**$()下的常用方法**

**一、addClass() removeClass() hasClass() toggleClass()**

作用：查找、添加和删除class

oDiv.addClass('pzh pzh1')

oDiv.hasClass('pzh') 这其实就是 is("." + class)

oDiv.removeClass('pzh')

**二、width() innerWidth() outerWidth()**

作用：元素的宽度（同理，还有高度）

$('div').width() 元素的宽，不带单位，默认只有像素，所以没有单位

$('div').innerWidth() //width + padding

$('div').outerWidth() //width + padding + border

$('div').outerWidth(true) //width + padding + border + margin

注意它们同css方法的区别，css方法返回的有单位。

$('div').css('width'); 元素的宽，带单位

以上除了获取宽度，也可以再设定宽度做为参数，如：

$('div').width(200); //width ：200

$('div').innerWidth(200); //width ：200 - padding

$('div').outerWidth(200) //width ：200 - padding - border

$('div').outerWidth(200,true) //width ：200 - padding - border - margin

注意：原生js获取不到隐藏元素的尺寸，而jq方法是可以获取到隐藏元素的尺寸的

// 可视区的宽高

var w = $(window).width();

var h = $(window).height();

alert(w)

alert(h);

// 文档的宽高

var w1 = $(document).width();

var h1 = $(document).height();

alert(w1)

alert(h1);

**三、创建元素**

$('<div>') //仅仅只是创建一个div元素

注意：如果没有左右尖角号，则是选取div元素。

$('<div>div</div>'); //创建元素时，同时有内容

**clone()**

作用：克隆节点。

$('div').clone().appendTo( $('span') );

它还可以接收一个参数true，这样就可以复制之前的操作行为

$('div').click(function(){

alert(123);

});

$('div').clone(true).appendTo( $('span') );

**四、节点的操作**

insertBefore() before()

<div>div</div>

<span>span</span>

$('span').insertBefore($('div')); span添加到div的前面。（前面的元素是被操作的，后面的元素是参数物）

$('span').before($('div'));div添加到span的前面。（后面的元素是被操作的，前面的元素是参照物）

注意这两方法的区别。

（思考：为什么会有这样两个方法，一个不就行了嘛？）

关键是后续的操作不一样。因为有时候我们不是光把节点切换了就完事了，链式操作还要改变别的东西。如改变样式。（后续操作的都是前面的哪个元素）

insertAfter() after()

appendTo() append()

prependTo() prepend()

注意它们四组方法的区别：

单词长一点的：前面的元素是被操作的，后面的元素是参照物

单词短一点的：后面的元素是被操作的，前面的元素是参照物

**五、remove()**

作用：删除元素，该方法有返回值，返回被删除的元素。当删除的元素再次使用时，它之前绑定的事件就没有了。

如：

$('div').click(function(){

console.log(123);

});

var oDiv = $('div').remove();

$('body').append( oDiv );

**detach()：**跟remove方法一样，只不过会保留删除这个元素的操作行为

$('div').click(function(){

console.log(123);

});

var oDiv = $('div').detach();

$('body').append( oDiv );

**清空empty() html(‘’)**

**替换节点**

**replaceWith()**

**replaceAll()**

$('div').replaceAll($('span')); // 前面的替换后面的

$('div').replaceWith($('span')); // 后面的替换前面的

replace同其它的节点操作相反，它是长的前面是参照物

**六、on() off()**

作用：事件的另外一种写法

之前：$('div').click(function(){});

现在用on： $('div').on('click',function(){});

好处：

1、不仅可以是系统事件，也可以是自定义的事件（用trigger触发）

2、同时可以写多个事件。

$('div').on('click mouseover',function(){});

3、还可以写成json的形式

$('div').on({

'click' : function(){

alert(123);

},

'mouseover' : function(){

alert(456);

}

});

off()取消事件。如果没有参数，则取消全部的，如果有参数，则取消参数的这个事件。

$('div').on('click mouseover',function(){

alert(123);

$('div').off(); // 取消全部的事件

});

$('div').on('click mouseover',function(){

alert(123);

$('div').off('mouseover'); // 只取消mouseover的事件

});

on是bind delegate live三者的替代

**七、one()**

事件只执行一次。

$('div').one('click',function(){ //只执行事件一次

alert(123);

});

**九、scrollTop()、scrollLeft()**

横向、纵向滚动条

$(window).on('scroll', function() {

console.log($(window).scrollTop()); // 滚动要的高度

});

另外，即可以获取，也可以赋值

$(window).scrollTop(200)

**十、可视区宽高**

$(window).width() 窗口的宽（可视区）

$(window).height() 窗口的高（可视区）

$(document).width() 页面的宽（文档）

$(document).height() 页面的高（文档）

**十一、is**

元素.is(参数)，元素符合参数，即返回true。

如：

<form><input type="checkbox" /></form>

$("input[type='checkbox']").parent().is("form")

**案例：弹窗效果**

参见：弹窗.html