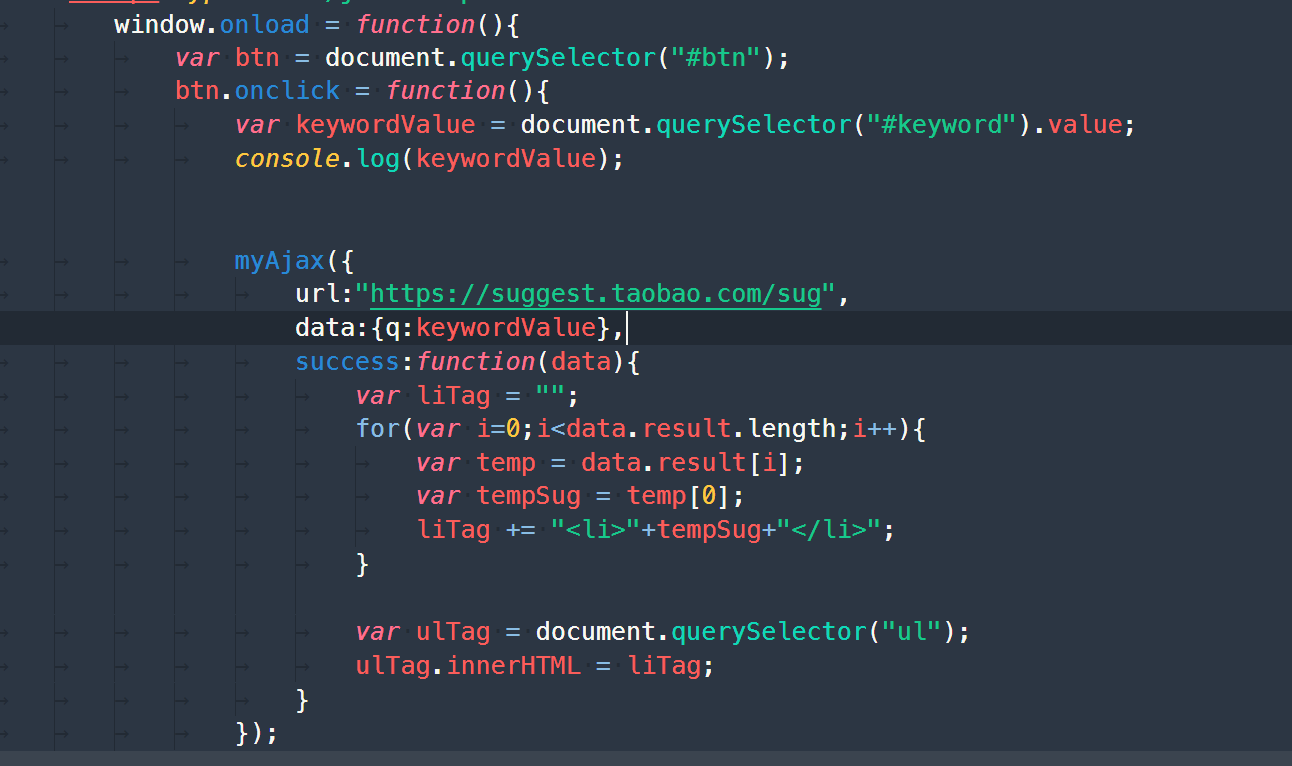


## 27.跨域\_封装01.avi-15:38-改造淘宝案例

总结：

封装的思路：

1. 哪些东西是变的。
2. 哪些东西是不变的。
3. 如何将结果通知调用者。
4. 如何调用方便。



## 28.跨域\_封装02.avi-10:33-改造百度案例

总结：

## 29.跨域\_使用jquery获取跨域数据.avi-12:26

总结：

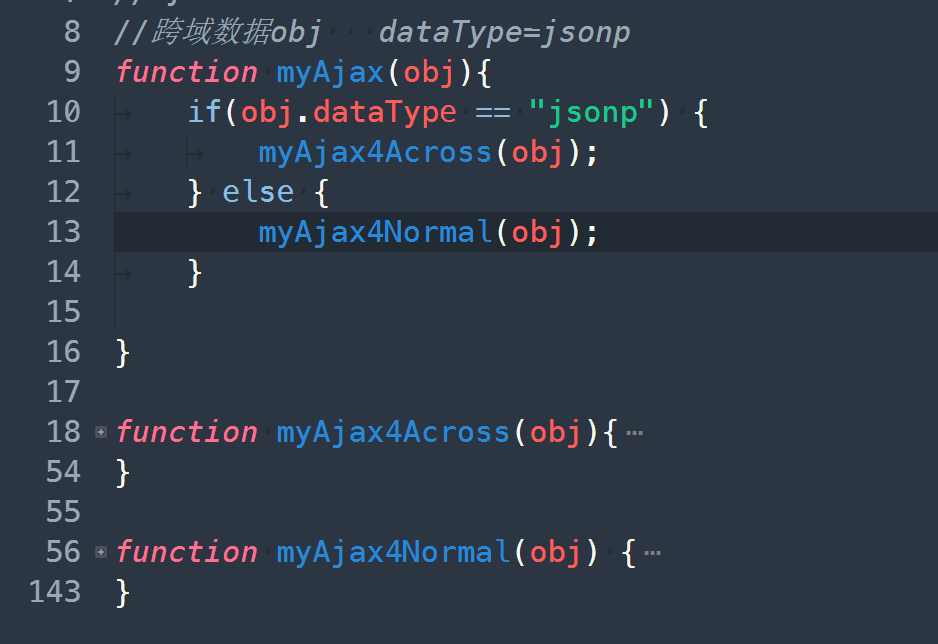


思考：

1. 如何看待jquery的参数形式
   1. su?cb=jQuery111106839969638488013\_1514611580138&wd=java&\_=1514611580139
   2. 随机性，保证不会造成全局污染

## 30.跨域\_完善myAjax方法达到能获取同源数据和非同源数据.avi-4:00

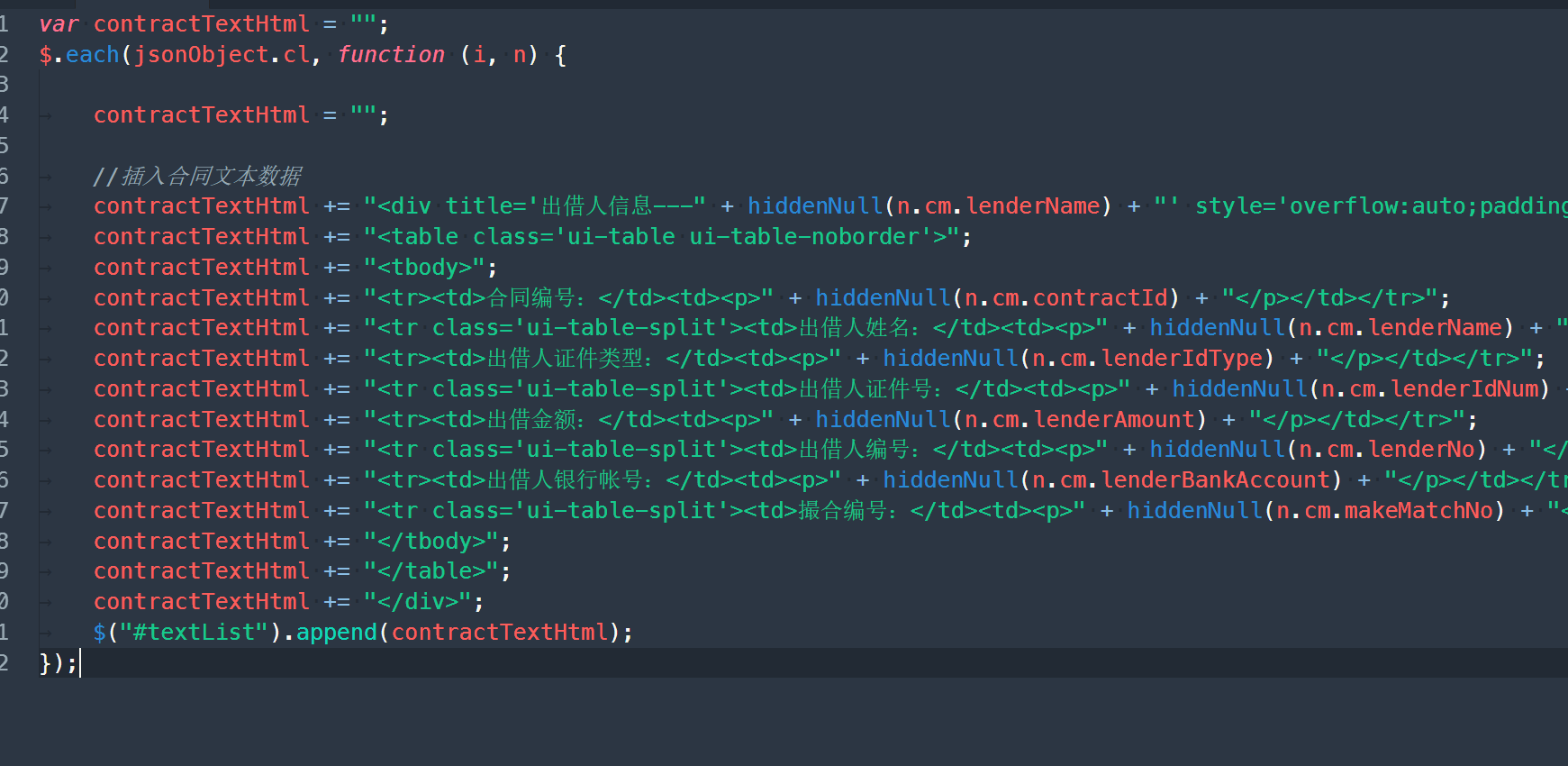
总结：



## 31.模板引擎的使用\_改造百度提示案例.avi-19:03

总结：

1. 模板引擎解决的问题



如果不用的话，字符串拼接一多会很丑

## 32.模板引擎的使用\_artTemplate的常用语法.avi-18:06

总结：