第一章

一、$(document).ready(function(){

//do something

});

和window.onload的区别

1. 前者同一页面可以使用多次，后者出现多次只有最后一次能正确显示
2. 前者在DOM模型加载完成后就开始执行，后台在所有元素（包括图片）加载完成后执行
3. 前者可以简化为$(function(){

//do something

});

后者没有简写形式。

二、jQuery比较常使用的是链式操作，比如一个书目导航条：

$(function(){

$(“ul”).click(function(){

$(this).addClass(“current”).children(“li”).slideDown().end().sublings().removeClass(“current”).children(“li”).slideUp();

})

})

对于同一元素不超过3个操作的，一般写同一行，如果操作过多，可以将类似的操纵写成一行，便于阅读。

三、jQuery对象和DOM对象的转换

1.jQuery转化为DOM对象

Var $variable = $(“tr”); //jQuery object

Var variable = $variable[0]; //DOM object

The transpartent method.

DomObject = jQueryObject[index] or = jQueryObject.get(index).

2.DOM Object To jQuery Object

Var variable = getElementsByTagName(“tr”); //DOM object

Var $variable = $(variable) // jQuery object;

The transpartent method.

jQueryObject = $(DomObject)

四、jQuery开发工具

1.Dreamweaver 可以安装插件提示jQuery 插件名称：jQuery\_API.mxp

2.Aptana 功能非常强大的、开源、专注javascript和Ajax开发IDE;

3.Visual Studio 2008。想要安装插件之后才能提示jQuery

第二章

一、选择器

jQuery的选择器和CSS的选择器非常相似，只不过两者的作用效果不同，CSS是对元素增加样式，jQuery是对元素增加行为。

jQuery写法

$(“.classname”){

//do something

}

CSS写法

.classname{

//do something

}

jQuery获取的永远是对象，如果要判断页面是否有没个元素不能直接写

if($(“.classname”)){

//do something

}

而要根据获取到元素的长度来判断，写成

If($(“.classname”).length>0){

// do something

}

或者转化为DOM对象来判断，写成

If($(“.classname”).get(0)){

//do something

}

jQuery的基本选择器：

ID选择器（$(“#IDname”)），类选择器（$(“#className”)），标签选择器（$(“p”)），通配选择器（$(“\*”)）。

层次选择器：

后代选择器（$(“ancestor descendent”)），子选择器（$(“parent > child”)），相邻选择器（$(“prev+next”)），兄弟选择器（$(“prev~sublings”)）

注意，相邻选择器和兄弟选择器用的比较少，因为有更好的替代方法，前者有.netx()来代替，后者用.sublings()来代替，这个方法可以选择所有的同辈元素，不受先后顺序影响。

过滤选择器主要有 $(“demo:first”),$(“demo:last”),$(“demo:not(\*\*)”),$(“demo:even”)

,$(“demo:odd”),$(“demo:eq(index)”),$(“demo:gt(index)”),$(“demo:lt(index)”),$(“demo:header”),$(demo:animated)

内容过滤选择器$(“demo:contains(text)”),$(“demo:empty”),$(“demo:has(selector)”),

$(“demo:parent”)

可见性过滤选择器$(“demo:hidden”),$(“demo:visible”)

属性过滤选择器$(“demo[attribute]”),$(“demo[attribute=value]”),

$(“demo[attribute!=value]”),$(“demo[attribute^=value]”),$(“demo[attribute$=value]”),

$(“demo[attribute\*=value]”),$(“demo[attribute][attribute]…”)

子元素过滤选择器$(“demo:nth-child(index/even/odd)”),$(“demo:first-child”),

$(“demo:last-child”),$(“demo:only-child”)

表单对象属性过滤选择器$(“demo:enable”),$(“demo:disabled”),$(“demo:checked”),

$(“demo:selected”)

表单选择器$(“demo:input”),$(“demo:text”),$(“demo:password”),$(“demo:radio”),

$(“demo:checkbox”),$(“demo:image”),$(“demo:submit”)…

第三章

所谓DOM，就是将HTML中的每一个节点看成一个DOM，多个DOM组成DOM树。

jQuery中的DOM操作主要包括：

1. 创建元素节点：
   1. $(“html”) 比如：var $li = $(“<li>first DOM</li>”);
   2. 创建文本节点 var $li = $(“<li>firstDOM</li>”); $(“ul”).append($li);
2. 插入节点使用下面方法：
   1. append() 向每个匹配的元素增加内容；
   2. appendTo(),比如A appendTo (B)，把A加到B，同上面相反；
   3. prepend() 向每个匹配的元素前置内容；
   4. prependTo() 同appendTo()用法；
   5. after() 在每个匹配的元素之后增加内容；
   6. insertAfter() 同appendTo()；
   7. before() 在每个匹配的元素之前增加内容；
   8. insertBefore() 同appendTo()；
3. 删除节点
   1. remove() 删除选中的元素及其子元素；
   2. empty() 清空节点内容；
4. 复制节点
   1. clone() 可添加参数true设置是否觉有原元素的属性；
5. 替换节点
   1. replaceWith() 将匹配的元素设置为给定的HTML；
   2. replaceAll() 同appendTo()；
6. 包裹节点
   1. wrap() 将匹配的元素用给定的HTML包裹起来；
   2. wrapAll() 将匹配的元素用一个元素包裹；
   3. wrapInner() 将匹配的元素的子内容用给定的HTML包裹起来；
7. 属性操作
   1. attr() 将匹配的元素设置属性；
   2. removeAttr() 将匹配的元素删除属性；
8. 样式操作
   1. attr() 可以为匹配的元素设置class属性，但是这是替换，不是追加；
   2. addClass() 为匹配的元素增加样式；
   3. removeClass() 将匹配元素的给定样式移除；
   4. toggle() 切换样式；
   5. hasClass() 判断是否具备给定的样式，返回ture/false
9. 设置获取HMTL，文本值
   1. html() 读取或设置匹配元素的HTML内容；
   2. text() 读取或者设置匹配元素的HTML文本；
   3. val() 读取或者设置元素的值，主要指下拉菜单，文本框，单选框等；
10. 遍历节点
    1. children() 获得匹配元素子元素的集合；
    2. next() 获得匹配元素相邻的下一个同辈元素；
    3. prev() 获得匹配元素相邻的上一个同辈元素；
    4. siblings() 获得匹配元素所有的同辈元素；
    5. closet() 获得最近的匹配元素，子元素，父元素，这样的顺序寻找；
11. CSS-DOM操作
    1. css() 为匹配元素设置CSS样式；
    2. height() 获得匹配元素的高；
    3. offset() 获得元素在当前视窗的偏移，返回两个属性：top，left；
    4. position() 获得元素相对于最近的一个position设置为relative或者absolute的元素的偏移量，返回两个属性：top，left
    5. scrollTop(),scrollLeft()方法 获取滚动条距顶端和左侧的距离

第四章

1. jQuery中的事件
   1. window.load与$(function(){ // do something})的区别，前者一个页面存在多个，只能执行最后一个，后者全部执行；前者在所有元素加载完成后执行（包括图片），后者在DOM模型建立后执行；
   2. 事件绑定：bind(type,[,data],fn)第一个参数是事件类型，包括blur,focus,load,resize,

scroll,unload,click,dblckick,mouseover,mousemove,mouseout,mouseup,mousedown,mouseenter,mouseleave,change,select,submit,keydown,keypress,keyup,error等；第二个是可选参数，传递给事件的其他参数；第三个用来绑定事件参数；该方法可以简写，即常见的$(“selector”).click/hover/mouseover/…

* 1. 合成事件：hover(enter,leave)，当鼠标指上去的时候，会触发第一个函数，当鼠标移除的时候，会触发第二个函数；toggle(“fn1,fn2,fnN…”)，用于模拟鼠标连续单击事件，第一次点击，触发第一个函数，第二次点击触发第二个函数..
  2. 事件冒泡：加入对p标签绑定了单击事件，又对他外围的DIV绑定了单击事件，这样在单击P标签的时候，会将两个事件都触发。这里包括以下知识点：事件对象：在函数中增加一个参数event比如$(“#demo”).click(function(event){})。停止冒泡可以使用event.stopPropagation();获取对象的属性，可以用event.type()方法，

event.preventDefault()阻止默认行为，event.target()获取到触发事件元素的属性，event.PageX,event.PageY获取到光标相对于页面的x位置和y位置。event.which()从单击事件中获取到左中右键

* 1. 移除事件：unbind([type](,data))第一个参数是事件类型，不设置则移除所有事件，第二个参数是要移除的函数。one(type,[data],fn);
  2. 模拟操作：trigger(type[,data])，页面载入后，模拟type设定的操作，同时也可以触发自定义的事件,data可以传递参数过去，通常用来区别是认为操作还是模拟操作；

1. jQuery中的动画
   1. show()和hide()方法：两函数作用相反，效果同CSS种display:none或者其他的，可用参数slow,fast,normal，获取数值表示毫秒；
   2. fadeIn()和fadeOut()方法。通过控制不透明度，逐渐消失或者逐渐出现，参数同上；
   3. slideUp()和slideDown()方法。通过控制高度，逐渐伸开或者逐渐收缩，参数同上；
   4. animate()：自定义动画的方法，animate(params,speed,callback)。第一个参数设定动画效果，可以是一个样式或者属性，第二个设置动画速度，第三个设置动画结束后的调用的函数。
   5. 动画的停止：stop(clearQueue,gotoEnd)方法，两参数均为boolean型，第一个参数设置是否清空未完成的动画列队，第二个参数设置是否跳到最后的画面。
   6. 判断元素是否处于动画状态：$(“#demo”).is(:animated)；
   7. 其他动画方法：toggle(speed,callback),slideToggle(speed,callback),  
      fadeTo(speed,callback)

第五章

1. 表单应用
   1. 应用样式一，对于文本框，在获得焦点和失去焦点的时候改变样式，提高用户体验
   2. 多行文本框应用：评论位置添加增加按钮和减少按钮，改变文本框高度；
   3. 滚动条应用：通过改变滚动条的位置，是文本内容滚动，增加动画效果后就非常细腻迷人了
   4. 下拉框应用：会员权限设置，左侧为全部权限，单击某一权限后，增加到右侧已允许的权限列表中
   5. 复选框应用：全选，全部选，反选等
   6. 表单验证：当用户输入完毕后自动验证是否正确，并给予提示；
2. 表格应用
   1. 隔行变色：通过设置不同的CSS样式，在奇偶行添加不用的样式；
   2. 表格内容筛选：通过contains()方法筛选；
3. 其他应用
   1. 网页字体大小
   2. tab选项卡：对当前设置样式同时取消同辈元素的样式，并且根据ID显示匹配的内容
   3. 网页换肤：对引用CSS的语句增加ID属性，通过ID获取后使用attr()对href属性设置不同的值

第六章

jQuery中使用AJAX的方法概要，看完这篇只想说AJAX原来也不难。

1. load(url[,data][,callback])：最常用的方法,远程载入HTML代码并插入到DOM文档中
2. .get(url[,data][,callback][,type])：使用GET方法获得异步请求，url为请求的地址，data发送到服务器的数据，会以get方式传入，callback成功是调用函数，type服务器端返回的格式。
3. .post(url[,data][,callback][,type])：参数和get一样，与.get()不同是，传入的数据量更大，理论上无限制，更安全。
4. .getScript(url[,callback])：动态载入js文件，url为js文件地址，callback回调函数；
5. .ajax(options)：jQuery中AJAX底层函数，可以实现上面所有功能，参数众多；
6. serialize()：将DOM元素序列化为数据，供AJAX传输。

这些就是常用的AJAX操作了，基本想要的功能都可以实现了，具体的方法百度一下吧

**选择器专题**

ready方法保证js在DOM加载完毕后执行，一般写在外联的CSS样式之后，保证通过CSS加载的属性页已经被置于DOM元素之上，一般来说如果只为提高加载速度的话，把js放在最后最好

* + 1. 选择方法有两种，一种是通过jQuery提供的css方式或者特定方式调用，另外一种是通过js获取目标文档的方式调用
    2. 在特定的上下文中查找元素，可以在jQuery()函数中添加两个参数，第一个参数指定选择的元素，第二个参数指定选择的范围；
    3. 总一个集合中新创建一个集合可以使用filter()方法和find()方法，两者不同的是，前者从当前集合中中找到指定的元素，后者从当前元素的子元素中找到指定的元素；
    4. 返回有损操作的原始对象使用end()方法，可以返回filter(),find()等操作的原始对象，一般用在链式操作上面；
    5. 同时操作原始对象和新创建的对象可以使用andSelf()方法，这样就可以同时操作两个对象了，比如jQuery(“div”).find(“p”).andSelf().css(“background”,”#f90”)这样就可以将两个对象都可以加上橘黄色背景了；
    6. 根据当前元素选择上下文元素，当前某个元素可以使用eq()函数，然后使用prev(),next(),parent(),parent().children(),nextAll(),分别选择上一个，下一个，父辈元素，全部同辈元素，一下的全部元素
    7. 设置获取元素属性使用attr()方法
    8. 设置获取文本text()，设置获取网页元素html()；区别在于前者会转义HTML后者不会转义HTML