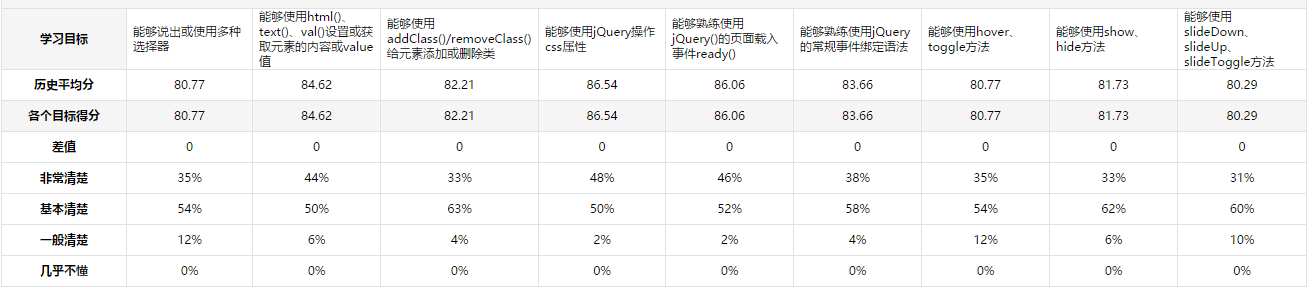
jQuery（1）

# 昨日回顾



# 今日目标

1、能够知道jQuery是如何释放$符号的

2、能够将jQuery对象和DOM对象相互转化，以及为什么要相互转化

3、能够掌握jQuery普通事件的绑定方法

4、能够掌握jQuery高级事件的使用方法

5、能够掌握jQuery中封装的发送Ajax的get方法

6、能够掌握jQuery中封装的发送Ajax的post方法

7、能够掌握jQuery中封装的发送Ajax的ajax方法

# 三、杂项

1、释放$

1、什么是释放$

$表示jQuery对象，或者说它表示jQuery函数。使用jQuery必须要使用$符号。

释放$，也就是在使用jQuery的时候，不用$符号。

2、为什么要释放$

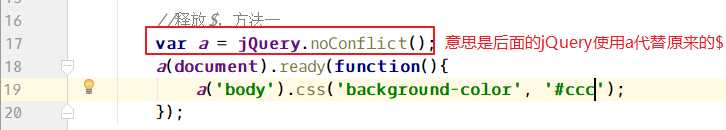
为了避免冲突，所以要释放$.

有些时候，可能项目中用到的js框架不止一个，其他框架有可能也用$.

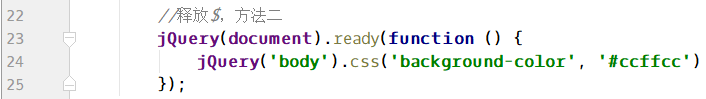
有些时候，我们自己定义一个函数，函数名就是$. 这些情况都会和jQuery中的$产生冲突。

3、如何释放$，如何不使用$，还能用jQuery

**方法一**：



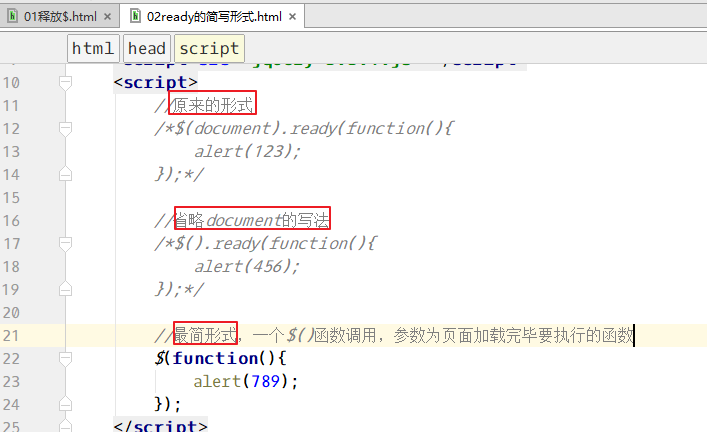
**方法二**：



**方法三**：



2、ready事件的简写形式



3、ready事件和onload事件的区别

**区别1**：执行时机不同

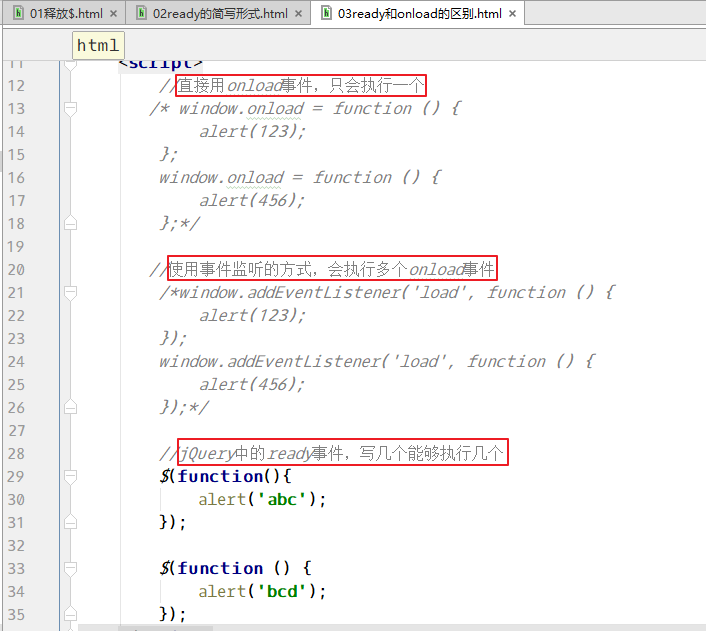
ready事件在页面的dom节点加载完毕就会执行，无需下载图片等资源。

onload事件在页面完全加载完毕才会执行，需要页面中所有的资源都下载完毕才会执行。

**区别2**：执行次数不同

默认的onload事件只能执行一个

ready事件，写几个，执行几个。



4、DOM对象和jQuery对象的区别

1、什么是DOM对象？什么是jQuery对象？

DOM对象就是通过纯JS的方式，比如document.getEle…..得到的对象就是DOM对象。

jQuery对象就是通过jQuery方法 $('li'); 得到的对象。

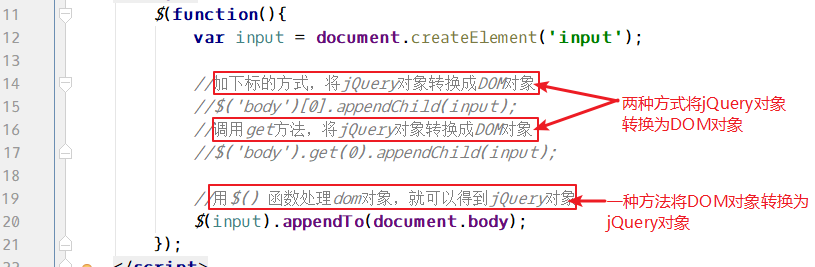
2、DOM对象和jQuery对象的区别

DOM对象能够调用DOM方法，而不能调用jQuery方法；

jQuery对象能够调用jQuery方法，而不能调用DOM方法；



3、DOM对象和jQuery对象互换



5、each方法

人性化的遍历

**语法一**：

需要遍历的对象.each(function(i, val){

//每遍历一次，执行一次该函数

//形参i，就是每次遍历时对象的下标

//形参val，就是每次遍历的dom对象

});

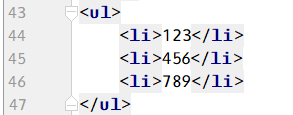
**语法二**：

$.each(需要遍历的对象, function(i, val){

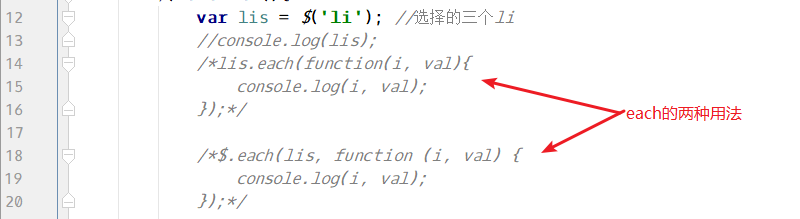
//每遍历一次执行一次该函数。形参i和val和上面的语法中的一样

});

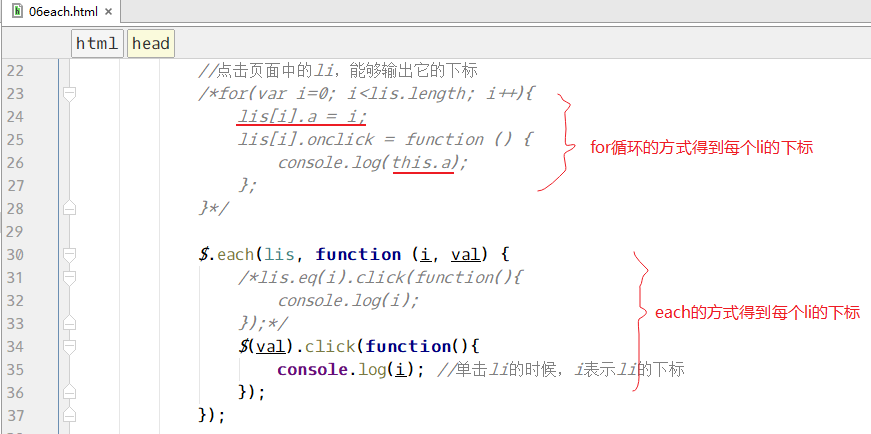
each用于遍历对象或数组，可以理解为for或for…in循环，但是比for和for…in要强大。



演示each的两个用法：



each和事件配合，能够得到每个li的下标。



# 四、jQuery中的事件

1、普通事件

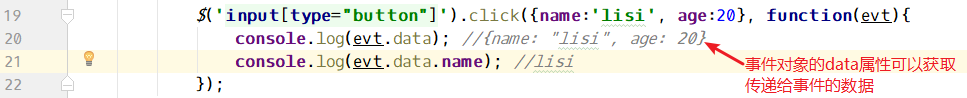
普通事件，比如有click、mouseenter、change、blur…等

语法：click([data], fn);

1、事件方法没有参数，可用于传递事件



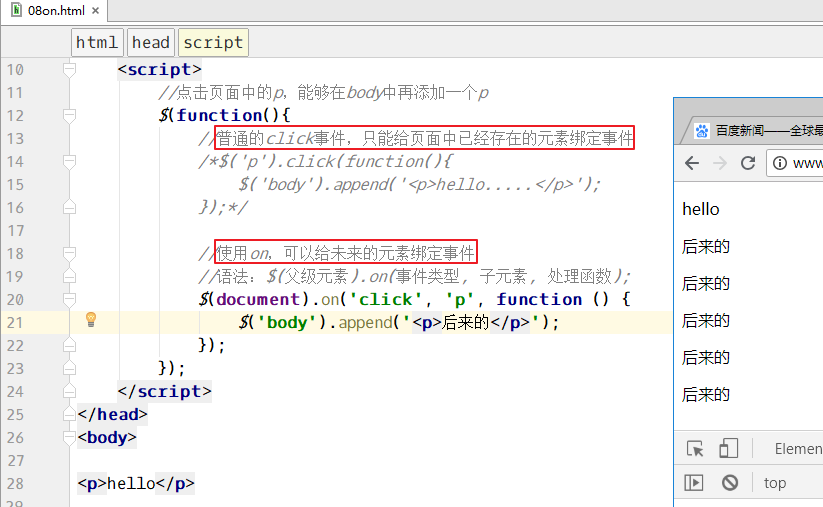
2、给事件传入的数据，可通过事件对象的data属性获取



2、on事件

1、给未来的元素绑定事件（事件委托）

**语法：$(父级元素).on(事件类型, 子元素, 处理函数);**

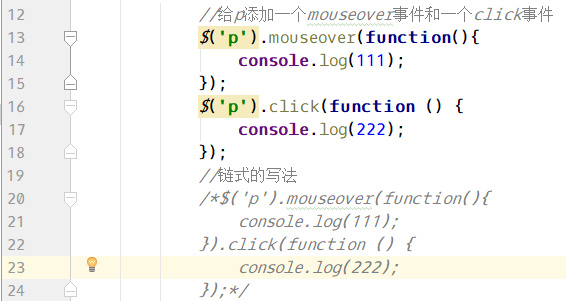


**给未来的元素绑定事件（事件委托）常用于给Ajax返回的数据添加事件**。

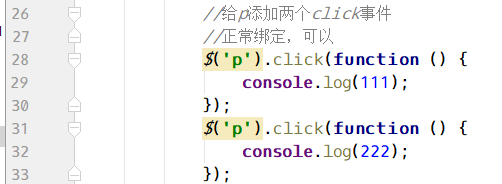
PS：1.7版本及之后，可使用on来完成事件委托；1.7之前的版本可以使用live或bind来完成事件委托

2、给一个元素绑定多个同类型的事件

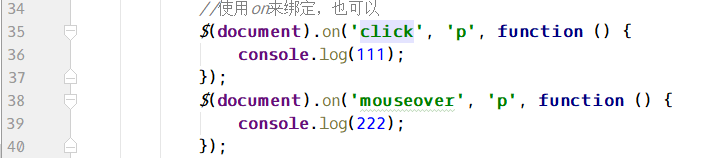
不同on也可以完成这个功能：

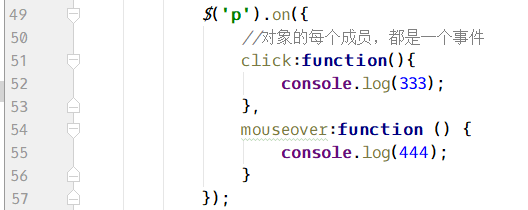


不用on，也可以为一个元素绑定多个同类型的事件：



用on给一个元素绑定两个不同的事件：





用on为一个元素绑定多个同类型的事件：



3、off事件

用于取消元素的某个事件。

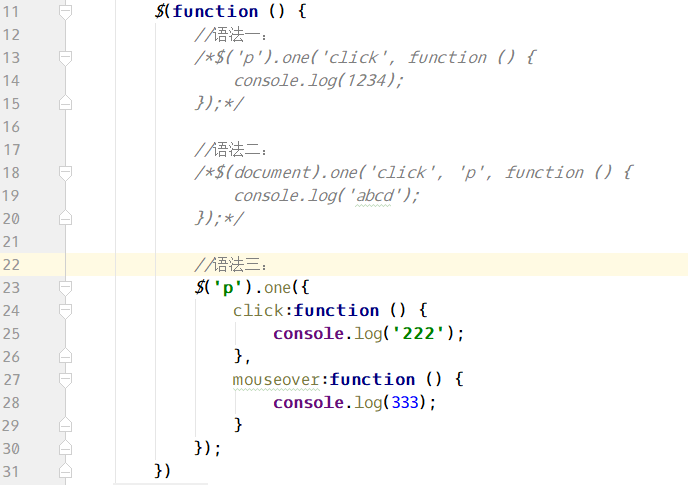
实际开发中，这个东西很有用，有些时候是需要先取消掉一个元素的所有事件，然后在绑定你要绑定的事件，可以解决一些bug。



**PS：off和on一组的，都是3.0版本后存在，3.0版本之前，可以使用unbind事件来完成事件的取消工作**。

4、one事件

给元素绑定的事件，只执行一次。



# 五、jQuery封装的Ajax

1、$.get方法

语法：

$.get(url, data, function(e){

//e就是服务器返回的数据

}, dataType);

**四个参数**：

url：

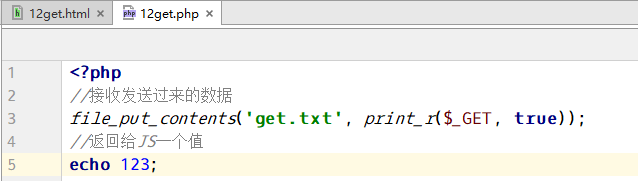
data：发送给服务器的数据，字符串的形式或对象的形式

fn：用于处理服务器返回的数据，形参e就是服务器返回的数据

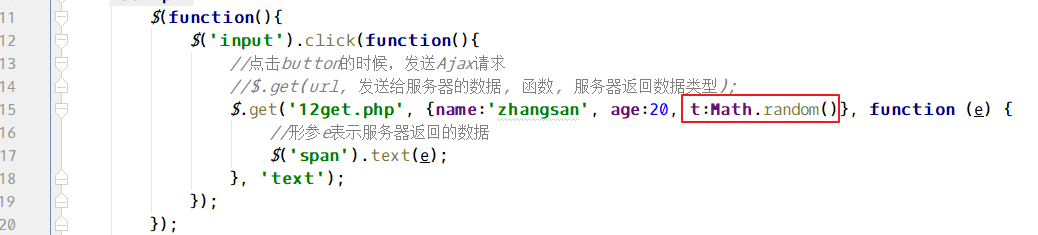
dataType：表示服务器返回数据的类型（text/json/html/script/\_default）



php 页面：



如果希望IE浏览器不缓存，也可以使用加随机数的方式解决：



2、$.post方法

语法：

$.post(url, data, function(e){

//e就是服务器返回的数据

}, dataType);

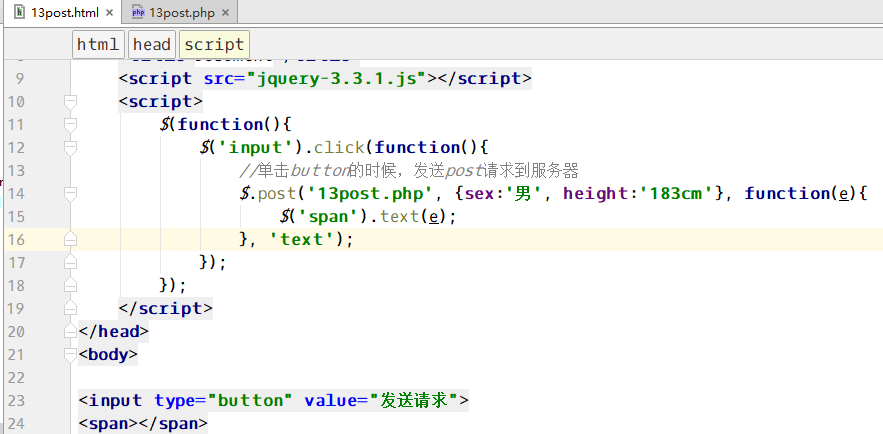
**四个参数**：

url：

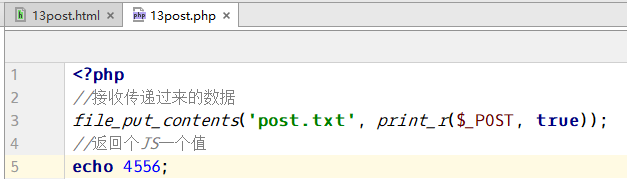
data：发送给服务器的数据，字符串的形式或对象的形式

fn：用于处理服务器返回的数据，形参e就是服务器返回的数据

dataType：表示服务器返回数据的类型（text/json/html/script/\_default）



PHP代码：

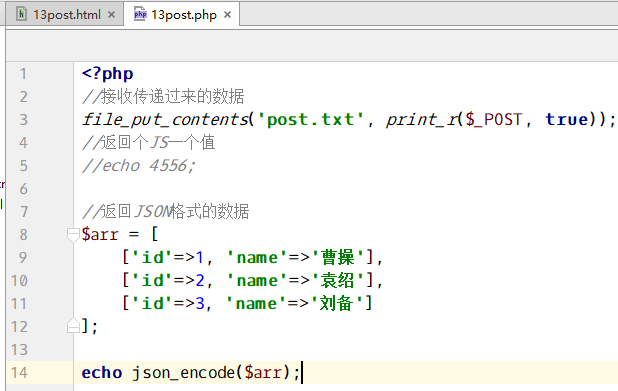


**服务器返回JSON格式的数据**：

JS代码，需要修改get或post方法的第4个参数为json，修改之后，接收到的数据，jQuery会自动将其转换为JS能够识别的数组或对象。



PHP代码：



3、$.ajax()方法

使用语法：

$.ajax({

type:'get', //或者post

url:'',

data:{}, //发送给服务器的数据

dataType:'json', //服务器返回数据的类型

success:function(e){

//形参e仍然是服务器返回的数据

},

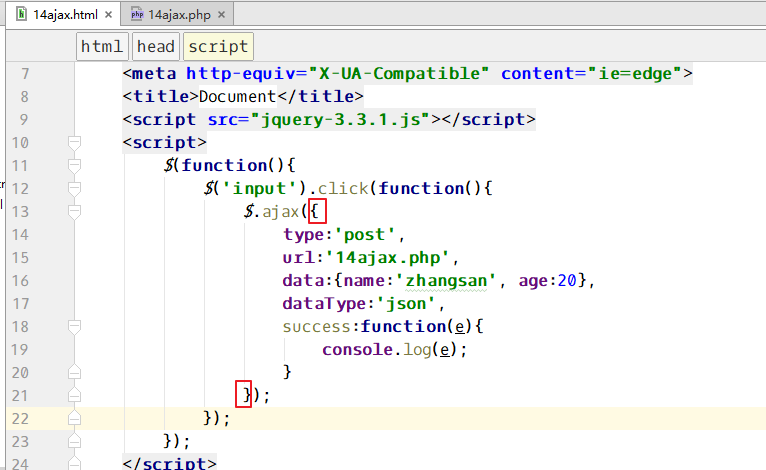
error:function(){

//ajax执行出错，在这里处理

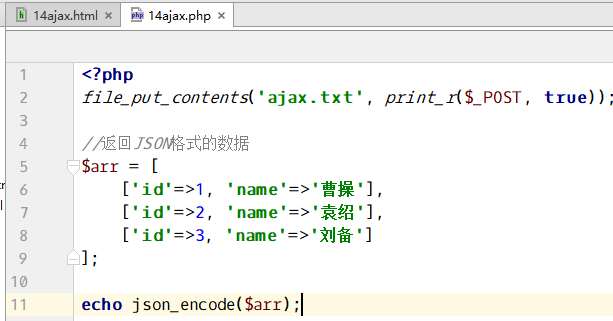
},

async:false, //表示同步请求，默认为true表示异步请求

});



PHP代码：



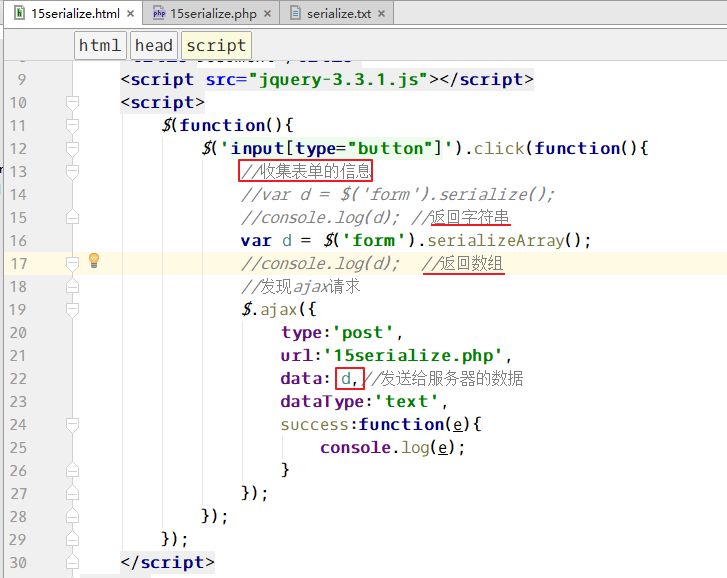
4、收集表单信息方法

使用方法：

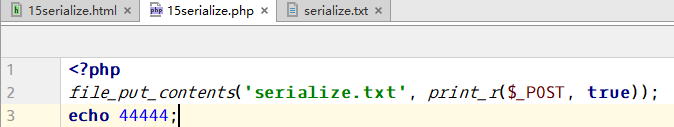
$(表单).serialize(); //返回一个经过编码的字符串

$(表单).serializeArray(); //返回一个JS数组

二者收集到的数据都可以直接用于Ajax。



PHP页面：



5、Ajax提示方法

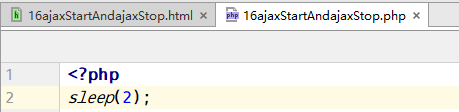
ajaxStart()：在ajax开始之后，会执行

ajaxStop()：在ajax结束时执行。

ps：低版本的写法是手册上的写法，高版本的jQuery写法如下：



PHP代码：



# 六、案例 -- 无刷新的分页

1、分页原理

//查询所有数据

select \* from sheng;

//假设每页显示3条数据，用变量$pageSize表示，即$pageSize=3;

//第一页：select \* from sheng limit 0,3;

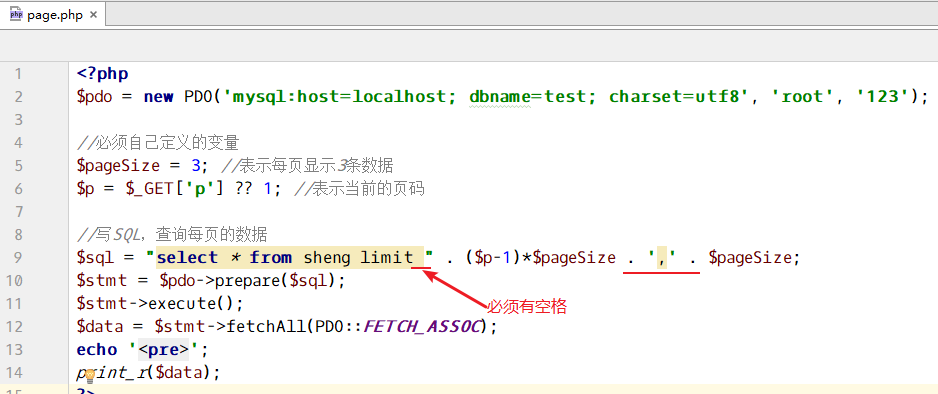
//第二页：select \* from sheng limit 3,3;

//第三页：select \* from sheng limit 6,3;

//当前的页码从地址栏获取，如果地址栏存在p参数，就获取它，如果不存在p参数，默认是第1页

//**第$p页：select \* from sheng limit ($p-1)\*$pageSize, $pageSize;**

2、实现分页数据查询

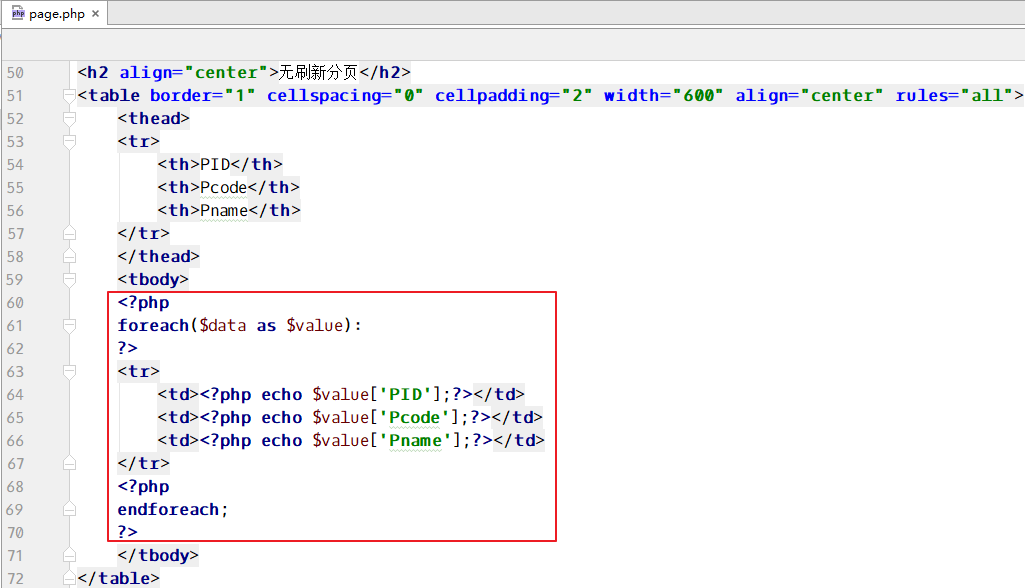


浏览器直接访问这个page.php页面，如果能够看到第1页的数据，可以手动在地址栏添加p参数，看一下是否能够获取到下一页的数据：

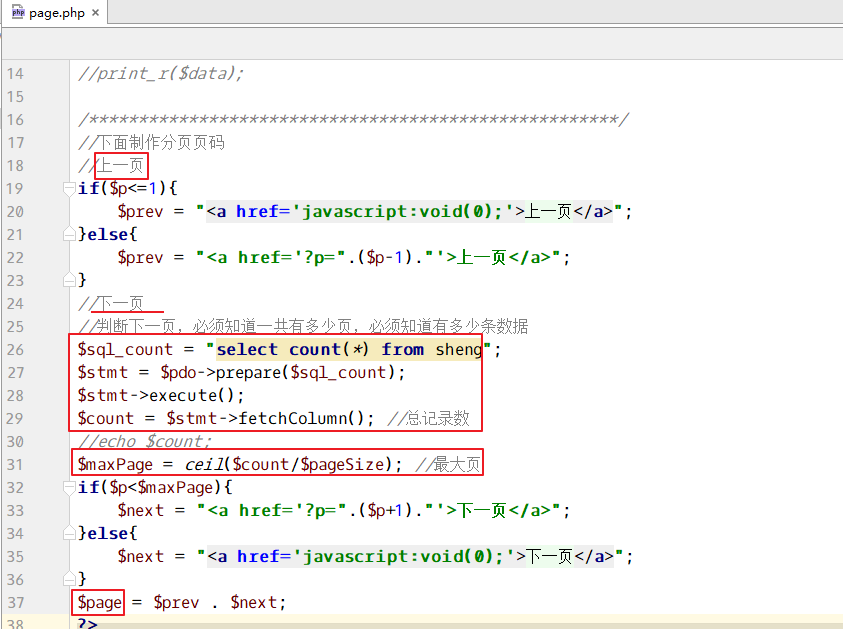


3、输出数据并实现页码功能

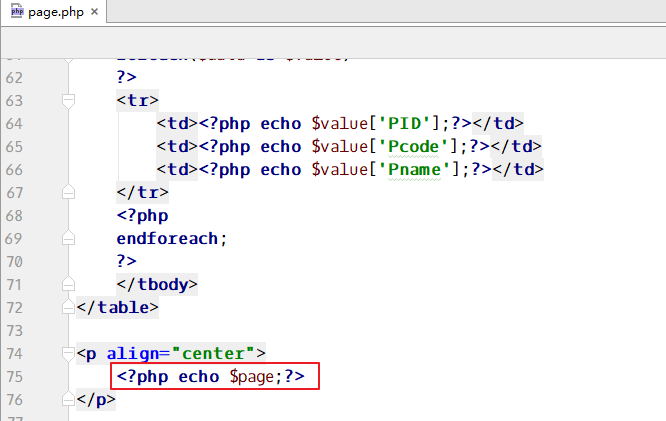
输出数据：



制作页码：



把得到的$page放到页面中即可：



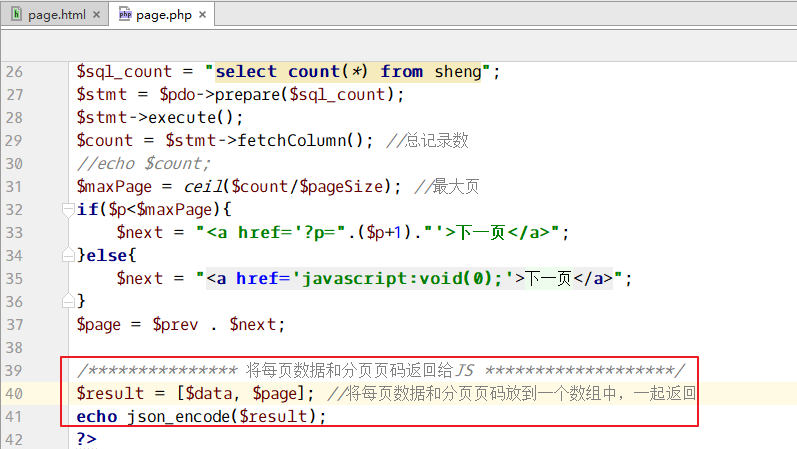
4、无刷新分页，获取第一页数据

首先将原page.php中的HTML部分剪切出来，放到page.html的文件中。

page.html中，写JS，当页面加载完毕，发送ajax请求到page.php。请求第一页的数据和分页页码：



page.php中将数据data和分页page合并到一起，然后转换成json返回：



page.html中，处理服务器返回的数据：



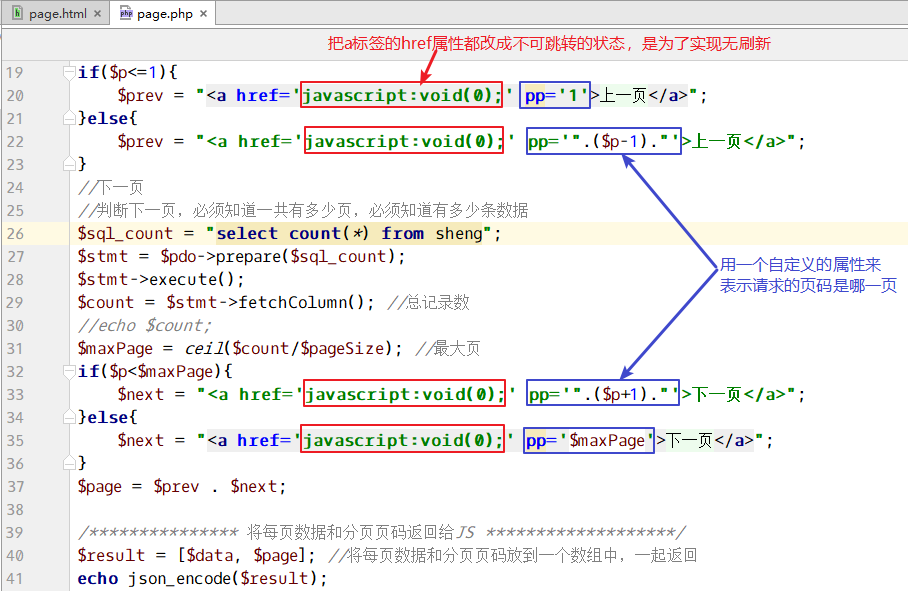
这样，访问page.html，就可以看到第一页的数据了。

5、实现无刷新的分页

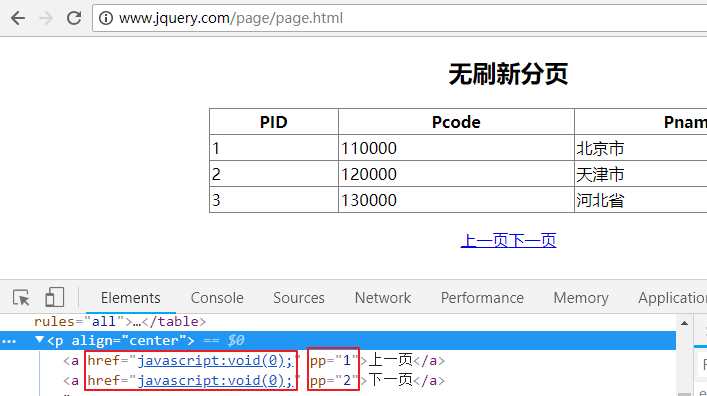
要实现无刷新的分页，必须让a标签不能跳转才可以。所以将a标签的href属性值都写成javascript:void(0);

又因为，点击a标签的时候，必须知道你要请求哪一页的数据，所以在a标签上还必须体现出请求的页码。

所以修改php代码：



检测一下，刷新page.html，通过浏览器工具，就可以看到a标签的样子如下：



下一步，点击页面中的a标签，发送ajax请求就可以了：



请求完数据，处理方式和第一页相同，可以复制处理数据的代码，也可以封装成一个函数：