# Day05笔记-JQuery

# 1、案例1：重写弹出广告

## 1.1、需求说明

* 网站顶部在页面打开两秒后显示广告图；
* 广告图展示两秒后关闭广告图；

## 1.2、知识讲解-jQuery(一)

### 1.2.1、jQuery动画效果

jQuery的隐藏和显示相对JS更为动感圆滑。

代码准备：

<style>

#d1{

background-color:lightskyblue;

width:300px;

height:300px;

}

</style>

<script>

//展示

function run1(){

}

//隐藏

function run2(){

}

//切换显示/隐藏

function run3(){

}

</script>

<div id="d1"></div>

<input type="button" value="展示" onclick="run1()" />

<input type="button" value="隐藏" onclick="run2()" />

<input type="button" value="切换显示/隐藏" onclick="run3()" />

以下所有动画效果方法中都有两个参数：

* speed 动画效果的持续时间，单位：毫秒

一般快速效果设置500，正常效果设置1000即可

* fn 动画效果成功展示完成后，回来调用的函数。（回调函数）

基本效果：（放大缩小滑动）

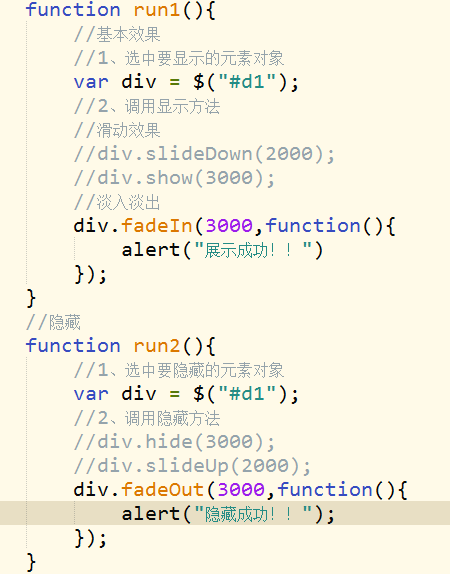
|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| show(speed,fn); | 显示元素 |
| hide(speed,fn); | 隐藏元素 |
| toggle(speed,fn); | 1、使得显示元素隐藏  2、使得隐藏元素显示 |

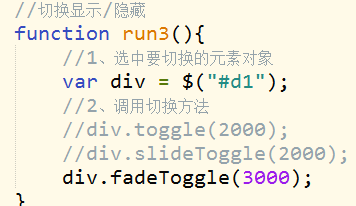
滑动效果：（上下滑动）

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **slideDown(speed,fn);** | 显示元素 |
| **slideUp(speed,fn);** | 隐藏元素 |
| slideToggle(speed,fn); | 1、使得显示元素隐藏  2、使得隐藏元素显示 |

淡入淡出效果：（改变透明度的效果）

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| fadeIn(speed,fn); | 显示元素 |
| fadeOut(speed,fn); | 隐藏元素 |
| fadeToggle(speed,fn); | 1、使得显示元素隐藏  2、使得隐藏元素显示 |
|  |  |





## 1.3、案例代码实现



## 1.4、知识讲解-jQuery（二）

### 1.4.1、层级选择器

和CSS层级选择器类似，可以通过层级关系获取对应标签对象。

* **A B** 获得A元素内部的所有的B元素。 （子子孙孙）
* **A>B** 获得A元素下面的所有B子元素。 儿子
* *A+B 获得A元素同级下一个B元素 下面的第一个兄弟*
* *A~B 获得A元素之后的所有B元素 所有的弟弟*

代码准备：

<script>

$(function(){

//1、获取id为main的span标签 内 所有的div标签

//2、获取id为main的span标签 内 子元素div标签

//3、获取id为main的span标签 后 第一个div兄弟标签

//4、获取id为main的span标签 后 所有的div兄弟标签

});

</script>

<span id="main">

<div>111111</div>

<div>222222</div>

<div>333333</div>

<span>

<div>44444</div>

</span>

</span>

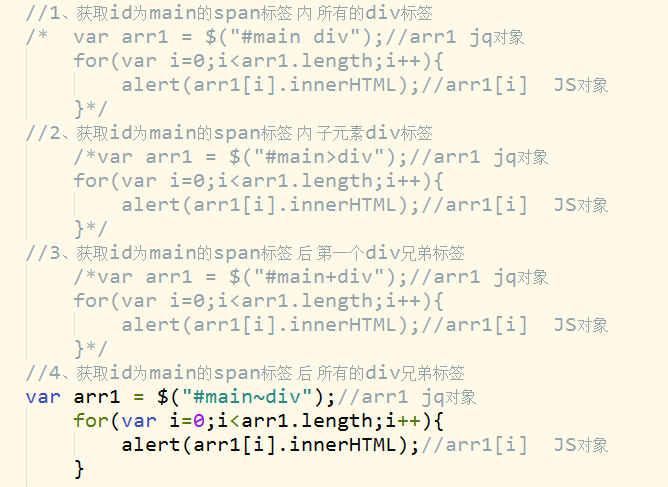
<div>55555</div>

<span>

<div>66666</div>

</span>

<div>77777</div>



### 1.4.2、属性选择器

我们可以通过属性选择器，以属性名或者值来匹配对应标签

* [属性名] 获得有 **指定属性名** 的标签对象。
* **[属性名=值]** 获得 指定属性名**等于**指定值 的标签对象 value = man
* **[属性名\*=值]** 获得 指定属性名 **含有** 指定值 的标签对象 value \*= a

多个属性选择器可以组合使用

* [选择器1][选择器2][选择器3]

代码准备：

<script>

$(function(){

//1、获取所有存在type属性的标签

//2、获取所有type属性为radio的标签

//3、获取所有type属性含有o的标签

//4、获取所有input标签中的单选框，且name为sex的标签

});

</script>

用户名：<input type="text" name="uname" value="小刘"/><br />

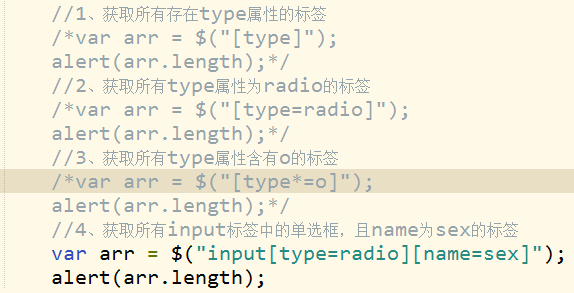
密码：<input type="password" name="pwd" value="123"/><br />

性别：<input type="radio" name="sex" value="man" />男

<input type="radio" name="sex" value="woman" />女<br />

是否VIP：<input type="radio" name="vip" value="no" />不是

<input type="radio" name="vip" value="yes" />是<br />



### 1.4.3、基本过滤选择器

通常我们在获取到一系列标签对象之后，会有一些筛选需求，

这里就可以使用基本过滤选择器。

* **:first**  第一个
* **:last** 最后一个
* **:even** 偶数，**索引**从 **0** 开始计数
* **:odd**  奇数
* :not(..) 除了指定内容 1234 : not(12) == > 34
* :eq(index) 获取指定索引的元素
* :gt(index) 大于index索引的元素
* :lt(index) 小于index索引的元素

代码准备：

<script>

$(function(){

//1、在所有DIV标签中，获取第一个div

//2、在所有DIV标签中，获取最后一个div

//3、在所有DIV标签中，获取所有偶数位的div

//4、在所有DIV标签中，获取所有奇数位的div

//5、在所有DIV标签中，获取除了第一位以外所有的div

//6、在所有DIV标签中，获取索引等于1的div

//7、在所有DIV标签中，获取索引大于1的div

//8、在所有DIV标签中，获取索引小于1的div

});

</script>

<div>11111111，索引是0</div>

<div>22222222，索引是1</div>

<div>33333333，索引是2</div>

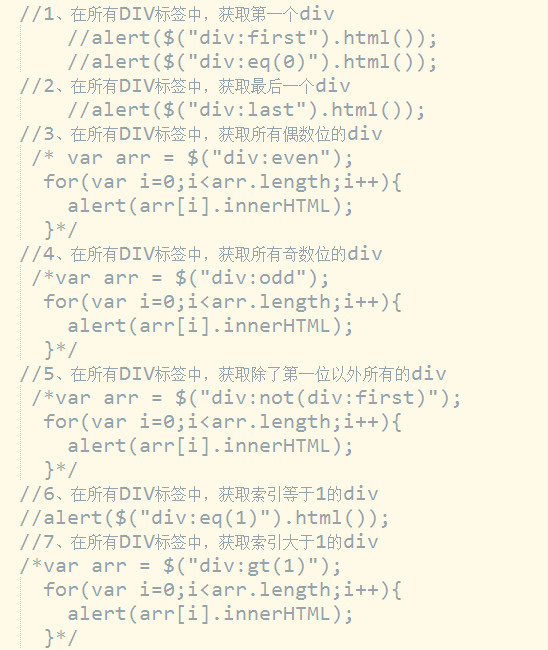
<div>44444444，索引是3</div>

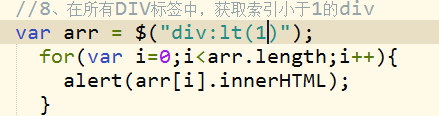
<div>55555555，索引是4</div>

<div>66666666，索引是5</div>

<div>77777777，索引是6</div>

<div>88888888，索引是7</div>





### 1.4.4、表单属性选择器

* **:checked**  选中
* **:selected** 选择
* :enabled 可用
* :disabled 不可用。

代码准备：

<script>

$(function(){

//获取可用的表单输入项

//获取不可用的表单输入项

//获取选中的复选框

//获取国家下拉框中，被选中的option

});

</script>

<h1>不可用的表单输入项</h1>

<input type="text" disabled="disabled" />

<input type="button" value="不可用按钮" disabled="disabled" />

<h1>复选框</h1>

<input type="checkbox" name="hobby" value="code"/>编程

<input type="checkbox" name="hobby" value="swimming" checked="checked"/>游泳

<input type="checkbox" name="hobby" value="reading" checked="checked"/>阅读

<hr />

<h1>下拉选择框</h1>

城市：<select id="city">

<option value="">请选择</option>

<option value="bj">北京</option>

<option value="sh">上海</option>

</select>

<hr />

国家：<select id="country" multiple="multiple">

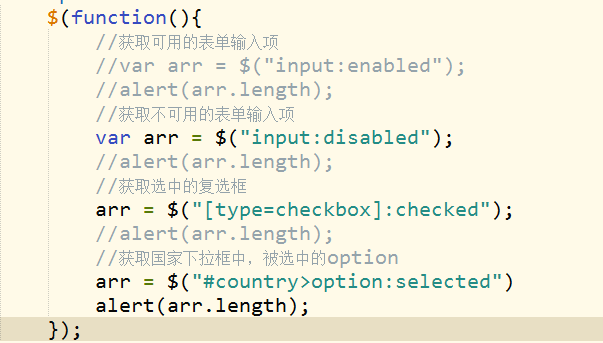
<option value="China" selected="selected">中国</option>

<option value="America" selected="selected">美国</option>

<option value="Russia">俄罗斯</option>

<option value="England">英国</option>

</select>



# 2、案例2：重写隔行换色

## 2.1、需求说明

开发中，需要使用表格陈列数据，数据过多不易查看，通常使用隔行换色进行表示。



## 2.2、知识讲解-jQuery（三）

### 2.2.1、jQuery操作class属性

addClass(“值”) 给元素对象追加样式

removeClass() 将元素对象的class删除

代码准备：

<style>

.blueDiv{

background-color: #87CEFA;

width:300px;

height:300px;

}

.redDiv{

background-color: red;

width:300px;

height:300px;

}

</style>

<script>

$(function(){

//1、设置d1这个div标签样式为blueDiv

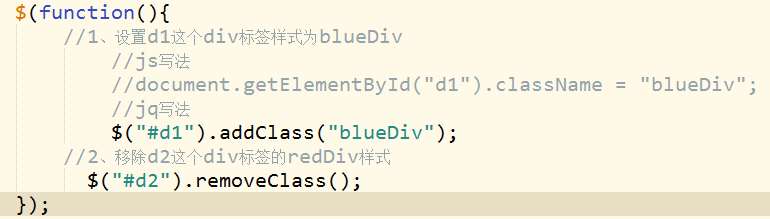
//2、移除d2这个div标签的redDiv样式

});

</script>

<div id="d1">11111</div>

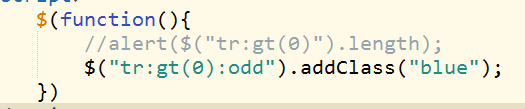
<div id="d2" class="redDiv">22222</div>



## 2.3、需求分析



## 2.4、案例代码实现



# 3、案例3：重写全选全不选、反选

## 3.1、需求说明及分析



## 3.2、知识讲解-jQuery(四)

* **prop("属性名");**  获取某个属性名对应的值 attr(“属性名”)
* **prop("属性名",属性值);** 将其属性名设置为某个属性值 **attr("属性名",属性值);**
* removeProp("属性名"); 移除改属性 removeAttr(“属性名”);

代码准备：

<style>

.blueDiv{

background-color: #87CEFA;

width:300px;

height:300px;

}

.redDiv{

background-color: red;

width:300px;

height:300px;

}

</style>

<script>

$(function(){

//1、获取d2这个div标签的class值

//2、设置d1这个div标签样式为blueDiv

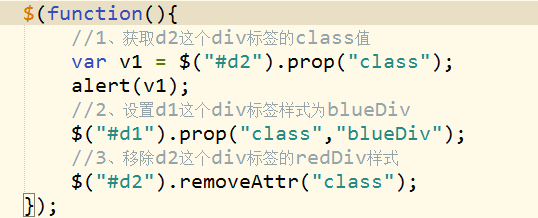
//3、移除d2这个div标签的redDiv样式

});

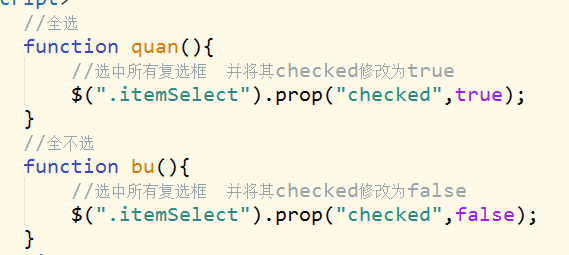
</script>

<div id="d1">11111</div>

<div id="d2" class="redDiv">22222</div>



## 3.3、案例代码准备及实现

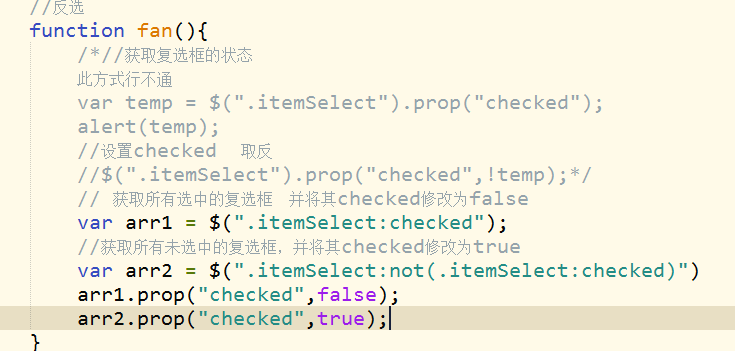


## 3.4、扩展-反选实现（一）

$(".itemSelect") 返回类型是数组

设置值：可以一次性将数组中的元素设置为相同的值(数组内部经过了遍历)

获取值：仅可以将下标为0的元素的属性值获取到



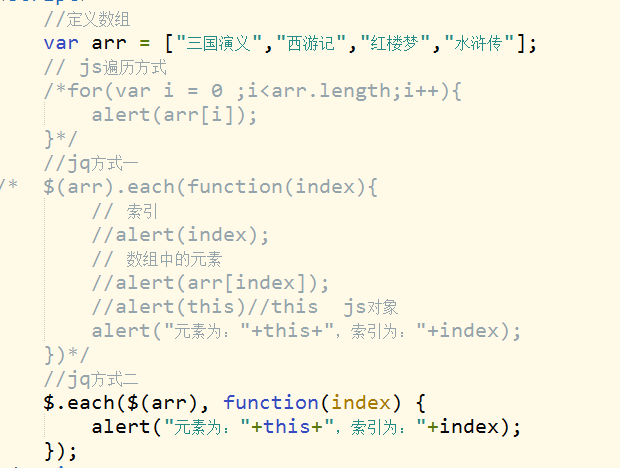
## 3.5、反选实现（二）

数组遍历：

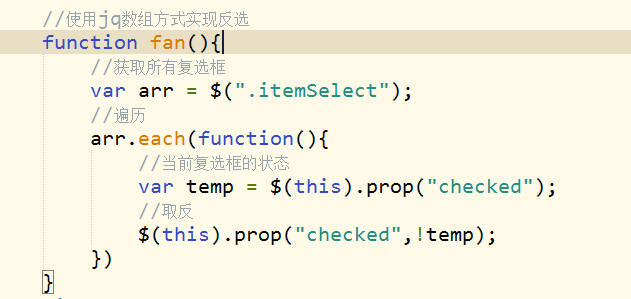
在JS中，我们使用普通FOR循环即可遍历数组。

在JQuery中，我们可以使数组的遍历变得更为简单

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **JQ数组.each(fn);** | 遍历JQ数组 |
| **$.each(JQ数组,fn);** | 遍历JQ数组 |

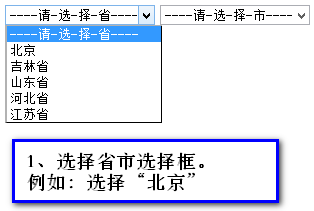


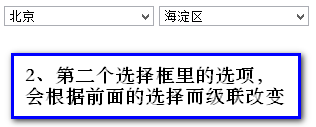
反选实现2：

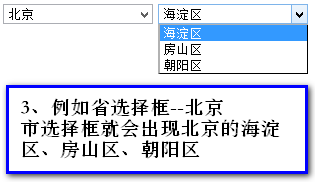


# 4：案例4：重写省市联动

## 4.1、需求说明及分析







## 4.2、知识讲解-jQuery(五)

### 4.2.1、元素value属性操作

jQuery中，还有一种快捷操作方式：

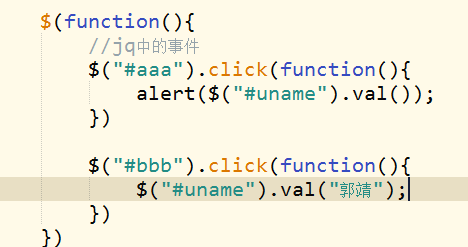
|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **JQ对象.val();** | 获取value属性的值  相当于Jq对象.prop(“value”); |
| **JQ对象.val(值);** | 设置value属性的值  相当于Jq对象.prop(“value”,”值”) |

准备代码：

<input type="text" id="uname" value="黄蓉" /><br />

<input type="button" value="点击获取输入框的value值" onclick="run1()" />

<input type="button" value="点击设置输入框的value值" onclick="run2()" />



### 4.2.2、获取/设置元素内容体

1、获取/设置元素内容体**HTML代码**

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **JQ对象.html()** | 获得内容体html代码  相当于 js 中 js对象.innerHTML |
| **JQ对象.html("HTML代码")** | 设置html代码  相当于js中 js对象.innerHTML = “” |

* 2、获取/设置元素内容体**纯文本**

|  |  |
| --- | --- |
| 属性名 | 属性说明 |
| **JQ对象.text()** | 获得文本，如果有标签，忽略。 |
| **JQ对象.text("纯文本")** | 设置文本，如果含有HTML标签，不进行解析。原样输出。 |

准备代码：

<div id="d1">

<font color="red">你好</font>

</div>

<hr />

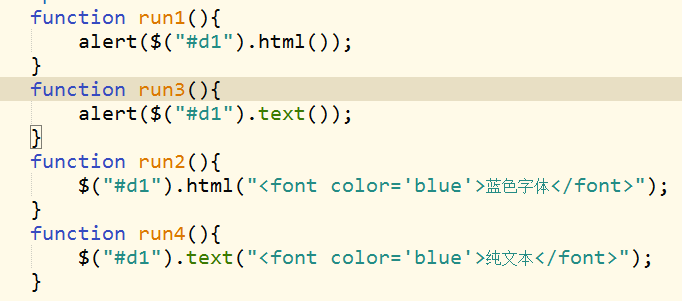
<input type="button" value="点我获取内容体代码" onclick="run1()"/>

<input type="button" value="点我设置内容体代码为蓝色字体" onclick="run2()"/>

<hr />

<input type="button" value="点我获取内容体纯文本" onclick="run3()"/>

<input type="button" value="点我设置内容体纯文本" onclick="run4()"/>

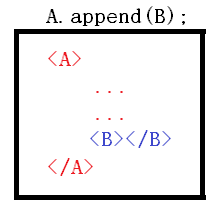


### 4.2.3、追加元素内容体

通常我们涉及到追加内容体，使得效果可以层层叠加。

以下两句效果相同，写法不同

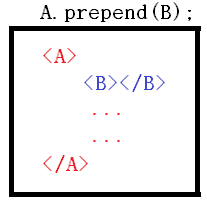
* **A.append(B)** , 向A的末尾追加B
* B.appendTo(A) ，将B追加到A的末尾



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

以下两句效果相同，写法不同

* **A.prepend(B)** , 向A的头部追加B
* B.prependTo(A) ，将B追加到A的头部



准备代码：

<ul id="rank">

<li>Php</li>

<li>Android</li>

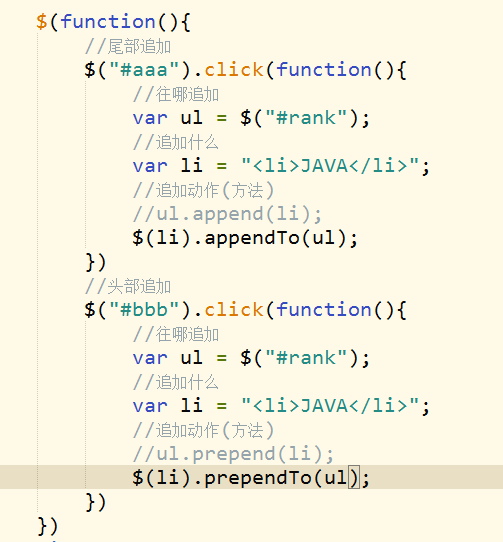
<li>Ios</li>

</ul>

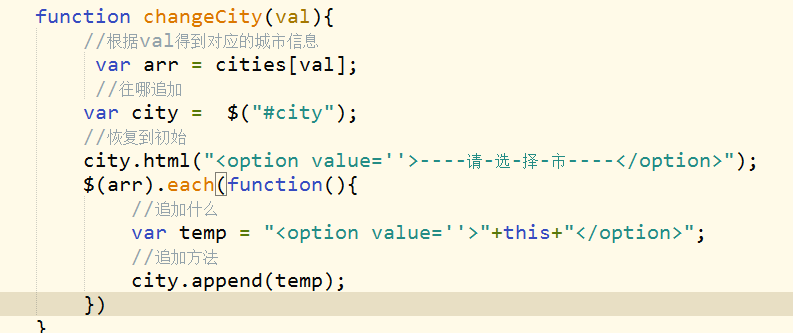
<hr />

<input type="button" value="列表尾部追加Java" onclick="run1()" />

<input type="button" value="列表头部追加Java" onclick="run2()" />

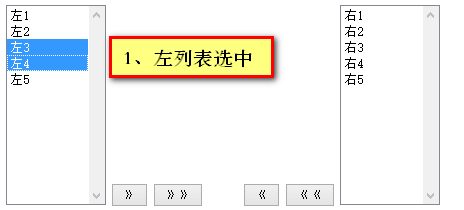


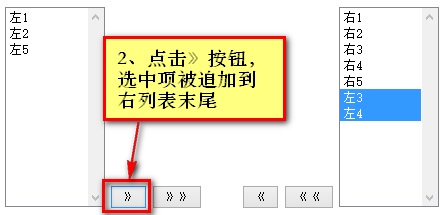
## 4.3、案例准备及代码实现

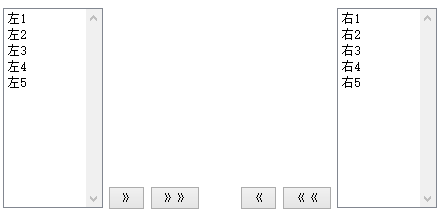


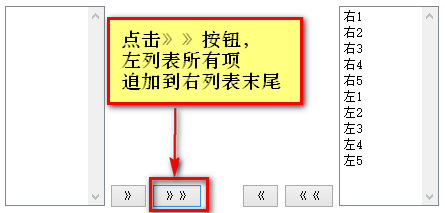
# 5、案例5：列表左右选择

## 5.1、需求说明及分析









## 5.2、案例实现

