jQuery框架

目录

[jQuery框架 1](#_Toc58691195)

[一、each方法 2](#_Toc58691196)

[1、回顾jQuery对象 2](#_Toc58691197)

[2、each方法功能 2](#_Toc58691198)

[3、each相关案例 2](#_Toc58691199)

[二、jQuery中的Ajax技术 3](#_Toc58691200)

[1、回顾Ajax技术 3](#_Toc58691201)

[2、jQuery中的Ajax 4](#_Toc58691202)

[3、底层应用$.ajax 4](#_Toc58691203)

[4、解决$.ajax的缓存问题（Ajax中get请求在IE浏览器） 5](#_Toc58691204)

[5、$.ajax中的post请求与传值 6](#_Toc58691205)

[6、$.ajax中的dataType属性 7](#_Toc58691206)

[7、jQuery中Ajax的高级实现 10](#_Toc58691207)

[8、解决Ajax中get请求的缓存问题 10](#_Toc58691208)

[9、高级应用$.post 11](#_Toc58691209)

[10、$.get与$.post中期待的返回值类型 12](#_Toc58691210)

[三、jQuery课程总结 14](#_Toc58691211)

[1、each方法 14](#_Toc58691212)

[2、jQuery中的Ajax技术 14](#_Toc58691213)

# 一、each方法

## 1、回顾jQuery对象

jQuery对象是一个类数组的特殊对象，其每个数组元素又是一个DOM对象，所以jQuery对象是DOM对象的集合。

## 2、each方法功能

在jQuery中，each方法主要用于遍历jQuery对象，基本语法：

jQuery对象.each(callback);

callback是一个回调函数，其结构如下：

function callback(i,item) {

参数说明：

i ：每次遍历时，系统会自动将得到的索引下标放入变量i中

item ：每次遍历时，系统会自动将遍历得到的结果放入变量item中

总结：jQuery对象是一个DOM对象的集合，每次遍历时，其得到的结构就是一个DOM对象，所以item其实一个是DOM对象。

}

## 3、each相关案例

例1：对页面中的所有div元素进行遍历，并设置其值



例2：使用each方法对页面中的所有img进行遍历，并设置其src属性



# 二、jQuery中的Ajax技术

## 1、回顾Ajax技术

① Ajax中的get请求 ② Ajax中的post请求

① Ajax中的get请求

五步走：

第一步：创建Ajax对象

var xhr = XMLHttpRequest();

第二步：设置回调函数

xhr.onreadystatechange = function() {

//第五步：判断与执行

if(xhr.readyState==4 && xhr.status==200) {

//返回结果

文本形式：xhr.responseText;

XML形式：xhr.responseXML;

}

}

第三步：初始化Ajax对象

xhr.open(‘get’,’deal.php’);

第四步：发送Ajax请求

xhr.send(null);

② Ajax中的post请求

六步走：

第一步：创建Ajax对象

var xhr = XMLHttpRequest();

第二步：设置回调函数

xhr.onreadystatechange = function(){

//第六步：判断与执行

if(xhr.readyState==4 && xhr.status==200) {

文本形式：xhr.responseText;

XML形式：xhr.responseXML;

}

}

第三步：初始化Ajax对象

xhr.open(‘post’,’deal.php’);

第四步：设置请求头信息

xhr.setRequestHeader(‘Content-type’,’application/x-www-form-urlencoded’);

第五步：发送Ajax请求

xhr.send(data);

## 2、jQuery中的Ajax

底层应用：

$.ajax

高级应用：

$.get

$.post

## 3、底层应用$.ajax

基本语法：

jQuery.ajax(options)

参数说明：

options ：只有一个参数，要求是JSON格式的数据，其可以设置如下属性：

async ：是否异步，true代表异步，false代表同步。默认为true

cache ：是否缓存，true代表缓存，false代表不缓存，默认为true

complete ：当Ajax状态码为4时所触发的回调函数

contentType ：请求头，如果是POST请求，此参数为application/x-www-form-urlencoded

data : 发送Ajax请求时所传递的参数，要求是一个字符串

dataType ：期待的返回值类型，可以是text/xml/json数据类型

success ：当Ajax状态码为4且响应状态码为200时所触发的回调函数

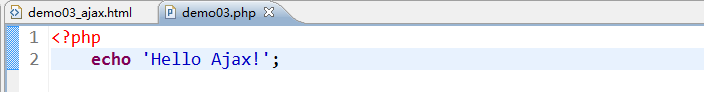
type ：发送的http请求，可以是get，也可以是post

url ：请求的url地址

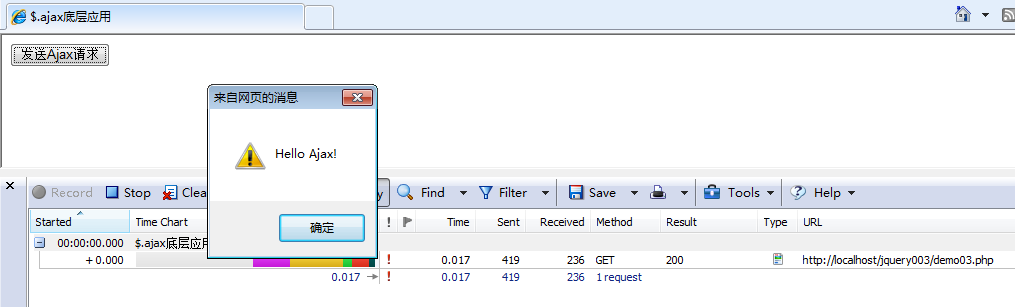
demo03\_ajax.html 示例代码：



demo03.php 示例代码：



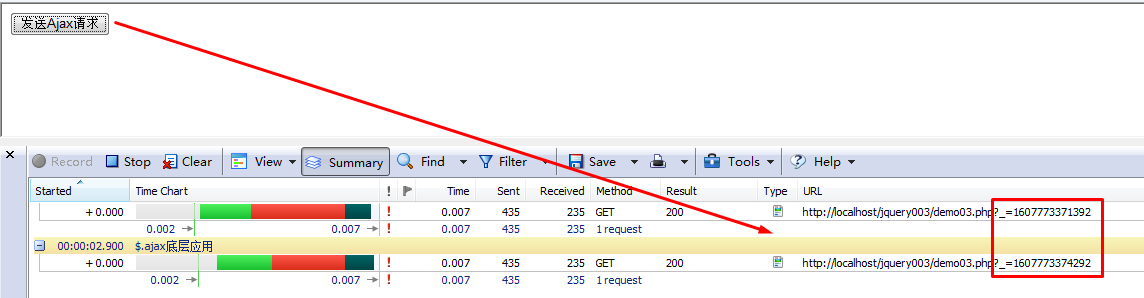
运行结果：



## 4、解决$.ajax的缓存问题（Ajax中get请求在IE浏览器）



运行结果：



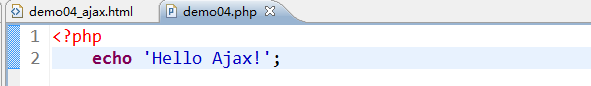
由上图可知，jQuery中是通过时间戳的形式来解决的Ajax缓存问题。

## 5、$.ajax中的post请求与传值

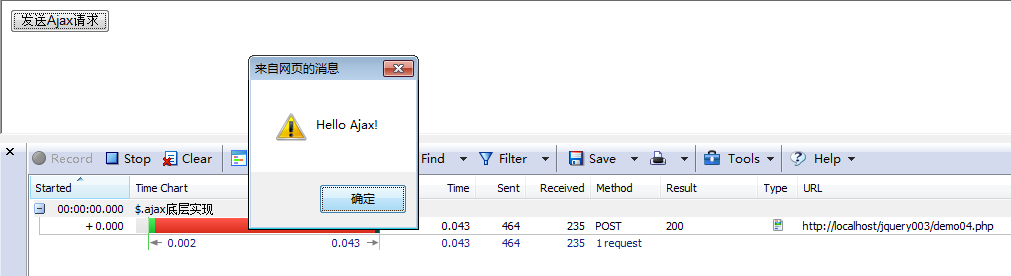
demo04\_ajax.html 示例代码：



demo04.php 示例代码：

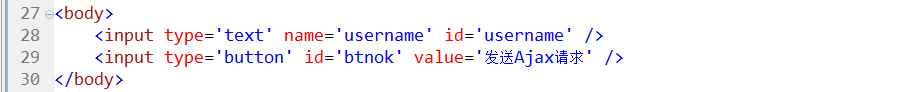


运行结果：

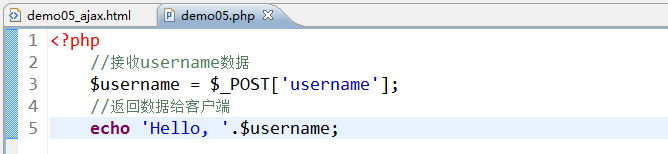


demo05\_ajax.html 示例代码：

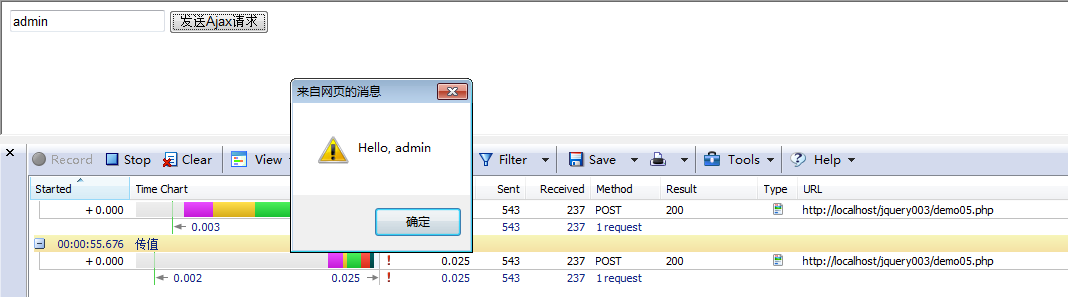




demo05.php 示例代码：



运行结果：



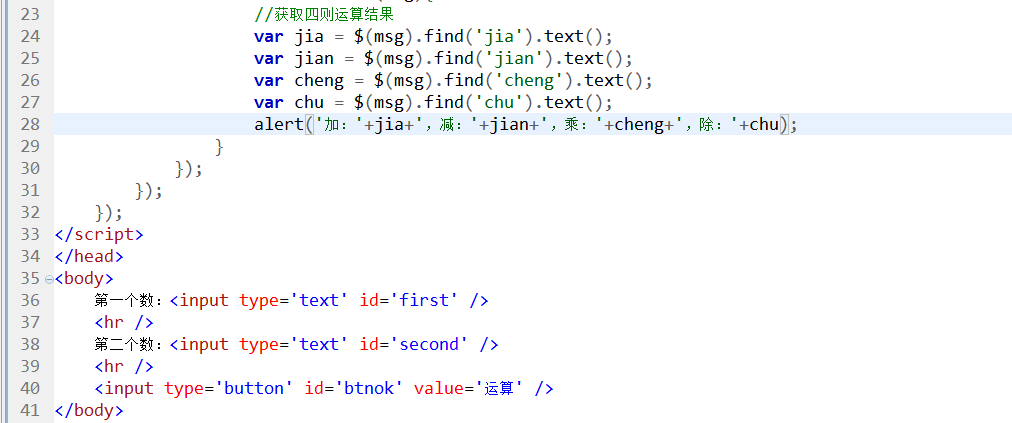
## 6、$.ajax中的dataType属性

dataType ：期待的返回值类型，可以text/xml/json，默认为text文本格式。

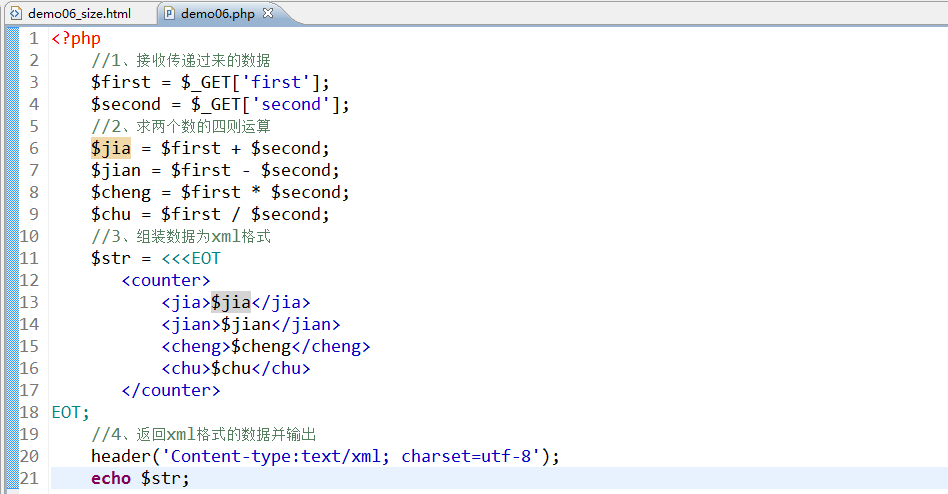
① 使用dataType返回xml格式的数据（案例：使用ajax+xml返回两个数的四则运算）

demo06\_size.html 示例代码：



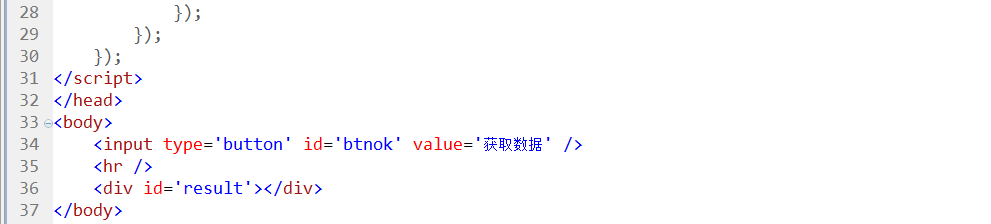


demo06.php 示例代码：



② 使用dataType返回json格式的数据（案例：使用ajax+json实时返回数据表的信息）

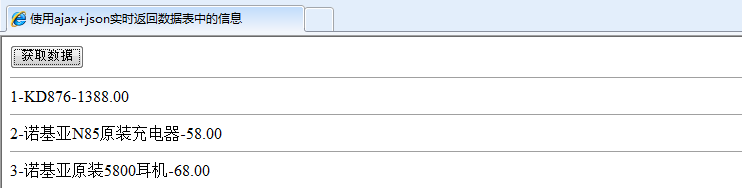




demo07.php 示例代码：



运行结果：



## 7、jQuery中Ajax的高级实现

* $.get(url,[data],[callback],[type]) ：发送Ajax中的get请求
* $.post(url,[data],[callback],[type]) ：发送Ajax中的post请求

参数说明：

url ：请求的url地址

[data] ：可选参数，代表发送Ajax请求时所传递的数据，可以是字符串格式也可以是json格式

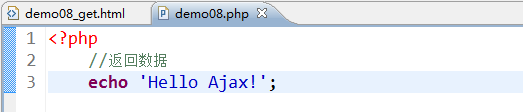
[callback] ：可选参数，代表当Ajax状态码为4且响应状态码为200时所触发的回调函数

[type] ：可选参数，代表期待的返回值类型，可以是text/xml/json，默认为text文本类型

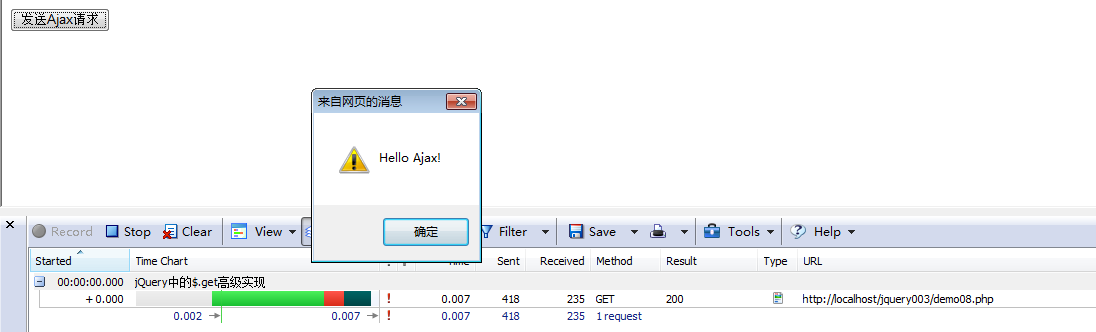
demo08\_get.html 示例代码：



demo08.php 示例代码：

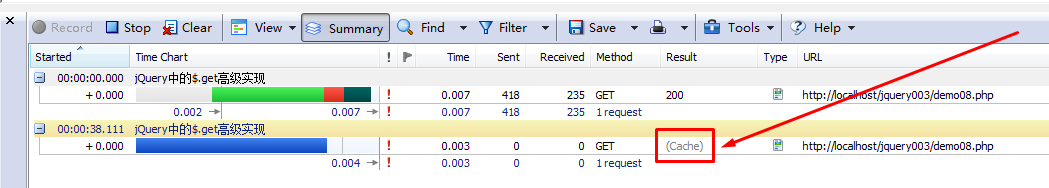


运行结果：



## 8、解决Ajax中get请求的缓存问题

再次运行以上案例，如下图所示：

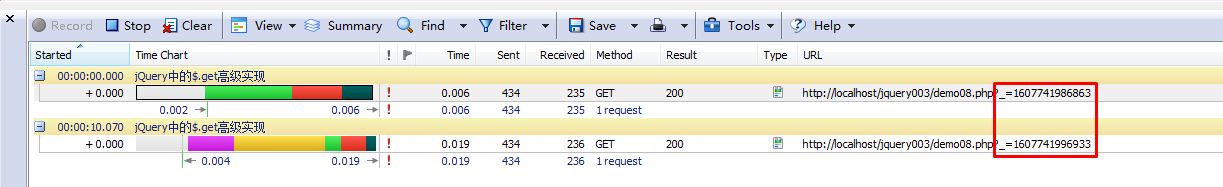


由上图运行结果可知，$.get高级实现在IE浏览器下向同一url发送请求时也会产生缓存问题。

解决之道：使用时间戳的方式解决Ajax的缓存问题



运行结果：

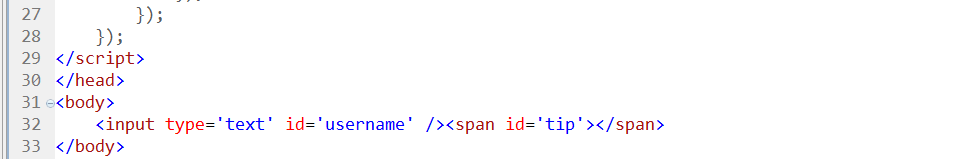


## 9、高级应用$.post

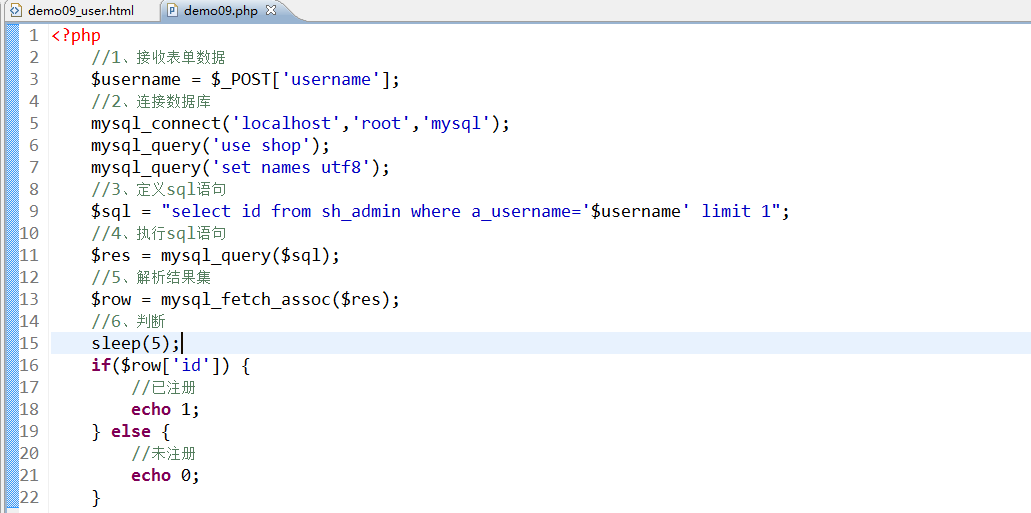
* $.post(url,[data],[callback])

demo09\_user.html 示例代码：使用$.post判断用户名是否已注册





demo09.php 示例代码：



## 10、$.get与$.post中期待的返回值类型

基本语法：

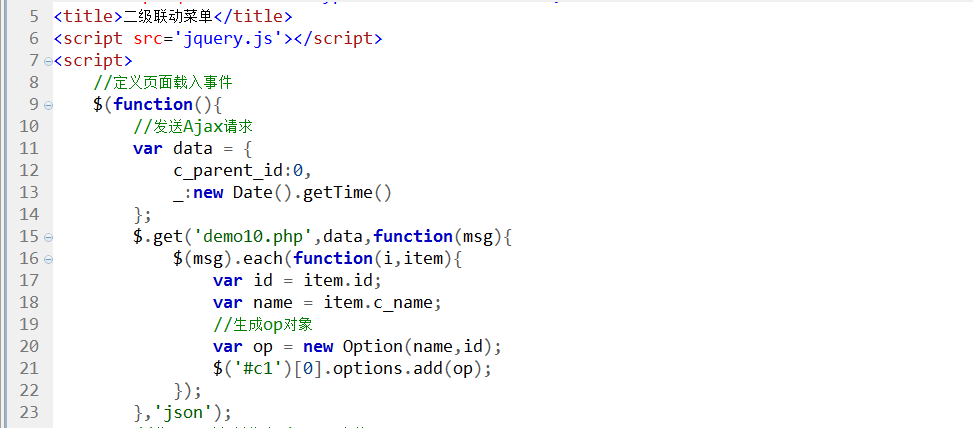
$.get(url,[data],[callback],[type])

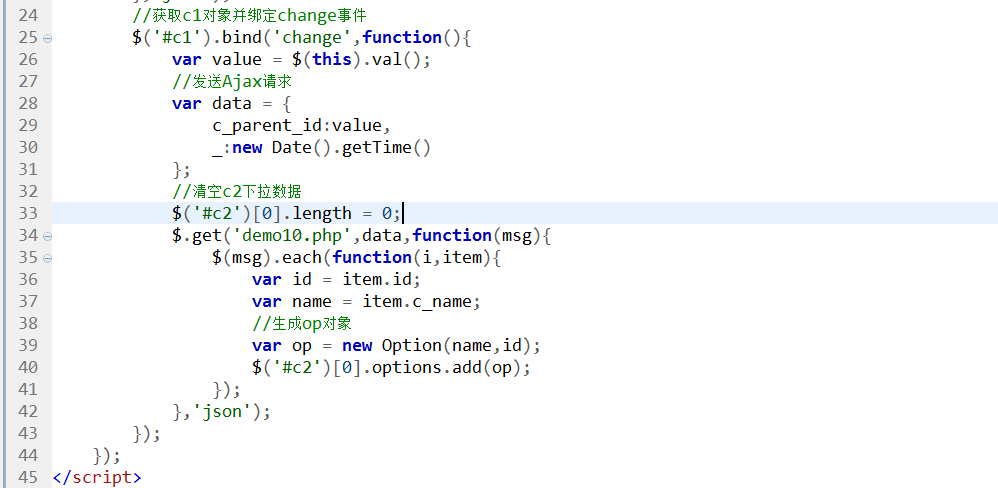
$.post(url,[data],[callback],[type]]);

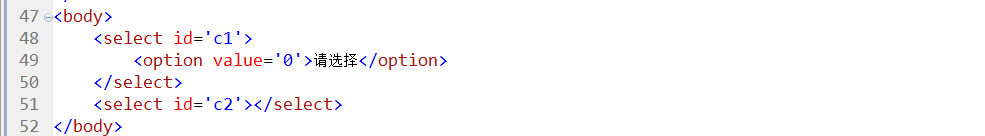
参数说明：

[type] ：期待的返回值类型，可以是text/xml/json

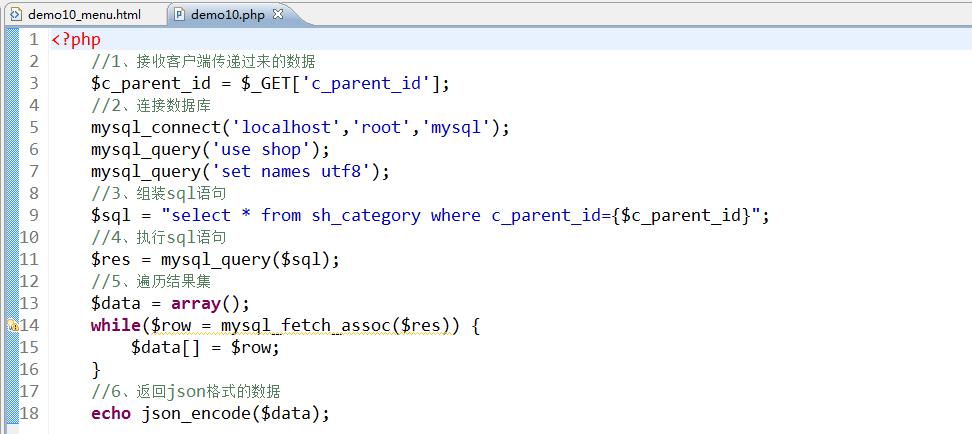
demo10\_menu.html 示例代码：使用ajax+json实现二级联动菜单







demo10.php 示例代码：



# 三、jQuery课程总结

## 1、each方法

each方法的主要是用于遍历jQuery对象。jQuery对象是一个DOM对象的集合，所以每次遍历时得到的结果都是一个DOM对象。

jQuery对象.each(function(i,item) {

//i代表索引下标

//item代表当前正在遍历的DOM对象

});

## 2、jQuery中的Ajax技术

① 回顾Ajax技术

Ajax中的get请求（五步走）

Ajax中的post请求（六步走），post请求要求设置请求头

xhr.setRequestHeader(‘Content-type’,’application/x-www-form-urlencoed’);

② 底层实现+高级实现

③ 底层实现$.ajax

url、type、data、dataType、success

get请求：有缓存问题，我们可以使用cache:false来解决缓存

④ 高级实现$.get与$.post

url、data（要求是字符串或json格式）、callback、type

$.get在IE向同一个url发送请求也会存在缓存问题，可以使用时间戳来进行解决

var data = {

\_:new Date().getTime()

};

⑤ 期待的返回值类型

底层实现dataType属性来设置的

高级实现type第四个参数来实现的

⑥ xml结构的数据，服务器端必须设置响应头信息

json格式的数据，必须使用json\_encode转义，在客户端可以使用each方法对数据进行遍历操作