jQuery JS库

**语法**

**$(selector).action( );**

工厂函数$( )：将DOM对象转化为jQuery对象

选择器selector：获取需要操作的DOM元素

方法action( )：jQuery中提供的方法，其中包括绑定事件处理的方法

$(“h2”).css(“background-color”,“#CCFFFF”);

**一.$(callbackFn)**

$(document).ready(callbackFn);当页面中所有DOM文件结构绘制完毕后即刻执行，可能与DOM元素关联的内容（图片，flash，视频等）并没有加载完

可以缩写成**$(callbackFn)**

同一个页面可以多次使用

确保在<body>元素的onload事件中没有注册函数，否则不会触发

$(function(){

$("#A").css("color","#f40");

})

<div id="A">AAAAAAA</div>

**二.$和jQuery的关系**

$=jQuery

$(function( ){ })和jQuery(function( ){ })一样

$是jQuery的别名

如果$命名冲突，则不会认为是执行jQuery代码

许多JavaScript库都用$作为标记，所以同时使用多个库时，会出现冲突

**解决方法：var jq=jQuery.noConflict( );**然后用jq代替了$

**三,jQuery对象**

**1**.jQuery对象是通过jQuery包装DOM对象后产生的对象

**2**.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$(selector).action( );

**3**.jQuery对象无法使用DOM对象的任何方法，同样DOM对象也不能使用jQuery对象里的任何方法

**4**.约定：如果获取的是jQuery对象，那么要在变量前面加$

var $variable = jQuery对象

var variable = DOM对象

**四.DOM对象转成jQuery对象**

$(DOM对象)

<div id="C">C</div>

var DOMC = document.getElementById('C');

DOMC.innerHTML

var jQueryC = $(DOMC);

jQueryC.html( );**改值就传参数**

**五.jQuery对象转成DOM对象**

var jQueryD = $(‘#mydiv’);实际上是一个数组，这里数组只包含一个元素

var DOMD = jQueryD[0];

var DOMD = jQueryD.get(0);

**六.jQuery语法的特点**

**1.读写使用同一方法**  
读.$(‘#E’).html( );

写.$(‘#E’).html(‘///’);

读.$(‘#E’).css(‘color’);

写.$(‘#E’).css(‘color’,‘red’);

**2.链式操作**

$(‘#F’).html(‘///’).css({

‘color’:‘red’，

‘border’: ‘1px solid red’

}).width(///).height(///);

**七.jQuery选择器**

选择器是jQuery的根基，在jQuery中，对事件处理，遍历DOM和Ajax操作都依赖于选择器

jQuery选择器的优点

写法简介 $(‘#G’);根据id选择

完善的事件处理机制

val( )获取第一个元素的当前值

选择器分类：基本选择器和过滤选择器

1.**基本选择器**是jQuery中最常见的选择器，也是最简单的选择器

$(“#id”)

$(“.class”)

$(“element”)

$(“\*”)

$(“selector1,selector2”)

$(“element.class”)并列

2.过滤选择器主要是通过特定的过滤规则来筛选出所需的DOM元素，过滤选择器都以**冒号开头**

3.按照不同的过滤选规则，过滤选择器可以分为6种：基本过滤、内容过滤、可见性过滤、属性过滤、子元素过滤和表单对象属性过滤器

**（一）层次选择器**

$(“ancestor descendant”)选取ancestor的所有descendant（后代）元素

$(“parent>child”)选取parent元素下的子元素child元素

$(“prey+next”)选取prey元素后的下一个next元素

$(“prey~siblings”)选取prey元素后所有siblings元素

注：只选择prey元素后面的同辈元素；而jQuery中的siblings（）与前后位置无关，只要值同辈节点就可以选取

**（二）基本过滤选择器**

$(“span:frist”)选取第一个元素

$(“span:last”)选取最后一个元素

$(“span:not(selector)”)去掉selector元素剩下的元素

$(“span:even”)偶数位的元素

$(“span:odd”)奇数位元素

$(“span:eq(index)”)等于index索引的元素

$(“span:gt(index)”)大于index索引的元素

$(“span:lt(index)”)小于index索引的元素

$(“:header”)选取所有标题元素h1 h2

$(“:animated”)选取当前正在执行动画的所有元素

**（三）内容过滤选择器**

$(“span:contains(text)”)选取文本内容含有text的元素

$(“span:empty”)选取不含子元素或者文本的空元素

$(“span:has(selector)”)选取含有选择器所匹配的元素的元素

$(“span:parent”)选取含有子元素或者文本的元素

**（四）属性选择器**

$(“span[attribute]”)获取有attribute属性的元素

$(“span[attribute=value]”)获取此属性值为value的元素

$(“span[attribute!=value]”)获取此属性值不为value的元素

$(“span[attribute^=value]”)获取值以value值开始的元素

$(“span[attribute$=value]”)获取值以value值结束的元素

$(“span[attribute\*=value]”)获取值中含有value值的元素

**（五）子元素选择器**

$(“div:nth-child(index/even/odd/equation)”)选取每个父元素下的第index个子元素或者奇偶元素（index从1算起）

$(“div:first-child”)父级下的第一个子元素

$(“div:last-child”)父级下的最后一个子元素

$(“div:only-child”)父级下的子元素如果唯一，返回这个子元素

**（六）可见性过滤选择器**

可见性过滤选择器是根据元素的可见还是不可见状态来选择相应的元素

可见选择器：hidden不仅仅包含样式属性**display:none**的元素，也包含文本隐藏域（<input type = “hidden”>）和**visibility:hidden**(<span></span>)之类的元素

$(“span:hidden”)选取所有不可见元素

$(“span:visible”)选取所有可见元素

**（七）表单选择器**

$(“form:input”)选取所有<input><textarea><select><button>元素

$(“form:text”)选取所有单行文本框 type=text

$(“form:password”)type=password

$(“form:radio”)type=radio

$(“form:checkbox”)type=checkbox

$(“form:submit”)type=submit

$(“form:reset”)type=reset

$(“form:button”)type=button

$(“form:image”)type=image

$(“form:file”)type=file

$(“form:enabled”)选取所有可用（激活）元素

$(“form:disable”)选取所有不可用（未激活）元素

$(“form:checked”)选取所有被选中的元素（单选复选）

$(“form:selected”)选取所有被选中的元素（下拉列表）

**八.jQuery中的事件**

**（一）基础事件**

**1.window事件**

**2.鼠标事件**

（1）click（）单击鼠标

（2）mouseover（）鼠标光标移入

（3）mouseout（）鼠标光标移出

<style>.over{color: #f40;}</style>

$('#A').mouseover(function(){

$(this).addClass('over');

}).mouseout(function(){

$(this).removeClass('over');

})

addClass（）添加css样式

removeClass（）删除css样式

（4）mousedown（）鼠标按下

（5）mouseup（）鼠标抬起

**3.键盘事件**

（1）keydown（）按下

（2）keyup（）释放

（3）keypress（）按下产生可打印的字符时

$(document).keydown(function(event){

console.log(event.keyCode);

})

keyCode返回按键的asc码不区分大小写

charCode区分大小写（只作用于keypress（））

**4.表单事件**

（1）focus（）获得焦点

（2）blur（）失去焦点

**（二）复合事件**

**1.鼠标光标悬停**

（1）hover（fn1，fn2）相当于mouseover和mouseout事件的组合

参数是两个匿名函数

一个是鼠标经过一个是鼠标移开

**2.鼠标连续点击**

（1）toggle（fn1，fn2，fn3...）用于鼠标连续点击，在1.9版本以后都被删除

里面可以有n个匿名函数作为参数，按顺序表示每次点击时要执行的函数

**九.绑定事件**

（1）bind（）1.7之前的版本

（2）on（）建议用on（）

两者一样，里面有两个参数，一个绑定的时间类型，一个回调函数

$(document).on(‘click’,function( ){ } )

可以同时绑定多个事件传入一个json的类似

$(document).on({

mousedown : function( ){ },

mouseup : function( ) { }

})

**十.移除事件**

（1）unbind([type],[fn]);

[type]事件类型，主要包括blur focus click mouseup等基础事件，也可以是自定义事件

[fn]处理函数，用来绑定的处理函数

当不带参数表示移出所有绑定的全部事件

**十一.表单的提交事件**

（1）submit( );

1.提交表单

$(‘form’).submit( );

$(‘form’).submit(function( ){

return true;

});

2.阻止表单的提交

$(‘form’).submit(function( ){

return false;

});

**十二.jQuery的DOM操作**

DOM分三种

（一）DOM Core任何一种支持DOM的编程语言都可以用它，如getElementById( )

（二）HTML-DOM用于处理HTML文档，如document.forms

（三）CSS-DOM用于操作CSS如element.style.color = ‘red’;

注：JavaScript用于对（x）html文档进行操作，它对这三类DOM操作都支持

**jQuery-DOM操作**

**（一）样式操作**

（1）css( ) 设置或返回样式属性

（2）addClass( ) 向被选元素添加一个或多个类

（3）removeClass( ) 从被选元素删除一个或多个类

（4）toggleClass( ) 对被选元素进行添加/删除类的切换操作

**（二）内容及Value属性值操作**

（1）html( )

不传参数表示获取第一个匹配元素的HTML内容或文本内容

传参表示设置HTML内容

（2）text( )

不传参数表示获取所有匹配元素的文本内容

传参表示设置TEXT内容

（3）val( )

不传参获取value值，传参是修改value值

（4）attr( )

传一个参数表示获得属性的值

两个参数表示设置属性的值

**（三）节点操作**

**1.查找节点**

$(///)

**2.创建节点**

$(html)

var div = $('<div id="1">123</div>');

可以直接把文本内容和节点属性写在里面

**3.插入节点**

（1）append( ) 在被选元素的**结尾**插入内容

$('ul').append(li)

（2）appendTo( )

li.appendTo($('ul'))

（3）prepend( ) 在被选元素的**开头**插入内容

$('ul').prepend(li)

（4）prependTo( )

li.prependTo($('ul'))

（5）after( ) 在被选元素之**后**插入内容

$('ul').after(p)在ul后面加入p

（6）before( ) 在被选元素之**前**插入内容

$('ul').before(p)在ul前面加入p

（7）insertBefore( )

p.inertBefore($('ul'))

（8）insertAfter( )

p.inertAfter($('ul'))

**4.删除节点**

（1）remove( )

$('ul').remove( ) 有返回值是被删的节点

（2）empty( )清空节点

$('ul').empty( )清空节点里面的节点，本身还在，没有返回值

（3）detach( )仅仅是不显示，但还存在

**5.复制节点**

（1）clone( )单纯复制节点(只是复制没有添加所以要结合添加)

$('li').clone( ).appendTo($('ul'));

（2）clone(true)不仅复制节点同样也复制绑定在节点中的事件

**6.替换节点**

（1）replaceWith( )

$(A).replaceWith(B)把B替换A

将所有匹配的元素都替换为指定的HTML或DOM元素

（2）replaceAll( )

B.replaceAll($(A))

与上面的方法颠倒

替换之前，元素绑定了事件，替换后事件也会随原来的元素一起消失

**（四）节点属性操作**

（1）attr( )设置属性和方法，一个参数表示获取属性的值，两个参数表示修改属性的值

（2）removeAttr( )删除指定元素的属性

（3）height( )设置或返回元素的高度（不包括内边距、边框或外边距）

（4）width( )设置或返回元素的宽度（不包括内边距、边框或外边距）

（5）innerWidth( )方法返回元素的宽度（包括内边距）

（6）innerHeight( )方法返回元素的高度（包括内边距）

（7）outerWidth( )方法返回元素的宽度（包括内边距和边框）

（8）outerHeight( )方法返回元素的高度（包括内边距和边框）

**（五）节点遍历**

（1）children( )获取所有子节点不再考虑后代节点

（2）parent( )获取父节点

（3）parents( )获取祖先节点

（4）next( )下一个元素

（5）prev( )上一个元素

（6）siblings( )同级所有元素

（7）1.$.each(A,function(index,item){///})A表示数组

2.A.each(function(index,item){///})

**十三.jQuery内置动画效果**

**（一）显示隐藏(大小和透明度的变化)**

（1）**显示**show([speed],[callback])

show(3000,function( ){ })

speed是效果的持续时间，毫秒或者预定义字符串（slow、normal、fast）如果传入这个参数会立刻显示，没有过度过程。

callback动画**结束时**自动调用的方法函数

（2）**隐藏**hide([speed],[callback])

speed是效果的持续时间，毫秒或者预定义字符串（slow、normal、fast）如果传入这个参数会立刻隐藏，没有过度过程。

callback动画**结束时**自动调用的方法函数

**（二）淡入淡出**

（1）**淡入**fadeIn([speed],[callback])

（2）**淡出**fadeOut([speed],[callback])

**（三）滑上滑下（像窗帘）**

（1）上滑slideUP([speed],[callback])

（2）下滑slideDown([speed],[callback])

**（四）自定义动画**

（1）$(selector).animate({params},speed,[callback])

params参数为制作动画效果的CSS属性名和属性值，必选

speed效果完成时间，记毫秒，可选

callback回调函数可选

$('div').animate({

width:"400px",

color:"red"

},3000,function(){})

常用的jQuery

jQuery.each(obj,callback);遍历对象和数组

jQuery.map();修改数据

jQuery.grep();数据筛选,返回一个经过筛选后的数组

jQuery.inArray(value,array);查找元素的下标

jQuery.merge(array1,array2);合并两个数组

jQuery.unique(dom);去除重复DOM元素

jQuery.makeArray(obj);将类数组对象转换为数组对象

jQuery.trim(str);//去掉字符串起始和结尾的空格

jQuery.contains(dom1,dom2);dom1节点是否包含dom2节点

jQuery.type();返回对象的数据类型

jQuery.isArray();是否为数组。

jQuery.isEmptyObject();是否为空对象（不含可枚举的属性）。

jQuery.isFunction();否为函数。

jQuery.isNumeric();是否为数组。

jQuery.isPlainObject();是否为使用“{}”或“new Object”生成的对象，而不是浏览器原生提供的对象。

jQuery.isWindow();是否为window对象。

jQuery.isXMLDoc();判断一个DOM节点是否处于XML文档之中。

jQuery.param(object);将对象的键值对转化为URL键值对字符串形式

jQuery.proxy();调整this的指向

jQuery的AJAX方法

$(function () {

var list = {}; 请求参数

$.ajax({

type: "POST", 请求方式

contentType: "application/json;charset=UTF-8",请求的媒体类型

url: "http://127.0.0.1/admin/list/",请求地址

data: JSON.stringify(list), 数据，json字符串

success: function (result) {请求成功

console.log(result);

},

error: function (e) {请求失败，包含具体的错误信息

console.log(e.status);

console.log(e.responseText);

}

});

});

1.**url** 必需。规定把请求发送到哪个 URL。

2.**data**可选。映射或字符串值。规定连同请求发送到服务器的数据。

3.**success(data, textStatus, jqXHR)** 可选。请求成功时执行的回调函数。

4.**dataType**可选。规定预期的服务器响应的数据类型。默认执行智能判断（xml、json、script 或 html）。

**$.ajax({**

**type: 'POST',**

**url: url ,**

**data: data ,**

**success: success ,**

**dataType: dataType**

**});**

**$.ajax需要注意的一些地方**

1.data主要方式有三种，html拼接的，json数组，form表单经serialize()序列化的；通过dataType指定，不指定智能判断。

2.$.ajax只提交form以文本方式，如果异步提交包含<file>上传是传过不过去,需要使用jquery.form.js的$.ajaxSubmit

**1.$.ajax带json数据的异步请求**

var aj = $.ajax( {

url:'productManager\_reverseUpdate',跳转到 action

data:{

selRollBack : selRollBack,

selOperatorsCode : selOperatorsCode,

PROVINCECODE : PROVINCECODE,

pass2 : pass2

},

type:'post',

cache:false,

dataType:'json',

success:function(data) {

if(data.msg =="true" ){

// view("修改成功！");

alert("修改成功！");

window.location.reload();

}else{

view(data.msg);

}

},

error : function() {

// view("异常！");

alert("异常！");

}

});

**2.$.ajax序列化表格内容为字符串的异步请求**

function noTips(){

var formParam = $("#form1").serialize();序列化表格内容为字符串

$.ajax({

type:'post',

url:'Notice\_noTipsNotice',

data:formParam,

cache:false,

dataType:'json',

success:function(data){

}

});

}

**3.$.ajax拼接url的异步请求**

var yz=$.ajax({

type:'post',

url:'validatePwd2\_checkPwd2?password2='+password2,

data:{},

cache:false,

dataType:'json',

success:function(data){

if( data.msg =="false" ) 服务器返回false，就将validatePassword2的值改为pwd2Error，这是异步，需要考虑返回时间

{

textPassword2.html("<font color='red'>业务密码不正确！</font>");

$("#validatePassword2").val("pwd2Error");

checkPassword2 = false;

return;

}

},

error:function(){}

});

**4.$.ajax拼接data的异步请求**

$.ajax({

url:'<%=request.getContextPath()%>/kc/kc\_checkMerNameUnique.action',

type:'post',

data:'merName='+values,

async : false, 默认为true 异步

error:function(){

alert('error');

},

success:function(data){

$("#"+divs).html(data);

}

});