Ajax

1. 介绍
2. web开发相关技术

html

css

javascript

客户端语言

php

jsp

asp

服务器端语言

1. 什么是ajax

* Asynchronous ：异步
* JavaScript
* And ：和
* XML

ajax技术就是利用javascript和xml实现异步交互的功能

javascript

json

xml

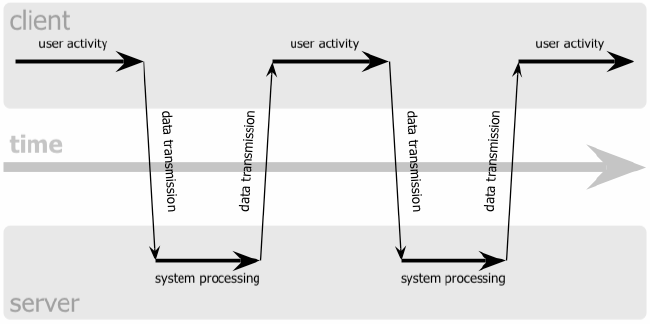
发展历程

* 1998年
* 2005年

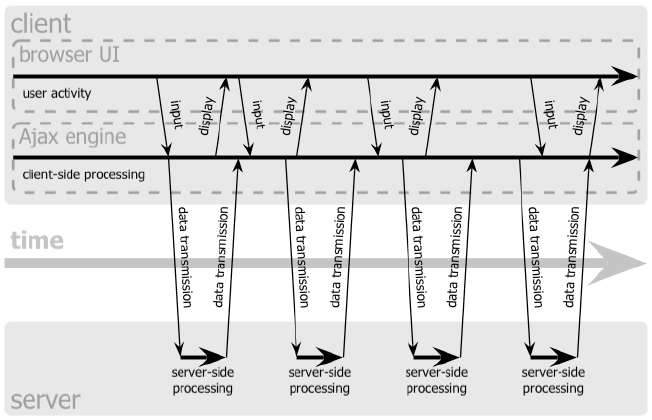
运行平台

* Google Chrome
* Mozilla
* Firefox
* Internet Explorer
* Opera
* Konqueror
* Safari

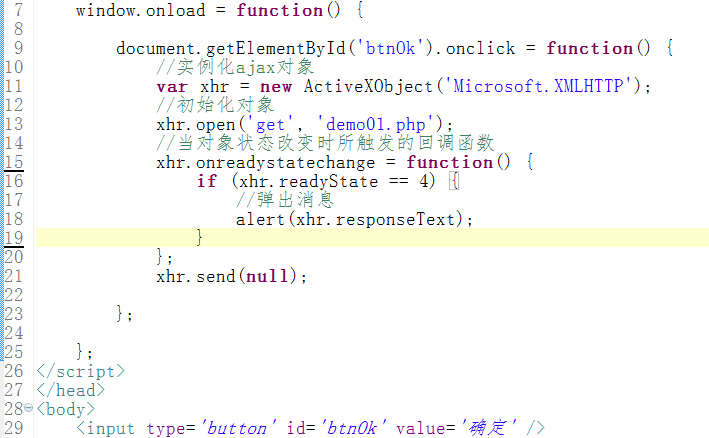
1. ajax运行原理
2. 传统的web应用程序运行原理：



1. ajax程序运行原理：



1. 快速入门



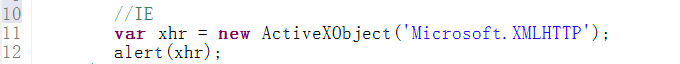
1. Ajax对象

XMLHttpRequest

必须要创建一个对象

1. 创建ajax对象的方式

1)



2)



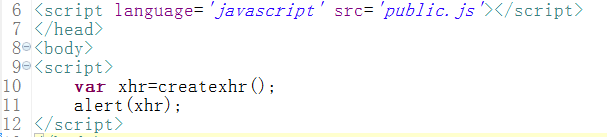
在IE的高版本中，已经支持了XMLHttpRequest类

1. 解决兼容性

创建公共文件：public.js



在需要使用ajax对象的页面中，包含以上文件



1. ajax对象的相关属性和方法

* open(method,url)

初始化ajax对象

method：请求方式 get、post

url：请求地址

* setRequestHeader(header,value)

设置请求头信息

header：请求头名称

value：请求头的信息

* send(content)

发送请求

只有当ajax对象的send方法被调用时，才会发送请求

content：post请求时所传递的数据

get请求时这里直接设置为“null”

* onreadystatechange

表示当ajax对象状态码发生改变时所触发的回调函数

它的值是一个函数首地址（匿名函数）

xhr.onreadystatechange=function(){

};

或

xhr.onreadystatechange=display;

* readyState

ajax对象的状态码

状态码就是一个数字 0—4

0：表示对象已建立，但未初始化

1：表示对象已初始化，但未发送

2：已调用send方法进行请求

3：正在接收数据（接收到一部分）

4：接收完成

* status

ajax对象接收到的http响应状态码 如：200 302 304 404

* statusText

ajax对象接收到的http响应状态文本 如：ok、not found

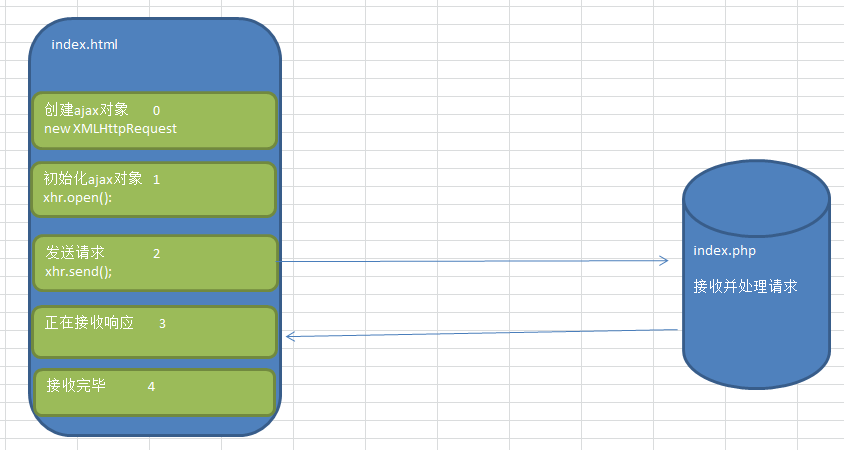
* reponseText

ajax对象接收到http响应主体字符串 （text/html）

* responseXML

ajax对象接收到的http响应主体内容（text/xml）

关于状态码解释：



关于状态码案例：



1. 发送GET请求
2. 向服务器发送用户名，并返回hello,zhangsan!

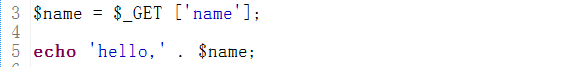
html：



if(xhr.readyState==4)

ajax对象状态每次发生改变时，都会触发生这个回调函数，只有当状态码为4时，也就是说数据才真正接收完毕，这时我们再对返回来的数据进行处理。

php：



return ‘hello’.$name;

return ：返回，将结果返回给php程序本身

echo ‘hello’.$name;

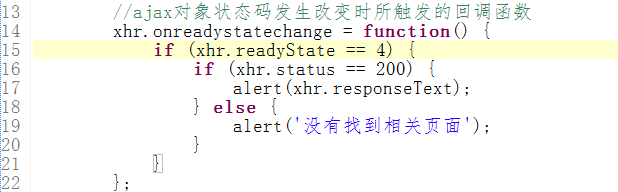
echo ：输出，利用http协议将数据响应给客户端



1. 找出上题的不足之处

1）请求错误的页面

如果将请求地址改为一个不存在的页面地址，那么服务器仍然会返回一个错误信息，而我们的程序应该在得到一个正确的返回结果后才去对数据进行处理。

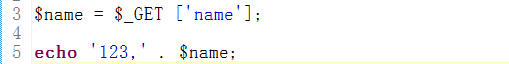


或



1. 解决IE下缓存问题

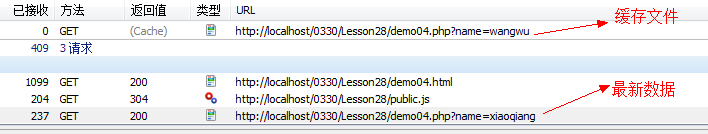
将服务器端的PHP程序略作修改



在IE下，仍然输出hello,zhangsan

在其它浏览器中，是正常输出

原因：在IE中，默认有缓存功能，将每次获取的php文件的输出结果缓存下来，下次ajax对象在请求时，如果在缓存目录下，找到对应缓存文件，就直接使用缓存文件。



如何解决？

* 随机数

Math.random();

* 时间

new Date.getTime();

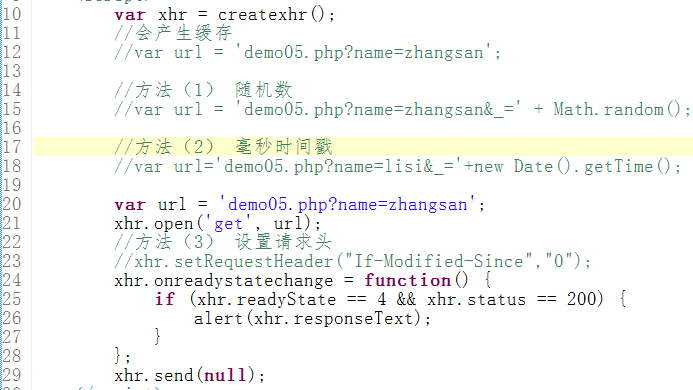
以上两种方法确保每次请求的url是唯一的

* setRequestHeader("If-Modified-Since","0");

这种方法设置ajax对象的请求头 if-modified-since，强制让ajax对象发送请求

0：表示文件最后修改时间，会和服务器上这个资源文件最后修改时间进行比较，肯定是不相，所以服务器返回了最新数据。

以上三种方式并没有根本上解决缓存问题，前二种方式更是缓存下来N个文件。



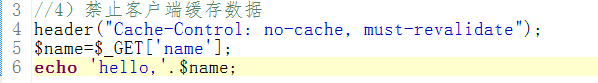
* header("Cache-Control: no-cache, must-revalidate");

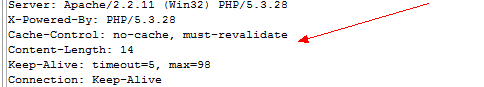
header：向http响应头中写数据

header(‘location:index.php’);

设置http响应头中的cache-control选项

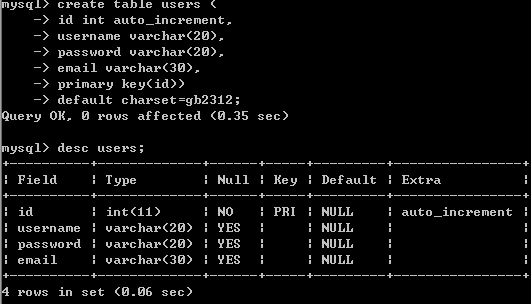
“告诉”浏览器：你不要缓存，必须每次重新请求。

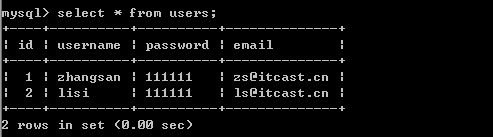




1. 检查用户名是否可用

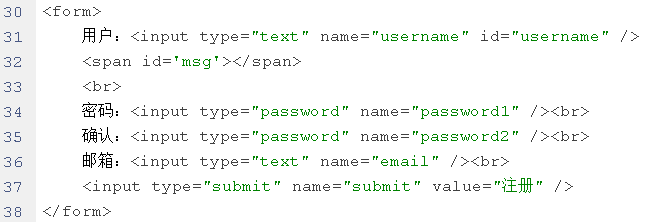






html：

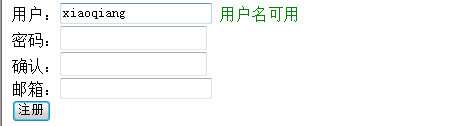




php：

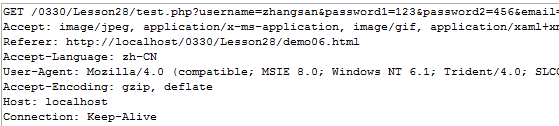


运行结果：

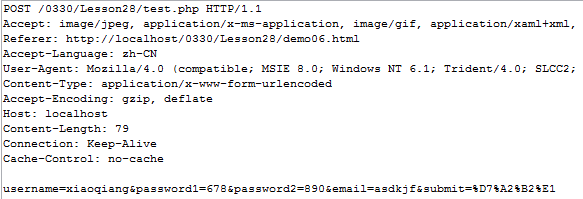


1. 发送post请求
2. get和post的区别：
3. get请求将参数放到请求地址url的后面
4. post请求时将参数放在http请求空白行的后面
5. get请求时参数大小有限制
6. post请求理论上对参数大小无限制
7. post比get安全一些
8. 其它不同：

get请求：



post请求：



通过比较，可以看出，post请求时，除了参数格式不同之外，还比get请求多了一个Content-type的请求头，它的值是application/x-www-form-urlencoded，表示本次提交的数据是字符数据，同时post还可以同时提交二进制数据和字符数据，如：multipart/form-data

ajax发送请求其实就是模拟http请求

ajax对象的post请求也要加上content-type的请求头。

1. 代码：

* xmlhttp.open("post",“action.php");

post

action.php 后面没有参数

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

设置请求头信息

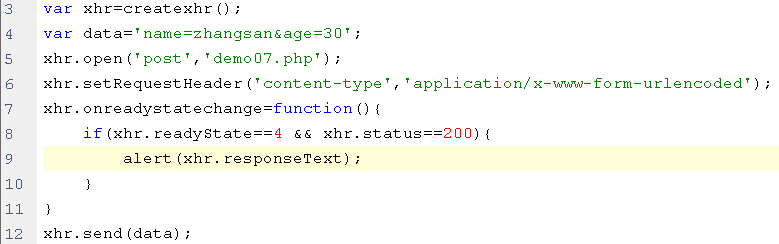
content-type：传送数据的数据类型

application/x-www-form-urlencoded ：表示数据是字符数据

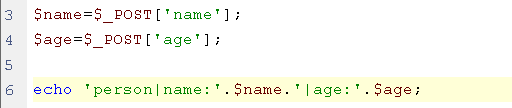
* xmlhttp.send(data);

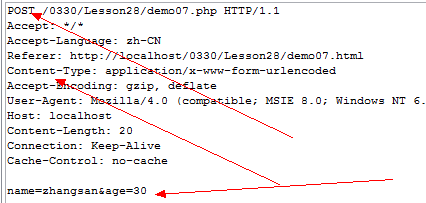
data：会自动将参数放到请求空白行的后面

html代码：



php代码：



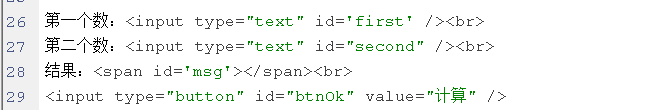


通过测试，发现ajax对象的post请求不会产生缓存问题

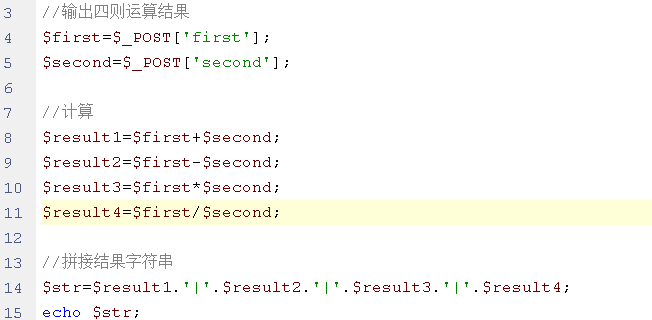
1. 计算两个数的和

html：

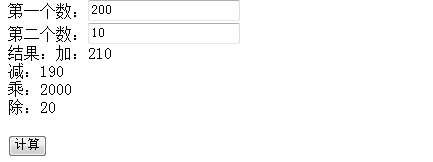




php：



运行结果：



如果需要从服务器端返回多个结果，可以将结果拼接一个字符串，使用一个指定的分隔符，如：“|”，在客户端程序中，再将字符串按照分隔符进行分割。

1. 文件上传

* method=post

以 post形式提交数据

* enctype="multipart/form-data"

指定提交的数据可以是二进制数据或字符数据

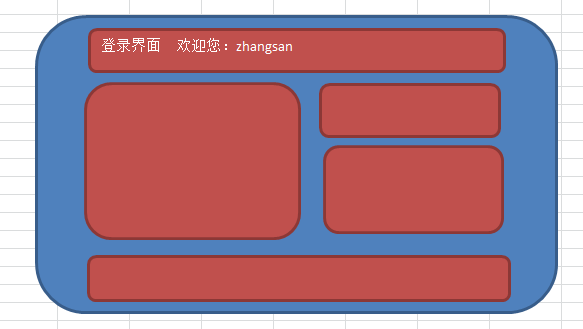
demo09.html

demo09.php

upload.php

五、关于Ajax的一点补充

无刷新是Ajax最大的特点，但不是Ajax技术出现的目的



如果使用传统的web应用程序，用户在登录时，整个页面重新刷新并请求新的页面地址，新的页面在验证之后，再重新跳转回来，但是对用户而言，除了登录界面之外，其它版本没有发生改变

使用ajax程序，可以异步发送请求，改变的仅仅是登录界面，其它版本没有重新请求和刷求，所以节省网络传输的流量。可以快速获取服务器端数据