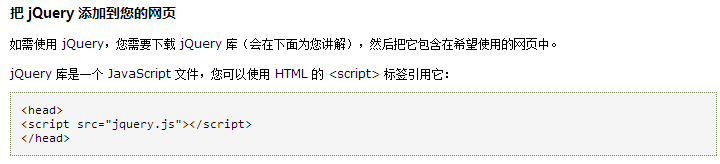
jQuery是一个JS库，它极大地简化了JS编程。



1、基础语法

jQuery 语法是为 HTML 元素的选取编制的，可以对元素执行某些操作。

基础语法是：$(selector).action()

- 美元符号定义 jQuery

- 选择符（selector）“查询”和“查找” HTML 元素

- jQuery 的 action() 执行对元素的操作

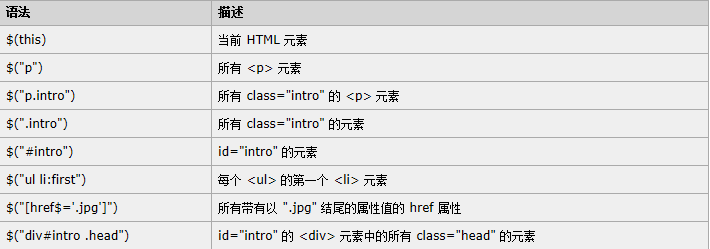
让出$操作符

jQuery.noConflict()

让出$后，jQuery依然能够运行，用jQuery代替$即可（只要引入了jQuery，使用jQuery(selector).action()就是可以的）



2、属性选择器



3、一些界面操作

$(selector).hide(speed,callback)/show()        toggle()        //隐藏显示

fadeIn()/fadeOut()            fadeToggle()        //淡入淡出        $(selector).fadeTo(speed,opacity,callback)            //可指定具体的透明度

slideDown()/slideUp()        slideToggle()        //滑动隐藏和显示

默认时间400ms,slow=200ms,fast=600ms

默认是swing,中间移动快，开头结尾移动慢，可修改为linear，匀速移动

$(selector).animate({params},speed,callback)            //自定义动画

{params}是一个或多个css属性的键值对所构成的Object对象。

callback前面不加参数默认是complete:每次执行完后执行，加step:后，每一个动画都会执行，可动态监控改变状态，加progress:后，每次动画调用后会执行回调，可以获取到进度

$(selector).stop(stopAll,goToEnd)    //不带参数的，停止当前动画                // 有一个true的，停止并清空整个动画队列            //有2个true的，立即完成所有动画

4、获取DOM中的内容

text() - 设置或返回所选元素的文本内容

html() - 设置或返回所选元素的内容（包括 HTML 标记）

val() - 设置或返回表单字段的值

加参数是设置，不加是获取

attr(attribute) - 获取属性值        //比如要获取value,直接用attr("value")即可

要修改属性时，使用attr(attribute,value)            //可传入对象一次设置多个属性的值，如attr({attribute : value, attribute : value})

还可以使用回调函数 attr(attribute, function(i, origValue){ return origValue};        //i为当前元素下标，origValue为属性原来的值，return的值为准备赋予的值

5、添加/删除元素

append()        //在被选元素的结尾插入内容

prepend()        //在被选元素的开头插入内容

after()        //在被选元素之后插入内容

before()        //在被选元素之前插入内容

wrap()            //把每个被选的元素防止在指定的HTML内容或元素中，如：$("p").wrap("<div></div>")会把每个p标签放到div中去

function appendText()

{

var txt1="<p>Text.</p>"; // 以 HTML 创建新元素

var txt2=$("<p></p>").text("Text."); // 以 jQuery 创建新元素

var txt3=document.createElement("p"); // 以 DOM 创建新元素

txt3.innerHTML="Text.";

$("p").append(txt1,txt2,txt3); // 追加新元素

}

remove()            //删除被选元素以及子元素

empty()            //删除元素的子元素

可加参数，参数为jQuery选择器语法，用来过滤删除的元素比如下面

$("p").remove("#abc")        //这个会删除p标签中id为abc的元素，等价于$("p#abc").remove();

6、操作CSS

addClass()                //向被选元素添加一个或多个类

removeClass()            //从被选元素删除一个或多个类

toggleClass()            //对被选元素进行添加/删除类的切换操作

css()                            //设置或返回样式属性（可通过参数指定具体的CSS属性的值）

使用css("propertyname")读取对应属性的值，使用css("propertyname","value")或css({"propertyname":"value","propertyname":"value"})来设置属性的值

使用addClass()增加类来获得的属性对应值，优先级低于使用css()方法赋予的值，即当元素的类中和样式属性中同时设置了一个css属性的值，生效的将是直接设置样式的，而不是class中的样式

$(document).ready(function(){

  $("button").click(function(){

    $("h1").css("color","red");

    $("h1").addClass("blue");

  });

});

</script>

<style type="text/css">

.blue

{

color:blue;

}

此时h1标签的字的颜色应该是红色的，标签属性如下

<h1 class = "blue" style = "background-color: red">test</h1>

设置尺寸

width()        //设置或返回元素的宽度（不包括内边距、边框和外边距）

height()        //设置或返回元素的高度（不包括内边距、边框和外边距）

innerWidth()        //返回元素的宽度（包括内边距）

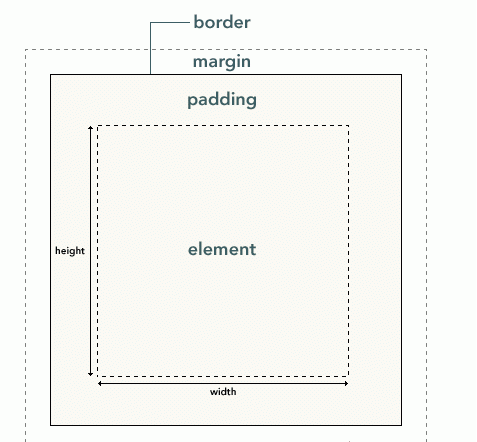
innerHight()        //返回元素的宽度（包括内边距）

outerWidth()

outerHight()        //包括内边距和外边距

outerWidth(true)

outerHight(true)        //包括内边距、边框和外边距



padding:内边距（填充）

border:边框

margin:外边距（默认是透明的）

7、遍历

parent()        //返回元素的直接父元素

parents()        //返回元素的所有祖先元素 （参数可以为选择器格式，用来过滤找到特定的元素）

parentsUntil()            //返回介于两个元素之间的所有祖先元素，参数为选择器格式，来指定查询到哪个父类为止

children()            //返回被选元素的所有直接子元素（参数可以用选择器格式进行过滤）

find()                //返回被选元素的所有后代元素（参数必填，不填查不出来，使用选择器格式，如果需要返回所有，参数填"\*"）

siblings()        //返回被选元素的所有同胞元素（参数可以用选择器格式进行过滤）

next()             //返回被选元素的下一个同胞元素

nextAll()            //返回被选元素后面的所有同胞元素（参数可以用选择器格式进行过滤）

nextUntil()            //返回介于两个元素之间的所有同胞元素（不填参数会查出所有后面的同胞元素，填写后会查两者之间的）

prev(),prevAll(),prevUntil()与上面next()的用法相同，只是找的是前面的同胞元素

first()            //返回被选元素的首个元素

last()            //返回被选元素的最后一个元素