# 更新

7月14日重新排版，更新了ajax在struts上的运用

# jQuery介绍

前提

为了简化 JavaScript 的开发, 一些 JavsScript 库诞生了. JavaScript 库封装了很多预定义的对象和实用函数。能帮助使用者建立有高难度交互的 Web2.0 特性的富客户端页面, 并且兼容各大浏览器

当前流行的 JavaScript 库有: jQuery, MooTools, Prototype, Dojo, YUI, EXT\_JS DWR

[www.open-open.com](http://www.open-open.com)开源网页

jQuery是一个快速的，简洁的javaScript库，使用户能更方便地处理HTML documents、events、实现动画效果，并且方便地为网站提供AJAX交互。

什么是jQuery对象？

jQuery 对象就是通过jQuery包装DOM对象后产生的对象。

jQuery 对象是 jQuery 独有的. 如果一个对象是 jQuery 对象, 那么它就可以使用 jQuery 里的方法: $(“#test”).html();

虽然jQuery对象是包装DOM对象后产生的，但是jQuery无法使用DOM对象的任何方法，同理DOM对象也不能使用jQuery里的方法.乱使用会报错

DOM对象转成jQuery对象

对于已经是一个DOM对象，只需要用$()把DOM对象包装起来，就可以获得一个jQuery对象了。$(DOM对象)

<input type=”text” name=”a” id=”b” value=”温老师”/>

//dom方式获得id为b的input标签

var b=document.getElementById("b");

//转换为jquery对象

var a=$(b);

alert(a.val());

jQuery对象转成DOM对象 两种方式

(1) jQuery对象是一个数组对象，可以通过[index]的方法，来得到相应的DOM对象

//通过id选择器获得id为b的input标签

<input type=”text” name=”a” id=”b” value=”温老师”/>

var a=$("#b");

//转换成dom对象，使用dom方法value

var b=a[0];

alert(b.value);

(2) jQuery本身提供，通过.get(index)方法，得到相应的DOM对象

//通过id选择器获得id为b的input标签

var a=$("#b");

//转换成dom对象，使用dom方法value

var b=a.get(0);

alert(b.value);

dom中的value方法出错通过alert没有提示，jquery的val()方法alert有提示

# jQuery 选择器

常用

第一个：通过表单中的name获取指定的标签元素

<input name="userName" type="text" phone" >

$("input[name='userName']").click(function(){alert('点击我做什么'); })

基本选择器

1、#id 用法: $(“#myDiv”); 返回单个元素的组成的集合

相当于dom中的方法：document.getElementById("myDiv ");

2、$(“div”) 返回值 集合元素

说明: element的英文翻译过来是”元素”,所以element其实就是html已经定义的标签元素,例如 div, input, a等等.

$("input").click(function(){

alert("我被点击了");

});

3、class 用法: $(”.myClass”) 返回值 集合元素

说明: 这个标签是直接选择html代码中class=”myClass”的元素或元素组(因为在同一html页面中class是可以存在多个同样值的).

4、\* 用法: $(”\*”) 返回值 集合元素

说明: 匹配所有元素,多用于结合上下文来搜索

5、$(“div,span,p.myClass”) 返回值 集合元素

说明: 将每一个选择器匹配到的元素合并后一起返回.你可以指定任意多个选择器, 并将匹配到的元素合并到一个结果内.其中p.myClass是表示匹配元素

层次选择器

1 、 (“form input”) ; 返回值 集合元素

说明: 在给定的祖先元素下匹配所有后代元素.这个要下面讲的”parent > child”区分开.

2、$(”form > input”) ; 返回值 集合元素

说明: 在给定的父元素下匹配所有子元素.注意:要区分好后代元素与子元素

3、$(”label + input”) ; 返回值 集合元素

说明: 匹配所有紧接在 prev 元素后的 next 元素

4、 $(”form ~ input”) ; 返回值 集合元素

说明: 匹配 prev 元素之后的所有 siblings 元素.注意:是匹配之后的元素,不包含该元素在内,并且siblings匹配的是和prev同辈的元素,其后辈元素不被匹配.

5、$("#two").siblings("div").css();

选择所有兄弟使用siblings方法来获取

## 3.过滤器

(1)基础过滤选择器

1、:first  
用法: $(”tr:first”) ; 返回值 单个元素的组成的集合

说明: 匹配找到的第一个元素

2、:last  
用法: $(”tr:last”) 返回值 集合元素

说明: 匹配找到的最后一个元素.与 :first 相对应.

3、:not(selector)  
用法: $(”input:not(:checked)”)返回值 集合元素

说明: 去除所有与给定选择器匹配的元素.有点类似于”非”,意思是没有被选中的input(当input的type=”checkbox”).

4、:even  
用法: $(”tr:even”) 返回值 集合元素

说明: 匹配所有索引值为偶数的元素，从 0 开始计数.js的数组都是从0开始计数的.例如要选择table中的行,因为是从0开始计数,所以table中的第一个tr就为偶数0.

5、: odd  
用法: $(”tr:odd”) 返回值 集合元素

说明: 匹配所有索引值为奇数的元素,和:even对应,从 0 开始计数.

6、:eq(index)  
用法: $(”tr:eq(0)”) 返回值 集合元素

说明: 匹配一个

7、:gt(index)  
用法: $(”tr:gt(0)”) 返回值 集合元素

说明: 匹配所有大于给定索引值的元素.

8、:lt(index)  
用法: $(”tr:lt(2)”) 返回值 集合元素

说明: 匹配所有小于给定索引值的元素.

9、:header(固定写法)  
用法: $(”:header”).css(”background”, “#EEE”) 返回值 集合元素

说明: 匹配如 h1, h2, h3之类的标题元素.这个是专门用来获取h1,h2这样的标题元素.

10、:animated(固定写法) 返回值 集合元素

说明: 匹配所有正在执行动画效果的元素

(2)内容过滤选择器

1、:contains(text)  
用法: $(”div:contains(’John’)”) 返回值 集合元素

说明: 匹配包含给定文本的元素.这个选择器比较有用，当我们要选择的不是dom标签元素时,它就派上了用场了,它的作用是查找被标签”围”起来的文本内容是否符合指定的内容的.

2、:empty  
用法: $(”td:empty”) 返回值 集合元素

说明: 匹配所有不包含子元素或者文本的空元素

3、:has(selector)

用法: $(”div:has(p)”).addClass(”test”) 返回值 集合元素

说明: 匹配含有选择器所匹配的元素的元素.这个解释需要好好琢磨,但是一旦看了使用的例子就完全清楚了:给所有包含p元素的div标签加上class=”test”.

4、:parent  
用法: $(”td:parent”) 返回值 集合元素

说明: 匹配含有子元素或者文本的元素.注意:这里是”:parent”,可不是”.parent”哦!感觉与上面讲的”:empty”形成反义词.

(3)可见度过滤选择器

1、:hidden  
用法: $(”tr:hidden”) 返回值 集合元素

说明: 匹配所有的不可见元素，input 元素的 type 属性为 “hidden” 的话也会被匹配到.意思是css中display:none和input type=”hidden”的都会被匹配到.同样,要在脑海中彻底分清楚冒号”:”, 点号”.”和逗号”,”的区别.

2、:visible  
用法: $(”tr:visible”) 返回值 集合元素

说明: 匹配所有的可见元素.

(4)属性过滤器

1、[attribute]  
用法: $(”div[id]“) ; 返回值 集合元素

说明: 匹配包含给定属性的元素. 例子中是选取了所有带”id”属性的div标签.

2、[attribute=value]  
用法: $(”input[name='newsletter']“).attr(”checked”, true); 返回值 集合元素

说明: 匹配给定的属性是某个特定值的元素.例子中选取了所有 name 属性是 newsletter 的 input 元素.

3、[attribute!=value]  
用法: $(”input[name!='newsletter']“).attr(”checked”, true); 返回值 集合元素

说明: 匹配所有不含有指定的属性，或者属性不等于特定值的元素.此选择器等价于:not([attr=value]),要匹配含有特定属性但不等于特定值的元素,请使用[attr]:not([attr=value]).之前看到的 :not 派上了用场.

4、[attribute^=value]  
用法: $(”input[name^=‘news’]“) 返回值 集合元素

说明: 匹配给定的属性是以某些值开始的元素.,我们又见到了这几个类似于正则匹配的符号.现在想忘都忘不掉了吧?!

5、[attribute$=value]  
用法: $(”input[name$=‘letter’]“) 返回值 集合元素

说明: 匹配给定的属性是以某些值结尾的元素.

6、[attribute\*=value]  
用法: $(”input[name\*=‘man’]“) 返回值 集合元素

说明: 匹配给定的属性是以包含某些值的元素.

7、[attributeFilter1][attributeFilter2][attributeFilterN]  
用法: $(”input[id][name$=‘man’]“) 返回值 集合元素

说明: 复合属性选择器,需要同时满足多个条件时使用.又是一个组合,这种情况我们实际使用的时候很常用.这个例子中选择的是所有含有 id 属性,并且它的 name 属性是以 man 结尾的元素.

(5)子元素过滤选择器

1、:nth-child(index/even/odd/equation)

用法: $(”ul li:nth-child(2)”) 返回值 集合元素

说明: 匹配其父元素下的第N个子或奇偶元素.这个选择器和之前说的基础过滤(Basic Filters)中的 eq() 有些类似,不同的地方就是前者是从0开始,后者是从1开始.

2、:first-child

用法: $(”ul li:first-child”) 返回值 集合元素

说明: 匹配第一个子元素.’:first’ 只匹配一个元素,而此选择符将为每个父元素匹配一个子元素.这里需要特别点的记忆下区别.

3、:last-child

用法: $(”ul li:last-child”) 返回值 集合元素

说明: 匹配最后一个子元素.’:last’只匹配一个元素,而此选择符将为每个父元素匹配一个子元素.

4、: only-child

用法: $(”ul li:only-child”) 返回值 集合元素

说明: 如果某个元素是父元素中唯一的子元素,那将会被匹配.如果父元素中含有其他元素,那将不会被匹配.意思就是:只有一个子元素的才会被匹配!

(6)表单对象属性过滤选择器

1、:enabled

用法: $(”input:enabled”) 返回值 集合元素

说明: 匹配所有可用元素.意思是查找所有input中不带有disabled=”disabled”的input.不为disabled,当然就为enabled啦.

2、:disabled

用法: $(”input:disabled”) 返回值 集合元素

说明: 匹配所有不可用元素.与上面的那个是相对应的.

3、:checked  
用法:

$(”input:checked”) 返回值 集合元素

说明: 匹配所有选中的被选中元素(复选框、单选框等，不包括select中的option).这话说起来有些绕口.

4、:selected  
用法:

$(”select option:selected”) 返回值 集合元素

说明: 匹配所有选中的option元素.

(7)表单选择器

1、:input

用法: $(”:input”) ; 返回值 集合元素

说明:匹配所有 input, textarea, select 和 button 元素

2、:text

用法: $(”:text”) ; 返回值 集合元素

说明: 匹配所有的单行文本框.

3、:password

用法: $(”:password”) ; 返回值 集合元素

说明: 匹配所有密码框.

4、:radio

用法: $(”:radio”) ; 返回值 集合元素

说明: 匹配所有单选按钮.

5、:checkbox

用法: $(”:checkbox”) ; 返回值 集合元素

说明: 匹配所有复选框

6、:submit  
用法:

$(”:submit”) ; 返回值 集合元素

说明: 匹配所有提交按钮.

# AJAX

Ajax本质上和普通的HTTP请求是一样的，只不过普通的HTTP请求是给人看的，而Ajax请求是给JS代码去用的

struts2获取值栈中的数据

在Struts 2中使用Ajax，Action一般就不会调用一个jsp视图来显示了（如果Ajax请求内容是页面的一部分也可能调用jsp视图），而是通过一些其他的方式。

第一步：导包

struts2-json-plugin-x.x.x.x.jar 在struts-2.3.24\lib可以找到

第二步：在核心配置文件package中extends应该改为json-default，result的type设为json

json-default是已经继承了strtus-default的

<package name="user" namespace="/" extends="json-default">

<action name="verific\_\*" class="verificAction" method="{1}">

<result name="userNameStatus" type="json">/register.jsp</result>

</action>

</package>

第三步 往值栈中放值：

public String verificUserName() {

ValueStack ac = ActionContext.getContext().getValueStack(); //创建值栈对象

List<Users> users=us.serviceUserNsme(userName);

if(users.size()>0) {

ac.set("nameStatus", users.get(0));//往值栈放值

}else {

ac.set("nameStatus", "no");

}

return "userNameStatus";//找第二步中配置的result

}

第四步 只能在js中的ajax才能取值

$.post(

"${pageContext.request.contextPath}/verific\_verificUserName", //url

{userName:n}, //这里不知道为什么一定要有中括号

function(data){ //只能这里取值，（不使用ajax可以直接网页上标签取值）

alert(data.nameStatus.userName);//这里的值已经转换为json格式

if(data.nameStatus =='no'){

imgs.src="images/biao\_duigouh.jpg";

}else{

imgs.src="images/biao\_X\_h.jpg";

}

},"json"

);});

});