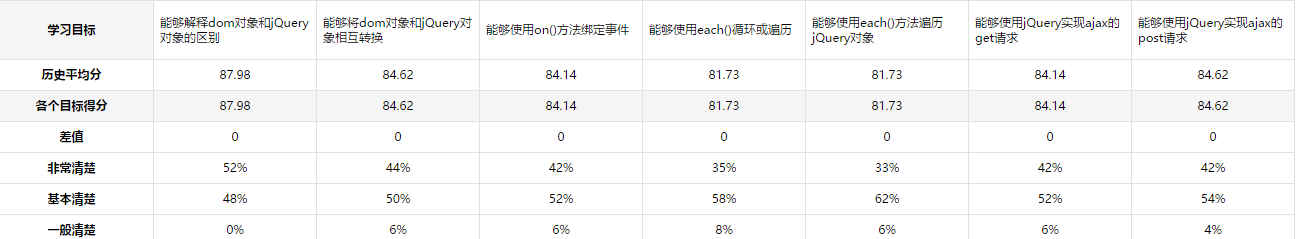
# 一、昨日回顾



1、dom对象和jQuery对象相互转化

DOM对象转换为jQuery对象 🡪 $(dom对象) ==== jQuery对象;

jQuery对象转换为DOM对象 🡪 jQuery对象[0] / jQuery对象.get(0) ==== DOM对象

2、on方法

$(document).on(事件, 给谁添加数据, 事件处理函数);

3、each遍历

$.each(需要遍历的对象, function(i, value){

I 表示下标

value表示值

});

对比PHP的foreach

foreach($arr as $key=>$value){

key表示下标

value表示值

}

4、get、post、ajax方法

$.get(url, {发送给服务器的数据}, function(e){

处理服务器返回的数据，数据就是e

}, 'json');

$.post(url, {发送给服务器的数据}, function(e){

处理服务器返回的数据，数据就是e

}, 'json');

$.ajax({

url:请求地址

type:'get',

data:{}, //发送给服务器的数据，如果没有则省略

dataType:'json', //服务器返回数据类型

success:function(e){

//处理服务器返回的数据

}

});

# 二、今日目标

# 三、效果

* 基本
  + show([s,[e],[fn]])
  + hide([s,[e],[fn]])
  + toggle([s],[e],[fn])
* 滑动
  + slideDown([s],[e],[fn])
  + slideUp([s,[e],[fn]])
  + slideToggle([s],[e],[fn])



* 淡入淡出
  + fadeIn([s],[e],[fn]) -- 淡入
  + fadeOut([s],[e],[fn]) -- 淡出
  + fadeTo([[s],o,[e],[fn]]) -- 将透明度慢慢变化到指定的值，参数o表示透明度（0~1之间的值）
  + fadeToggle([s,[e],[fn]])
* 自定义
  + animate(p,[s],[e],[fn])
  + stop([c],[j])
  + delay(d,q]) -- 延迟执行下一个动画
  + finish(queue])



效果方法中参数 s 表示时间或者速度， e 表示效果， fn表示效果结束后要执行的函数。

# 四、跨域请求

1、跨域的三种方式

1、使用代理

自己的网站发送Ajax请求 🡪 自己网站的php文件 🡪 请求其他网站

2、设置对方网站的header('Access-Control-Allow-Origin:\*');

3、使用JSONP技术

2、JSONP技术

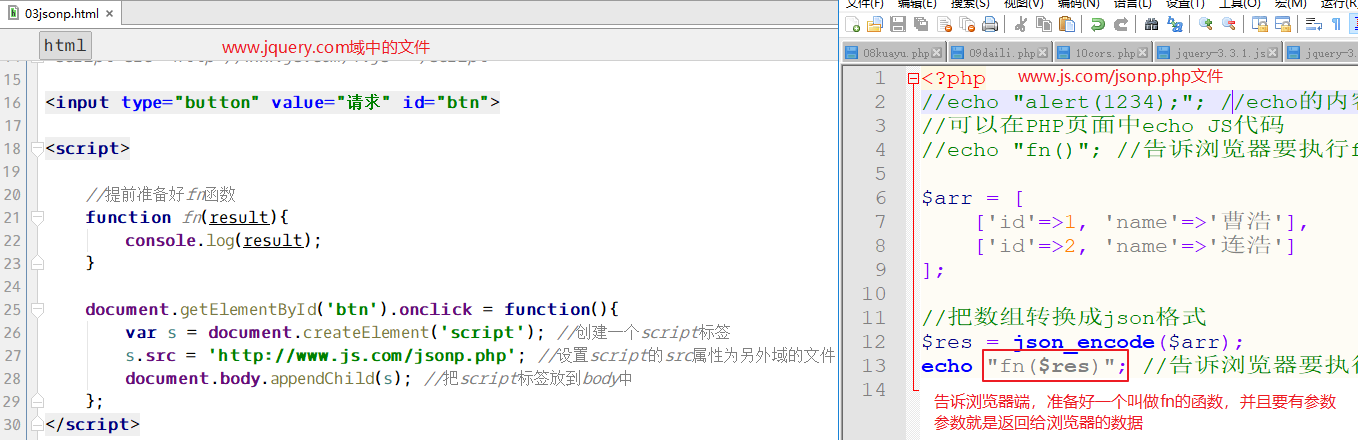
Jsonp(JSON with Padding) 是 json 的一种"使用模式"，通俗的讲，jsonp可以通过html标签中的src属性访问另外域的内容，可以让网页从别的域名（网站）那获取资料，即跨域读取数据。

什么样的HTML标签有src属性：

img

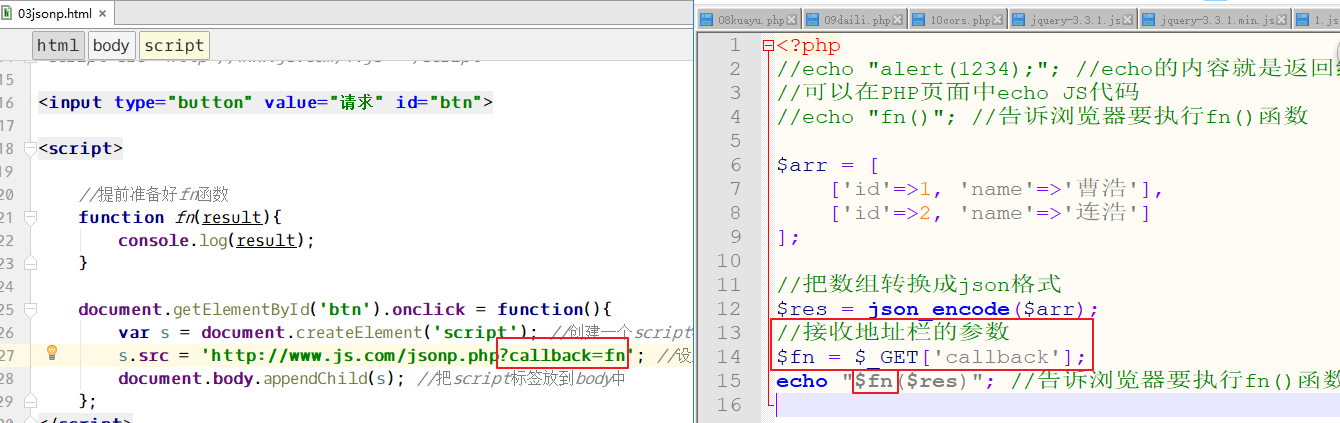
script

iframe



上面的例子，要求浏览器端设置的函数名必须和另外域的服务器中要求的函数名一直，案例中都必须是fn。

如果函数名不知道是什么？可以传递参数，达到函数名一致的要求。



3、jQuery中的Ajax跨域请求

$.get('http://www.js.com/jsonp.php?fn=?', {}, function(e){

//e就是另外域的PHP返回的数据

}, 'jsonp');

用另外的方法也可以

$.ajax({

type:'get',

data:{},

dataType:'jsonp',

url:'http://www.js.com/jsonp.php?fn=?',

success:function(e){

//处理返回的数据

}

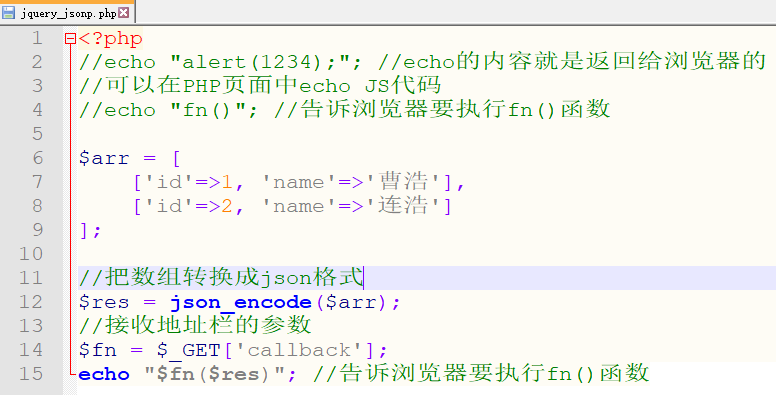
});

专门用于跨域的JSONP请求

*$*.getJSON(  
 **'http://www.js.com/jsonp.php?callback=?'**,  
 **function**(e){  
 **console**.log(e);  
 }  
);



另外域www.js.com中PHP文件：



# 五、文档操作

1、添加节点

1、append：父节点.append(子节点); 给父节点追加一个子节点

2、appendTo：子节点.appendTo(父节点); 把一个子节点追加到一个父节点中

3、prepend：父节点.prepend(子节点); 在一个父节点里开始的位置添加一个子节点

4、after：兄弟节点.before(兄弟节点); 在元素后，添加一个兄弟

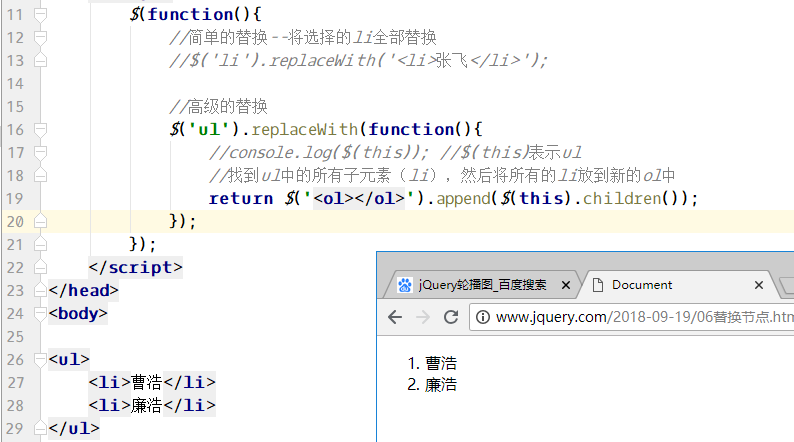
5、before：兄弟节点.before(兄弟节点); 在元素前，添加一个兄弟



2、修改节点

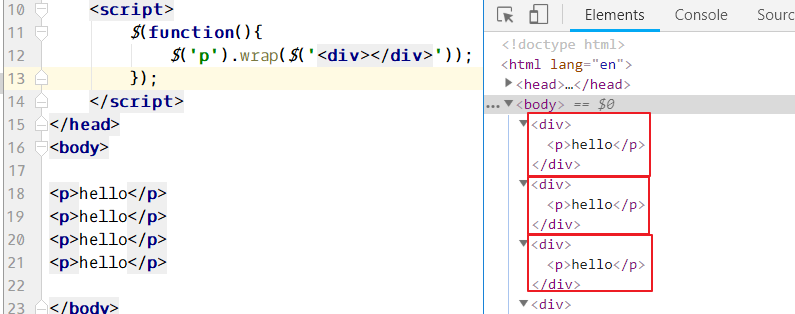
replaceWith：旧节点.replaceWith(新节点); 用新的节点替换旧的节点（**全部替换**）

稍微复杂一点的替换：旧节点.replaceWith(function(){ 返回替换后的结果; }) 用函数处理复杂的替换

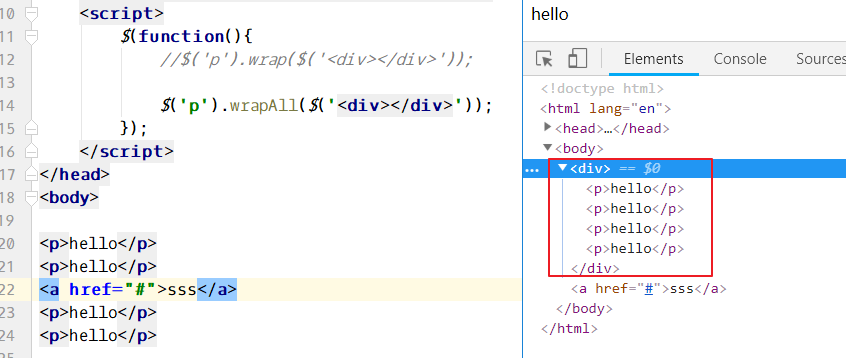


3、包裹节点

wrap：里面的节点.wrap(包裹的节点); 将每个查找到的节点**分别**用包裹的节点包裹



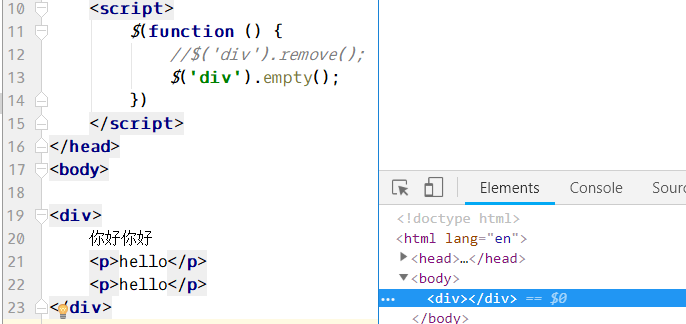
wrapAll：里面的节点.wrapAll(包裹的节点); 将查找到的节点用一个包裹的节点包裹，如果查找的节点不在一起，则强制放到一起，**统一包裹**。



4、删除节点

remove()：待删除的节点.remove(); 将选择的节点删除

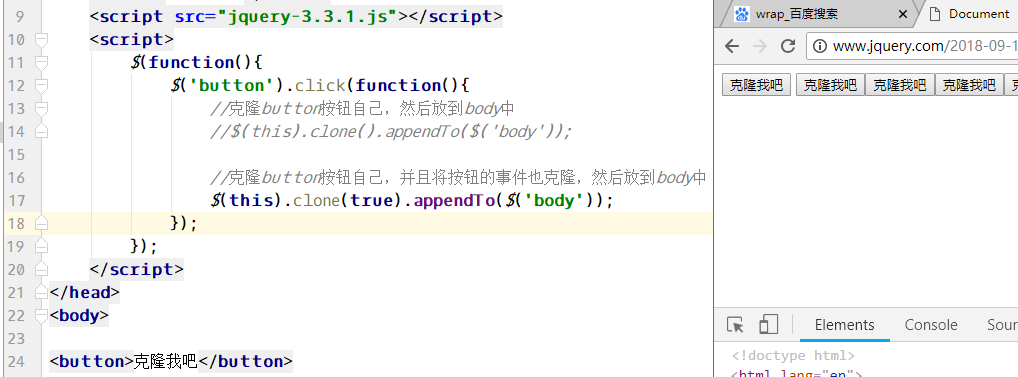
empty()：待清空的节点.empty(); 将选择的节点里面的内容清空



5、克隆节点

clone()：选择的节点.clone(); 将选择的节点（包括里面的内容）克隆

clone(true)：选择的节点.clone(true); 将选择的节点克隆，并克隆该节点的事件。



6、属性操作

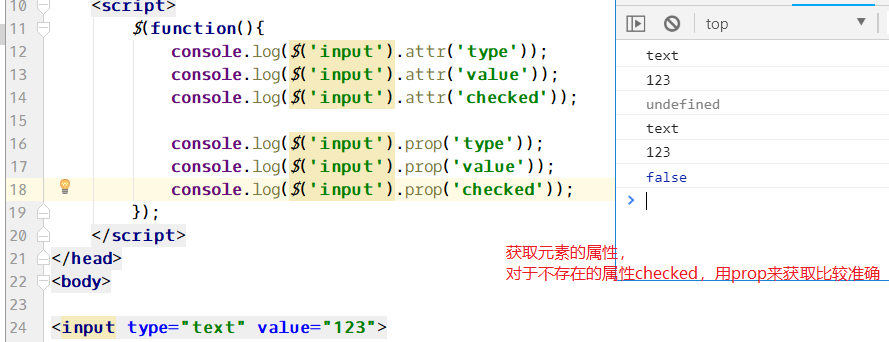
attr()：标准的设置和获取属性方法，类似 “对象.getAttribute()”或“对象.setAttribute()”

对应的移除方法：removeAttr();

prop()：获取/设置不存在的属性比较占优，类似 “对象**.**属性”或“对象**.**属性=值”

对应的移除方法：removeProp();

**获取属性**：

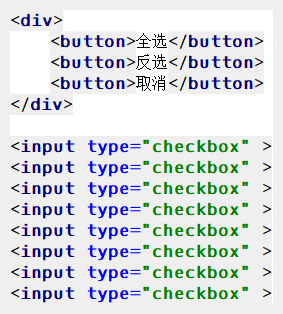


**设置属性**：



7、全选反选取消

HTML代码：



JS代码：



# 六、插件编写

1、$.fn.extend();

这种语法，可以一次性写多个插件，需要给extend传递一个对象，对象的每个成员都是一个插件。

插件就是一个函数，这个函数是一个对象的成员方法。这个对象是extend的参数。



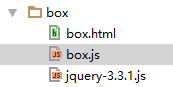
2、$.fn.xxx = function(){}

这种语法，只是定义一个插件。

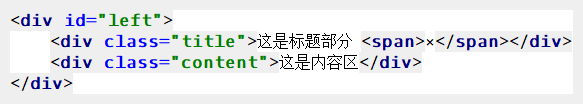


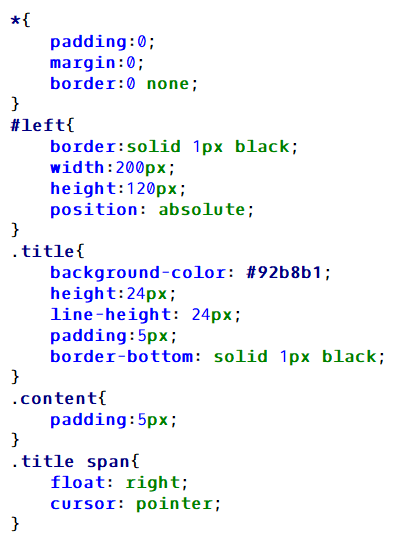
# 七、自定义插件

1、创建目录结构

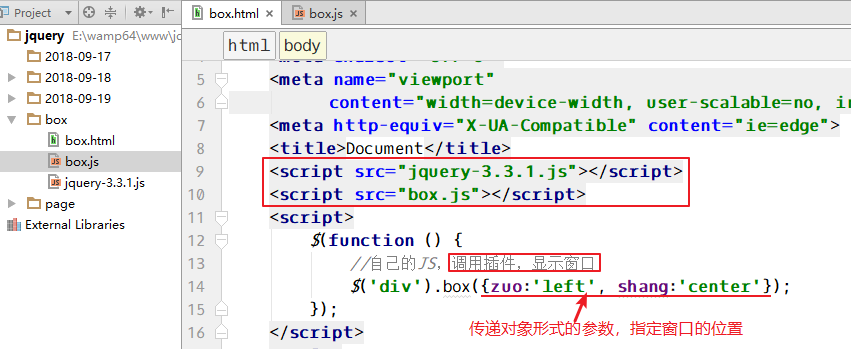


2、HTML+css





3、调用插件



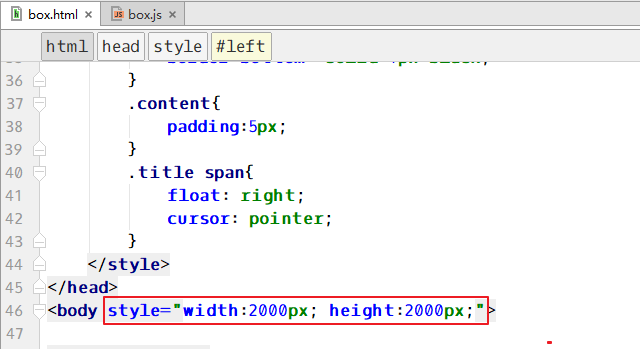
此时没有效果。因为还没有写box插件。

4、完成窗口的显示



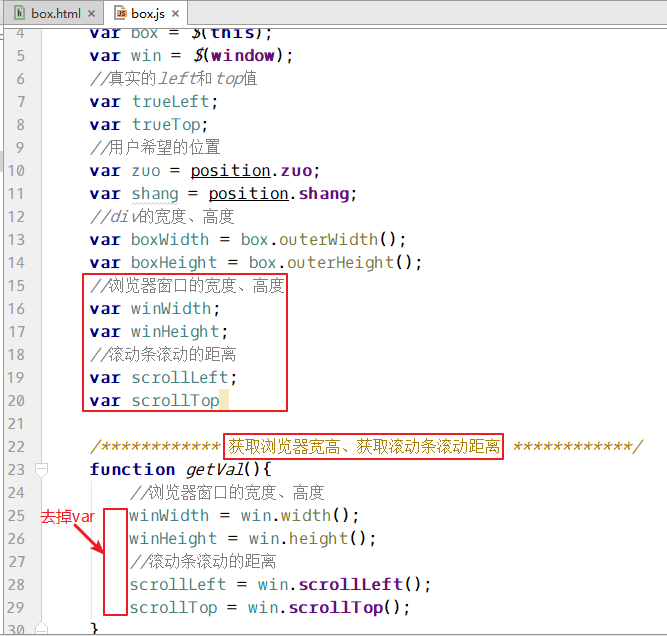
5、当浏览器改变时或滚动条滚动时，保持div的位置不变

首先要给body设置的宽一些、高一些，这样才会有滚动条：



**封装三个函数**：

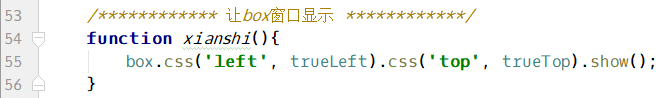
第一个：计算浏览器的宽高及滚动条的滚动距离：



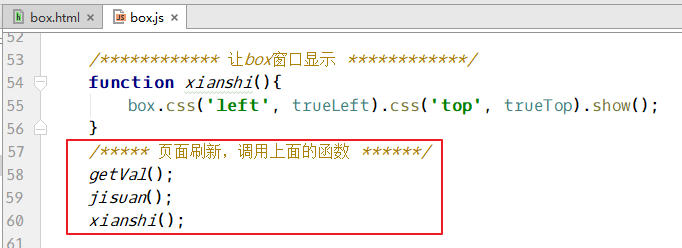
第二个：计算真实的left和真实的top值：



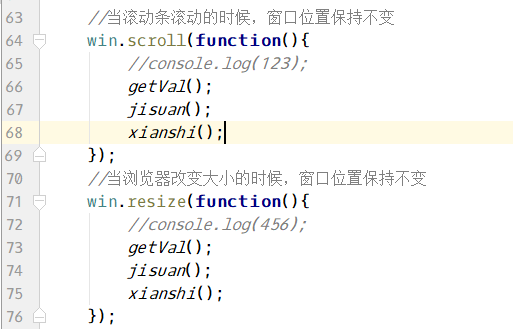
第三个：显示div：



先调用一次这三个函数：



绑定浏览器改变大小事件、滚动条滚动事件：



# 八、第三方插件

1、基本特效

http://www.jq22.com

2、旋转插件

http://www.jqueryrotate.com/

3、图表插件

http://echarts.baidu.com/

4、弹出层插件

https://layer.layui.com/

5、表单验证插件

6、异步上传插件

7、日历插件

8、天气插件