复习：

AJAX概述： AJAX = HTML + CSS + JS + DOM + HTTP + XML 实现“无刷新无提交的页面内容局部更新”效果。

使用AJAX发起两种请求：

var xhr = new XMLHttpRequest();

xhr.onreadystatechange = fn;

xhr.open('GET','x.php?k=v&k=v',true);

xhr.send(null);

------------------------------------

var xhr = new XMLHttpRequest();

xhr.onreadystatechange = fn;

xhr.open('POST','x.php',true);

xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded');

xhr.send('k=v&k=v');

使用AJAX接收五种响应：

服务器端：

header('Content-Type: text/plain');

echo '一段文本';

客户端：

xhr.responseText

------------------------------------

服务器端：

header('Content-Type: text/html');

echo '<tr><td>唐模</td></tr>';

客户端：

tbody.innerHTML = xhr.responseText

--------------------------------------

服务器端：

header('Content-Type: application/javascript');

echo 'alert("添加成功")';

客户端：

eval( xhr.responseText )

--------------------------------------

服务器端：

header('Content-Type: application/xml');

echo '<empList><emp></emp></empList>';

客户端：

var document = xhr.responseXML; //DOM树

----------------------------------------

服务器端：

header('Content-Type: application/json');

echo '[{"name":"Tom","age":20},{},{}]';

客户端：

JSON.parse( xhr.responseText )

今日目标：

(1)补充：接收JSON类型的响应——重点

(2)jQuery中封装的AJAX函数——重点

1.补充：使用XHR接收响应消息——application/json

JSON：JavaScript Object Notation，JS对象表示法，注意：JSON与JS没任何关系！JSON是一个字符串格式，用于描述数据，作用类似于XML，但比XML更简单。被几乎所有主流语言所支持，用于数据交换，如PHP数据=>JSON字符串=>JS数据。

JSON语法：

表示数组的字符串：[ 10,"Tom",true, "2015-10-11" ]

表示对象的字符串：{ "ename":"Mary", "age":20 }

注意：(1)一段JSON字符串数据要么是“一个数组”，要么是“一个对象”——不能是多个！ (2)JSON字符串中对象属性、string数据都只能使用双引号！

JSON字符串在项目中的使用方法：

服务器端：

header('Content-Type: application/json;charset=UTF-8');

$arr = [ [ ],[ ],[ ] ];

$str = json\_encode($arr); //把PHP数组编码为JSON字符串

echo $str;

客户端：

JSON.parse( xhr.responseText ); //把JSON字符串解析为JS数组/对象

2.jQuery中的AJAX封装函数——load

使用方法：

$('div').load('x.php', ['k=v&k=v'], [fn])

作用：

发起异步GET/POST请求，获取服务器端返回的html片段数据，把这些数据设置为当前选定元素的innerHTML——覆盖已有内容。

案例：提取全站多个网页都具备的相同内容，异步加载

3.jQuery中的AJAX封装函数——$.get

使用方法：

$.get('x.php', 'k=v'|{k:v}, fn);

作用：

向指定的URL发起异步的GET请求，获得响应消息，调用回调函数进行响应数据的处理。

练习：实现一个二级级联下拉列表

4.jQuery中的AJAX封装函数——$.post

使用方法：

$.post('x.php', 'k=v'|{k:v}, fn);

作用：

向指定的URL发起异步的POST请求，获得响应消息，调用回调函数进行响应数据的处理。

练习：创建一个不间断的“员工信息录入系统”——异步POST提交

|  |
| --- |
| 补充小知识点：jQuery提供的表单序列化（serialize）函数——把表单中的用户输入拼接为键值对字符串，可用于HTTP请求消息。  var data = $('#loginForm').serialize();  返回值形如： 'k=v&k=v'  其中的k就是input的name属性值。 |

5.jQuery中的AJAX封装函数——$.getScript

使用方法：

$.getScript('x.php', ['k=v'|{k:v}], [fn]);

作用：

向指定的URL发起异步的GET请求，获得application/javascript类型的响应消息，自动调用eval()把xhr.responseText加以执行。

提示：若响应消息指定了内容类型为javascript，$.get和$.getScript都会自动调用eval()；但响应消息没有指定内容类型为javascript，$.getScript仍然会调用eval()，$.get不会。

练习：异步请求服务器端的实现了国际化的欢迎消息。

6. jQuery中的AJAX封装函数——$.getJSON

使用方法：

$.getJSON('x.php', ['k=v'|{k:v}], fn);

作用：

向指定的URL发起异步的GET请求，获得application/json类型的响应消息，自动调用JSON.parse()把xhr.responseText解析为JS数据。

提示：若响应消息指定了内容类型为json，$.get和$.getScript都会自动调用JSON.parse()；但响应消息没有指定内容类型为json，$.getScript仍然会调用JSON.parse()，$.get不会。

练习：点击按钮，加载更多“好友消息”

177322995514

**服务器端：PHP把数组转换为JSON字符串：**

$str = json\_encode($arr); echo $str;

**客户端：把JSON字符串解析为JS数据：**

var data = JSON.parse(xhr.responseText);

上述五个函数有个共同的缺陷：只能处理成功的响应消息！

面试题：jQuery中AJAX函数，如何处理失败的响应？如何在请求之前执行一个回调函数？

7.jQuery中的AJAX封装函数——$.ajax——万能AJAX函数

$.ajax()功能最丰富，使用偏复杂，上述五个函数都是这个函数简化版本。

使用方法：

$.ajax( {

type: 'GET', //POST、PUT、DELETE、HEAD

url: 'x.php',

data: {uname:'tom',upwd:'123'}, //k=v&k=v,

beforeSend: fn, //请求发送之前的回调

success: fn, //响应成功的回调

error: fn, //响应失败的回调

complete: fn //响应完成的回调(无论成功还是失败)

} );

对应于原生AJAX的下述代码：

var xhr = new ...();

xhr.onreadystatechange = function(){

if(xhr.readyState===4){

if(xhr.status===200){

success();

}else{

error();

}

complete();

}

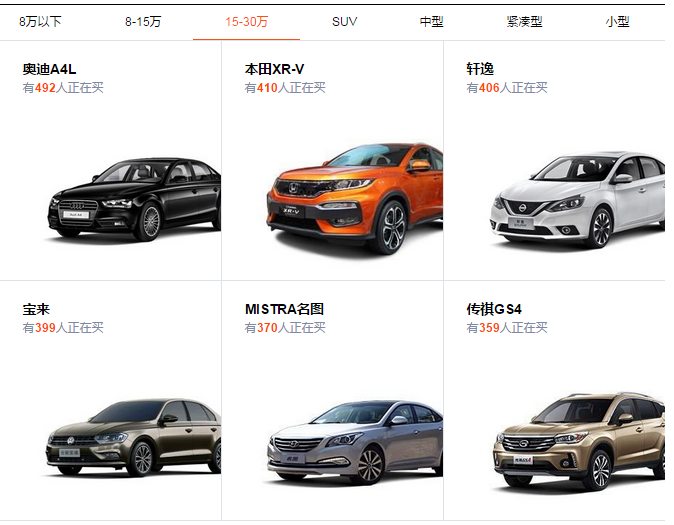
}

xhr.open();

beforeSend();

xhr.send();

课后练习：仿beijing.huimaiche.com中的车辆展示



(1)创建huimaiche.sql，保存车辆数据（cid,cname,price,count,type）

(2)创建car\_select.php，根据客户端提交的车辆类别，返回该类别下所有汽车，以JSON格式

(3)创建car\_select.html，用户鼠标悬停在不同的车辆类别上，异步请求该类别下所有的汽车

$.get(‘1.php’,{uname:’tom’,age:20},function(){});

$.post();