**jQuery学习笔记**

**第一部分**

1. 简介

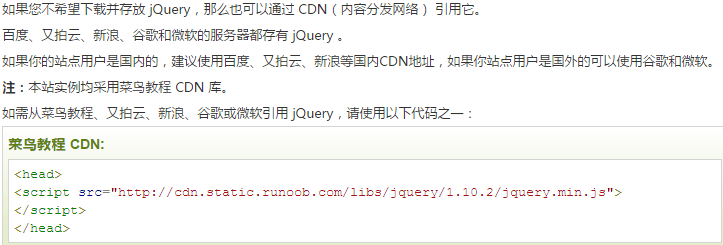
jQuery 库可以通过一行简单的标记被添加到网页中。

1. 安装

使用方法有两种，一是使用在线CDN（内容分发网络）方式，另一种则是使用本地jQuery库（下载刚到本地）。



使用各种CDN服务，可以免去本地存储的麻烦。具体使用可以去插件库（站点）查看链接；



注意：使用百度、又拍云、新浪、微软等公司的jQuery优势在于，以往用户访问其他网站时已经将jQuery加载好存在本地，这样你的站点使用的时候是直接在本地加载的，减少了加载时间。同时，大多数CDN都可以保证用户在想起发出请求时，会从里用户最近的服务器加载，这样也可以加载的更快。

1. 语法

jQuery是通过选取HTML元素，并对选取的元素执行某些操作。

基础语法：

$(selector).action()

各部分作用：

* 美元符号定义使用jQuery；（也有其他的JavaScript框架使用这个符号，件noConflict（）方法详解）；还有需要注意的是，**jQuery=$**
* 选择符（selector）“查询”和“查找”HTML元素；
* jQuery的action（）执行对元素的操作；

例如：$（this）.hide（）-----隐藏当前元素

**文档就绪事件**

**为了防止在文档加载之前就开始运行jQuery代码，影响页面加载，于是就有了文档就绪事件，即把所有的jQuery代码都放在文档就绪事件中来完成；**

**$(document).ready(function(){**

**//开始写jQuery 代码...**

**})**

**简单写法：**

**$(function(){**

**//开始写jQuery 代码...**

**})；**

1. 选择器

jQuery的选择元素的方法和CSS的风格很相似。在选择元素的这一方面比较重要的两个操作是：

**选择元素——$()**

**设置样式——CSS（）**

（1）、jQuery选择器

允许对HTML的元素组或者单个元素进行操作；选择器是基于元素的ID、类、类型、属性、属性值等查找或选择HTML元素的，基于已经存在的CSS选择器，也有一些自定义的选择器。

1. 、元素选择器

基于元素名选取元素；

例如，选择所有<p>元素-------$(‘p’);

（3）、#id选择器

通过HTML元素的id来选取指定的元素。

语法：$(‘#test’);

（4）、class选择器

通过指定的class查找元素；

语法：$(“.test”);

更多：http://www.runoob.com/jquery/jquery-ref-selectors.html

1. jQuery事件

事件：页面对不同访问者的响应叫做事件

事件处理程序：当HTML发生某些事件时所调用的方法；

常见的DOM事件：



1. 常用的jQuery事件方法

（1）ready（）文档就绪事件----------允许我们在文档加载完成后执行函数。

$(document).ready({

//将要执行的代码；

})

（2）click（）------当按钮被点击事件被触发时会调用一个函数，该函数在点击HTML元素时执行；

$(“p”).click(function(){

$(this).hide();

});

（3）dblclick()------双击元素时触发的事件

$(“p”).dblclick(function(){

$(this).hide();

})

（4）mouseenter（）--------鼠标滑过元素时触发；

$(“p”).mouseenter(function(){

alert(“您的鼠标移到了X元素上”)

})

1. mouseleave（）------鼠标指针离开元素时触发；

$(“p”).mouseleave(function(){

alert(“再见，你已经离开了该段落！”);

})

（6）、mousedown（）---------鼠标指针移到元素上方并按下鼠标键时触发；

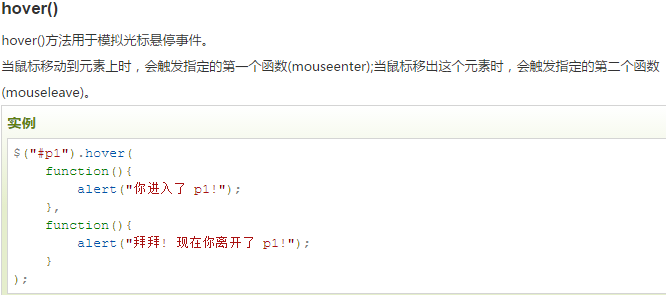
$(“p”).mousedown(function(){

alert（“鼠标在该段落上按下了！”）；

})

（7）、mouseup（）-----原理同上

（8）hover()-------用于模拟光标悬停事件



其中对的两个匿名函数会先后执行,意思就是说hover（）参数可以传入多个，这些参数都是“类鼠标悬停”的时候将要执行的代码。

（9）、focus()-------------当元素获得焦点时触发。

当通过鼠标点击或者tab键定位到元素时，该元素就会获得焦点。

$(“input”).focus(function(){

$(this).css(“background-color”,”#ccc”);

})

（10）、blur-----当元素失去焦点时触发的事件

$(“input”).blur(function(){

$(this).css(“background-color”,”#fff”);

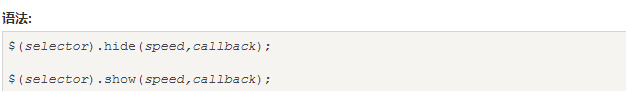
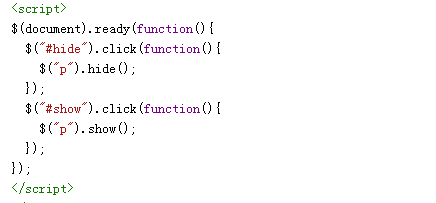
})

**第二部分------jQuery效果**

包括隐藏、显示、切换、滑动、淡入淡出以及动画

**基本语法复习：$(被操作对象).事件（要执行的代码）**

1. 隐藏/显示------hide（）/show（）



Speed参数为可选，规定隐藏和显示的速度，可取值为：slow、fast、或毫秒,其中的默认效果是fast。

callback参数也是可选，这是显示或者隐藏之后将会执行的函数名称。

**toggle（）**

通过jQuery，可以使用）方法来切换hide（）和show（）方法，显示被隐藏的元素或隐藏toggle（显示的元素。如下：

**$(“button”).click(function(){**

**$(“p”).toggle();**

**});**

**语法：$(selector).toggle(speed,callback);**

其中的speed和callback都是可选参数，意义和hide（）、show（）方法的一样。

对于可选的 callback 参数，有以下几点说明：

（1）$(selector)选中的元素的个数为n个，则callback函数会执行n次；

（2）callback函数名后加括号，会立刻执行函数体，而不是等到显示/隐藏完成后才执行；

（3）callback既可以是函数名，也可以是匿名函数；

1. 淡入淡出
2. fade方法---------实现淡入淡出的效果；

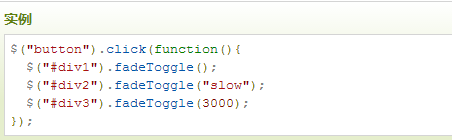
* fadeIn（）------用于淡入已经隐藏的元素；

$(selector).fadeIn(speed,callback);

* fadeout（）-----用于淡出已经显示的元素；

$(selector).fadeOut(speed,callback);

* fadeToggle（）--------用于在fadeIn（）和fadeOut之间切换；



* fadeTo（speed，opacity，callback）------允许渐变为给定的不透明度（opacity值介于0~1之间）实际上就是一个渐变的过程，在css中也可以实现同样的效果；

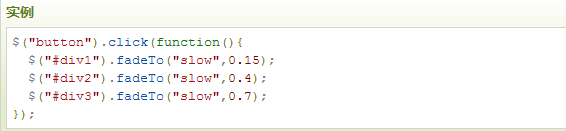
语法：

$(selector).fadeTo(speed,opacity,callback);

Speed:在此处是必须的参，用于规定效果的时长，取值依然是slow、fast或毫秒；

Opacity：值在0到1之间，参数是必须存在的，

Callback：可选的参数，是该函数完成后将会执行的参数；



3、滑动slide方法

滑动拥有如下方法：

slideDown（）

slideUp（）

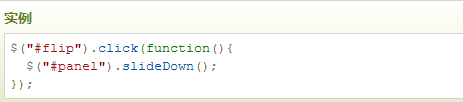
slideToggle（）

这个滑动的效果就相当于一块可以上下收起来的面板。

（1）、slideDown（）方法----------用于向下滑动元素；

语法：$(selector).slideDown(speed,callback);

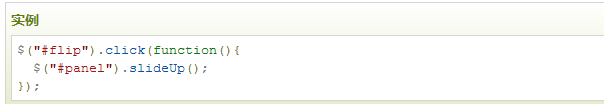
参数都是可选的，取值和之前的一样；

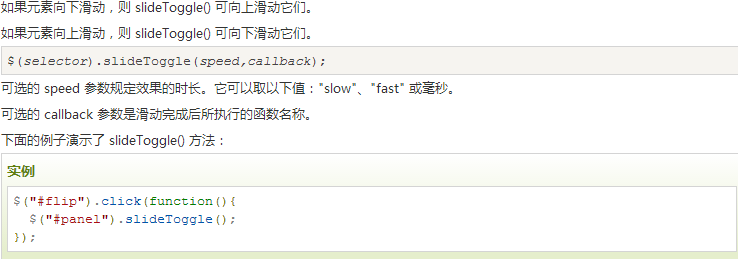


（2）、slideUp（）方法---------用于向上滑动元素；

语法：$(selector).slideUp(speed,callback);

参数都是可选的，取值和之前的一样；

（3）、slideToggle（）方法---------在slideDown（）和slideUp（）之间切换；



4、动画animate（）

animate（）创建自定义动画；

语法：

$(selector).animate（{params}，speed，callback）；

必需的params参数定义动画的CSS属性；后两个参数和前面的保持一致。

例如：$(“button”).click(function(){

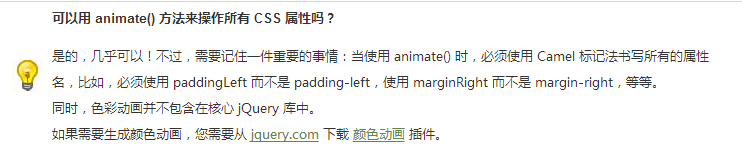
$(“button”).animate({left:250px;},speed,callback);

})

注意：默认情况下HTML元素是静止的，且无法移动；如需对位置进行操作则需要把CSS的position属性值设为absolute、relative或者fixed；

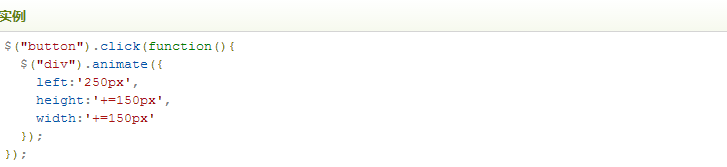
animate（）-------操作多个属性

在生成动画的过程中可以同时使用多个属性，就是在params那个板块设置更多的CSS属性，但是那些符合的属性名要使用驼峰命名法书写；



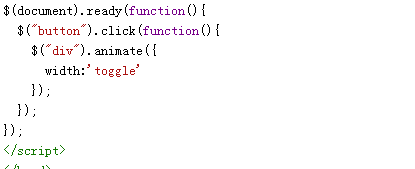
animate（）-------使用相对值

需要在值的前面加上+=或者-=；意思就是在当前值的基础上加上或者减去某一个数值。



animate（）使用预定义值

可以把属性的动画值设置为show、hide或者toggle；



这段代码的效果就是：点击按钮，div的宽会在hide、show之间切换；

animate（）----------使用队列功能

jQuery提供队列功能，这就是说在编写多个animate（）调用之后，jQuery会创建包含这些方法调用的“内部”队列，然后逐一运行这些animate（）调用；



1. 停止动画

Stop（）方法----用于在效果或者动画效果完成之前对它们进行停止。

Stop（）方法适用于所有的的jQuery效果函数，包括滑动、淡入淡出和自定义动画。

语法：$(selector).stop(stopAll,goToEnd);

stopAll:可选参数，规定是否应该清除动画队列；默认值是false，即仅停止活动的动画，允许任何排入队列的动画向后执行。

可选的goToEnd规定是否立即完成当前动画，默认是FALSE；因此，stop（）会清除在被选元素上指定的当前动画。

stop(true)等价于stop(true,false): 停止被选元素的所有加入队列的动画效果。

stop(true,true):停止被选元素的所有加入队列的动画，但允许完成当前动画。

stop()等价于stop(false,false):停止被选元素当前的动画，但允许完成以后队列的所有动画。

stop(false,true):立即结束当前的动画到最终效果，然后完成以后队列的所有动画。

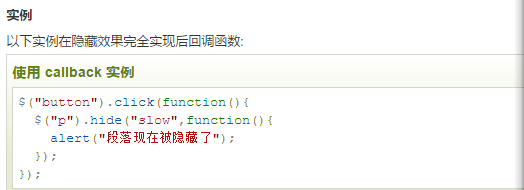


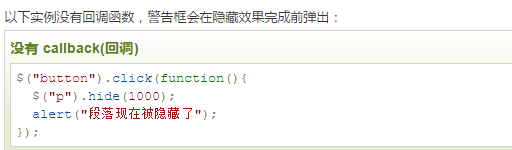
1. callback（）事件

Callback函数在当前动画100%完成后执行；

jQuery动画的问题：

许多jQuery函数涉及动画，这些函数或许会将speed和duration作为函数的可选参数，*speed* 或 *duration* 参数可以设置许多不同的值，比如 "slow", "fast", "normal" 或毫秒。

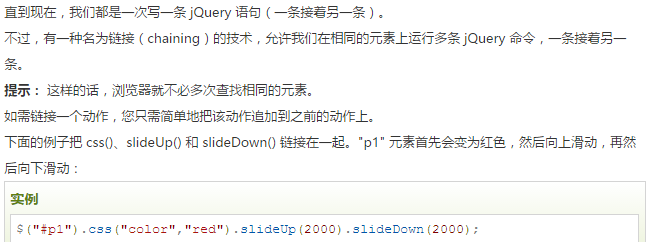


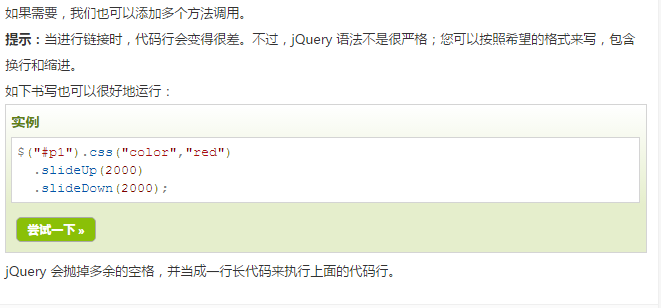


1. 链（chaining）

通过jQuery可以把动作、方法链接在一起。Chaining允许我们在一条语句中运行多个jQuery方法（在相同的元素上）；

jQuery方法链接：





**第三部分 jQuery HTML**

1、获取内容和属性

jQuery中很强大的一个功能就是操作DOM的能力，这使得访问和操作元素变得很容易。

DOM定义访问HTML和XML文档的标准：w3c文档对象模型独立于语言界面和平台，允许程序和脚本动态访问和更新文档的内容、结构以及样式。

1. 、获得内容-------text（）、HTML（）以及val（）、attr（）

三个简单实用的用于DOM操作的jQuery方法：

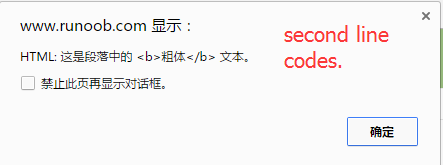
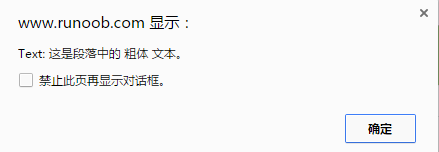
* text（）-------设置或者返回所选元素的文本内容；
* HTML（）------设置或返回所选元素的内容（包含HTML标记）；

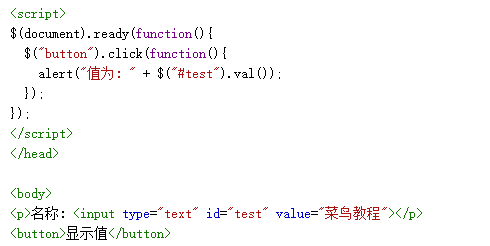
上面两个方法的区别很明显，前者是值获取文本信息，但是后者是获取被选中元素内部包含html标签在内的所有内容。在含有参数（设置内容）的时候，同理，前者是仅仅设置文本内容，但是后者是设置包含标签在内的内容。

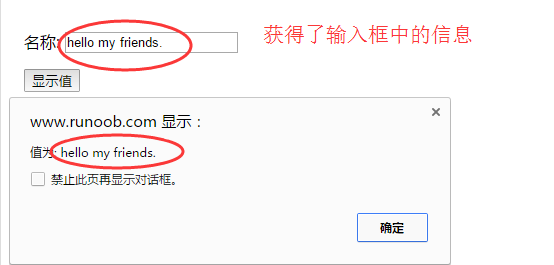
* val（）--------设置或或返回表单字段的值；
* attr（）——用于获取所选元素的属性。

对于元素的属性操作，在不含有参数的时候都是获取属性，但是如果含有参数了就是设置属性值，并且可以同时对多个属性进行操作，每一组用冒号联系起来，然后组与组之间用逗号隔开。

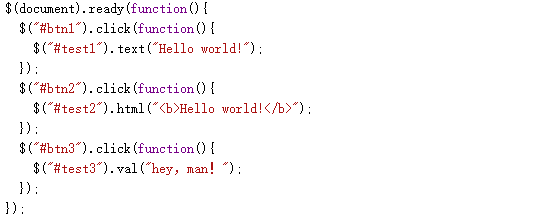






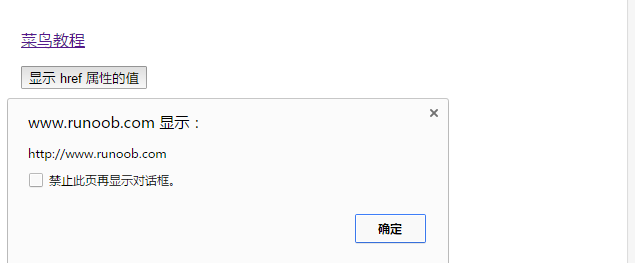
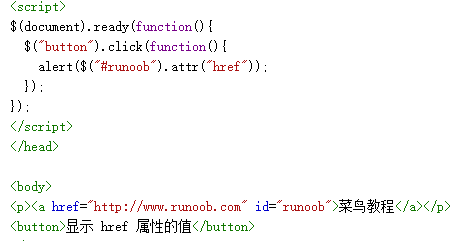


在以上三个方法（text（）、HTML（）以及val（））使用时，在调用它们之后，在括号内填上需要设置的内容就完成了对内容的设置。例如：



1. 获取属性------attr（）

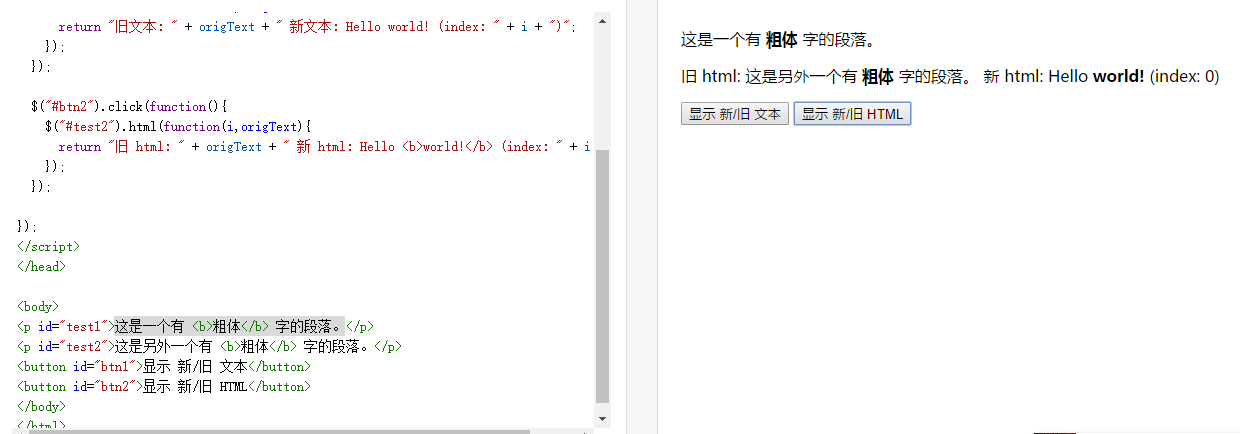
这个方法适用于获取元素的属性：



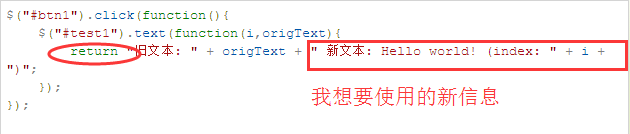
在使用attr（）来进行元素属性的设置的时候，使用方法是.attr（“属性名”，“要设置的属性值”）

1. text（）、HTML（）以及val（）的回调函数

回调函数拥有两个参数：被选元素列表中当前元素的下标，以及原始（旧的）值。然后以函数新值返回你希望使用的字符串。



意思就是在这几个方法的参数位置放上另一个函数（可以是匿名函数），其中的参数是你所选中的那个元素在他所在元素列表中的位置的下标；另一个则是在文档之中原本存在的内容，在函数返回值中返回你希望使用的那些信息（字符串）。



4、设置属性-----attr（）

这个方法用于设置或者改变属性值。

（1）$(“#XXX”).attr( “属性名” ,**：**”要设置的属性值” ） （注意中间那是冒号）

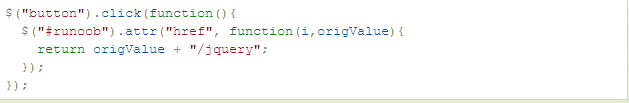


（2）jQuery也允许设置多个属性：

$(“#XXX”).attr( “属性名” **：**”要设置的属性值” ，“属性名” **：**”要设置的属性值” ）；

1. attr（）的回调函数

Attr（）方法也提供回调函数，同样也是有两个参数：被选元素列表中当前元素的下标，以及原始（旧）的值，然后以函数新值返回你希望使用的字符串。



1. 添加元素

添加元素的方法有如下四种：

* 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；

这两个方法都是讲追加的内容加入到所选元素的内部，也就是说追加的内容是所选元素是的子级。

* after（）-----------在被选元素之后插入内容；
* before（）---------在被选元素之前插入内容；

这两个方法追加的元素和被选中元素的关系是同级（兄弟）关系。

1. 、append（）方法---------------对应于JS中的appendchild（）

$(“选中元素”）append(“追加的文本”）；

1. 、prepend()方法-------在被选元素开头添加文本；

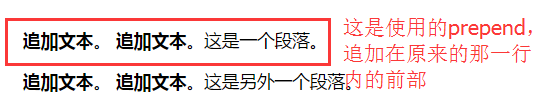
$(‘’P”).prepend(“在开头追加文本”)

通过append（）和prepend（）方法添加若干新元素

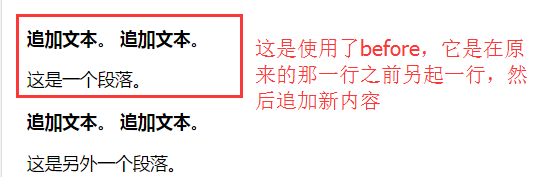
在上面的例子中，我们只在被选元素的开头/结尾插入文本或者HTML，但是append（）和prepend（）方法能通过参数接受无限数量的新元素。可以通过jQuery或者JavaScript和DOM元素来生成文本/HTML。



注意区分差异：



Append（）方法的作用方法和prepend一样，但是这二者追加的内容是出现在被选中元素内部。



before（）和after（）的作用方法一样，都是把新追加的元素当做一个全新的内容来追加的；这二者追加的是和选中元素并列的（成为兄弟元素）新元素。

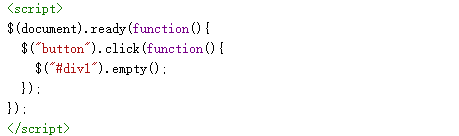
1. 删除元素/内容

删除元素/内容的方法有两个：

remove（）：删除选中的元素（包括其子元素）；同时也可以接受一个参数，指定删除的内容/元素（一个选择器来选出来）；



empty（）：删除选中元素的子元素；



综合二者之间的不同，总结起来就是：remove（）是将选中元素全部移除，包含他的子元素，可以使用选择器过滤，指定要删除的内容；而empty（）方法则是清空选中元素，也就是说只是把选中元素的所有子元素移除，选中元素依然存在于文档流之中。

1. 获取并设置CSS类

addClass()--------添加一个或多个新的类；

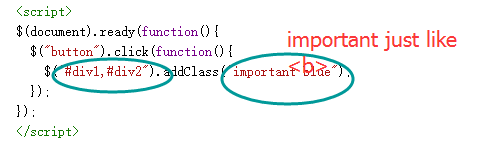
removeClass()---------删除一个或是多个类；

toggleClass（）--------在对类的添加和删除操作之间切换；

CSS（）-----------设置或返回样式属性；

1. 、addClass（）

可以接受多个参数，中间以空格隔开；但是注意，在选择器取得操作对象时也可以同时取多个，中间用逗号隔开。



（2）removeClass()、toggleClass()也可以同时对多个对象同时进行操作，选取方法同上，但是不能同时操作多个属性（还有待求证）

1. CSS方法----------用于返回或设置被选元素的一个或多个属性；

1、返回CSS属性

语法：

$(“选择器”).css(“样式名称”)；

（2）、设置属性

语法：

$(“选择器”).css（“属性名”,”属性值”）；

(3)、设置多个属性值

语法：

$(“选择器”）.css(”属性名1”：”属性值1”，”属性名2”：”属性值2”，....);

注意：在每一对属性对之间是由冒号隔开的，每一对之间是由逗号隔开的。

这个设置属性的操作和前面获取和设置attr（）相似，在内部的代码的书写形式也是一样的。

1. jQuery尺寸



（1）、jQuery width() 和 height() 方法

width() 方法设置或返回元素的宽度（不包括内边距、边框或外边距）。

height() 方法设置或返回元素的高度（不包括内边距、边框或外边距）。

1. 、jQuery innerWidth() 和 innerHeight() 方法

innerWidth() 方法返回元素的宽度（包括内边距）。

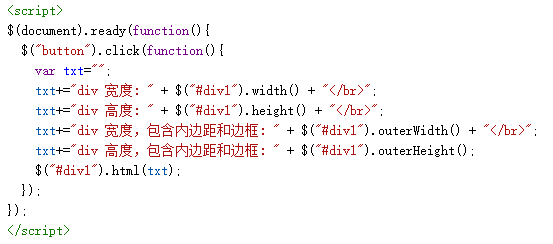
innerHeight() 方法返回元素的高度（包括内边距）。

（3）、jQuery outerWidth() 和 outerHeight() 方法

outerWidth() 方法返回元素的宽度（包括内边距和边框）。

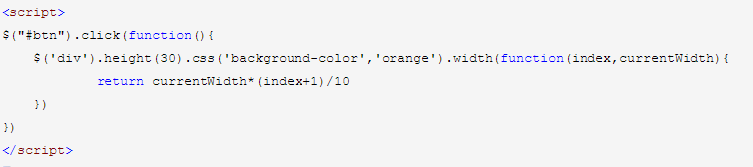
outerHeight() 方法返回元素的高度（包括内边距和边框）。

注意：当参数为true的时候，那么结果也将包含margin值。



1. css（‘width’）和width（）的区别就是前者返回一个含有单位的字符串，但是后者返回的是一个没有单位的数值。同理，height也是一样的用法，同时也能够用来获取document和window的尺寸。
2. 当width(value)\height(value)有参数的时候就是设置选中元素的值，单位默认是px，也可以自己设置。
3. height(function(index,currentHeight))/width(function(index,currentWidth))

height()/width()方法也可以以一个函数作为参数，该函数接受两个参数，index参数表示元素在集合中的位置，currentHeight/currentWidth参数表示原来的宽高。在这个函数中，this指向元素集合中的当前元素，最终返回设置的宽高。



1. 位置设置

（1）offsetParent()

jQuery通过offsetParent()找到元素的定位父级，但是jQuery和JavaScript规则有些不同：

a、当元素本身不是fixed定位，且父级元素存在经过定位的元素，offsetParent()的结果为离自身元素最近的经过定位的父级元素；

b、当元素本身具有fixed定位，或父级元素都未经过定位，则offsetParent()的结果为html；

c、body元素的offsetParent()的结果也是html；

（2）position()

　　position()方法不接受参数，用来获取匹配元素中第一个元素的相对于定位父级的坐标；

position()返回一个包含top和left属性的对象，相当于javascript中的offsetTop和offsetLeft；

（3）offset()

当offset()方法没有参数时，在匹配的元素集合中，获取的第一个元素的当前坐标，坐标相对于文档；

offset也是可以接受一个参数的，这时候返回一个带有left和top的对象，可以直接将left、top的值进行设置（left：20），这样就能完成设置。

**offset(function(index,coords))**

offset()方法可以接受一个函数作为参数。在函数中，元素在匹配的元素集合中的索引位置作为第一个参数，当前坐标作为第二个参数。这个函数返回一个包含top和left属性的对象；



（4）scrollTop()/scrollLeft()

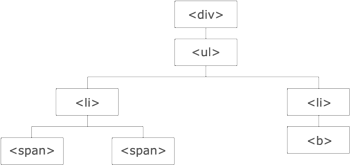
　　scrollTop()/scrollLeft()方法不带参数时，用来获取匹配元素集合中第一个元素的当前水平或垂直滚动条位置；要是添加一个参数的话就可以设置相应的参数值（一个正整数）。

**第四部分 遍历**

1. jQuery遍历

jQuery遍历，意为“移动”，用于根据其相对于其他元素的位置来进行查找（或选取）HTML元素。从某一项选择开始，并沿着这个选择开始移动，直到抵达我们期望的那一个元素。

下图展示了一个家族树。通过 jQuery 遍历，您能够从被选（当前的）元素开始，轻松地在家族树中向上移动（祖先），向下移动（子孙），水平移动（同胞）。这种移动被称为对 DOM 进行遍历。



图示解析：

<div> 元素是 <ul> 的父元素，同时是其中所有内容的祖先。

<ul> 元素是 <li> 元素的父元素，同时是 <div> 的子元素

左边的 <li> 元素是 <span> 的父元素，<ul> 的子元素，同时是 <div> 的后代。

<span> 元素是 <li> 的子元素，同时是 <ul> 和 <div> 的后代。

两个 <li> 元素是同胞（拥有相同的父元素）。

右边的 <li> 元素是 <b> 的父元素，<ul> 的子元素，同时是 <div> 的后代。

<b> 元素是右边的 <li> 的子元素，同时是 <ul> 和 <div> 的后代。

2、jQuery遍历——遍历DOM

jQuery提供了多种遍历DOM的放法，其中最大的遍历种类是树遍历（tree-traversal）。

1. jQuery遍历——祖先

祖先是父、祖父、曾祖父等等。通过jQuery遍历我们能够进行向上的遍历，以查找元素的祖先。

1. 、向上遍历DOM树的方法

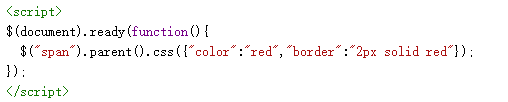
parent（）

parents（）

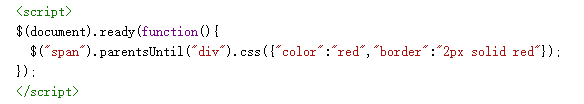
parentsUntil（）

1. parent（）方法----返回被选元素的直接父元素。

该方法只会向上一级对DOM树进行遍历。



1. parents（）方法----返回被选元素的所有祖先元素，它一路向上直到文档的根元素（<html>为止）。
2. ParentsUntil（）方法----返回介于两个给定元素之间的所有祖先元素。（注意是介于二者之间的，也就是说是不包含被选元素和另一个截止元素）



Parentsuntil（）方法的意思相当于取出在div之前的所有span元素的祖先元素，不包含div。

1. jQuery遍历——后代

后代是子、孙、曾孙等。通过jQuery我们可以向下遍历DOM树，以查找元素的后代。

1. 向下遍历DOM树的jQuery方法：

children（）

find（）

1. children（）方法----返回被选元素的直接子元素；

该方法只会对下一级DOM树进行遍历。其中children（）接受一个参数，括号中可以传入一个选择器来对需要操作的元素进行筛选，比较有针对性。



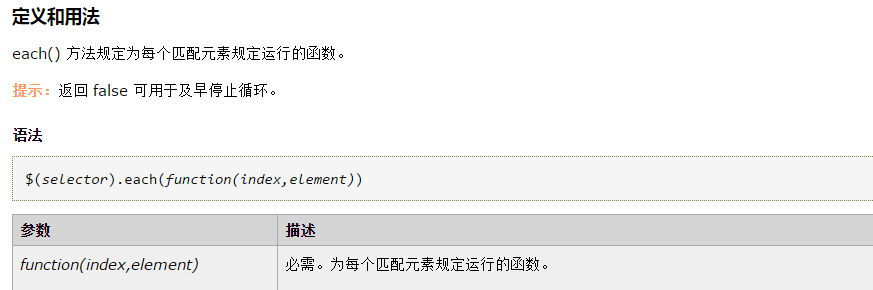
1. find（）方法————返回被选元素的所有后代元素，一直到最后一个后代元素；



其中find（）必须传入一个参数来指定你要查找的元素。参数类型有：元素、类名、ID名或者通配符“\*”（获取所有子元素）；

1. 、each（）方法————能使DOM循环结构简洁，不容易出错。

each()函数封装了十分强大的遍历功能，使用也很方便，它可以遍历一维数组、多维数组、DOM, JSON 等等



1. jQuery遍历——同胞（sibling）

同胞就是拥有相同父元素的元素。

在DOM树中水平遍历的方法：

siblings（）

next（）

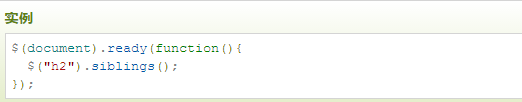
nextAll（）

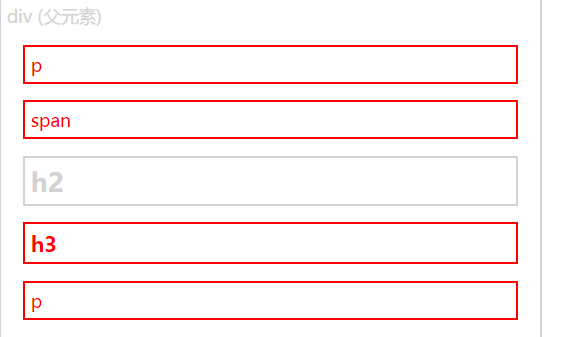
Prev()

prevAll()

prevUntil()

1. 、siblings（）方法————返回被选元素的所有同胞元素（但不包括自己）；

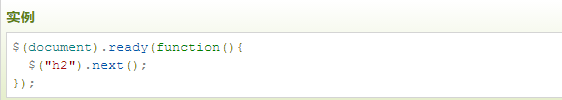




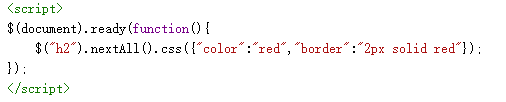
同样可以使用可选参数来筛选对同胞元素的搜索，就像平常使用选择器一样。

1. next（）方法————返回被选中元素的下一个同胞元素

该方法值返回一个元素，即被选中元素的下一个元素，但是并不包括被选中元素的子元素。



1. nextAll（）方法————返回被选中元素之后的所有同胞元素（不包含本身及各个元素的子元素）；



1. nextUntil（）方法————返回介于两个给定参数之间的所有同胞元素（不包含作为边界的那两个元素）；
2. Prev（）、prveAll（）、prevUntil（）方法

这三种方法的工作原理和之前的哪些类似，只是在方向上不同，刚好相反。

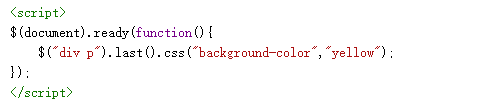
1. jQuery遍历——过滤

**缩小搜索元素的范围**

http://blog.csdn.net/csujiangyu/article/details/47776567过滤选择器可以参考此文章

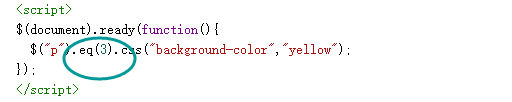
三个最基本的过滤方法是：first（）、last（）和eq（）。这三个方法允许基于其在一组元素中的位置来选择一个特定的元素。其他过滤方法，如filter（）、not（）允许使用匹配或不匹配某项指定标准的元素。

1. first（）方法————返回被选元素的首个子元素；
2. last（）方法————返回被选元素的最后一个子元素；



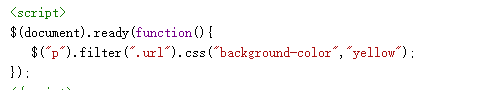
1. eq()方法————返回被选元素中带有指定索引号的元素

索引号是从0开始的，因此首个元素的索引号是0而不是1；



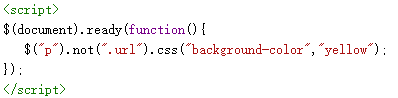
1. filter（）方法————选取返回和标准匹配的元素

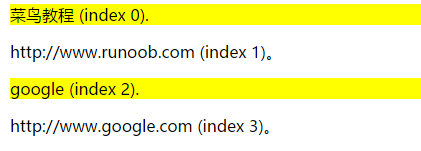
filter（）方法允许规定一个标准，不匹配这个标准的元素会被从集合中删除，匹配的元素的元素则会被返回。



这是一个返回带有url类名的实例。

（5）not方法————返回不匹配标准的所有元素（和filter（）方法刚好相反）；





返回的是不包含.url类的元素；

第五部分 AJAX

1. AJAX简介

AJAX是与服务器交换数据的技术，在不重新加载全部页面的情况下，实现了对部分网页的更新。

1. 什么是AJXA?

AJAX = 异步JavaScript和XML（Asynchronous JavaScript and XML）。

简单的说，在不重新加载整个网页的情况下，AJAX通过后台夹杂数据，并在网页上显示。使用案例如：谷歌地图、腾讯微博、优酷视频、人人网等。

通过jQuery AJAX方法，我们能够使用HTTP Get和HTTP Post从远程服务器上请求文本、HTML、XML或json，同时还能够把这些外部数据直接载入网页被选中的元素中。

1. jQuery load（）方法————从服务器加载数据，并把服务器返回数据返回选中的元素；

语法：$(“selecteor”).load（URL，data，callback）；

URL：必需，规定想要加载的参数。

Data：规定与请求一同发送的查询字符串键/值对集合；

Callback：规定load（）函数指向完毕后将要执行的函数名称；

实例：加载外部TXT文件到指定元素中



也可以把jQuery选择器添加到URL中：这是一个把TXT文件中ID=p1的元素的内容加载到指定元素中。

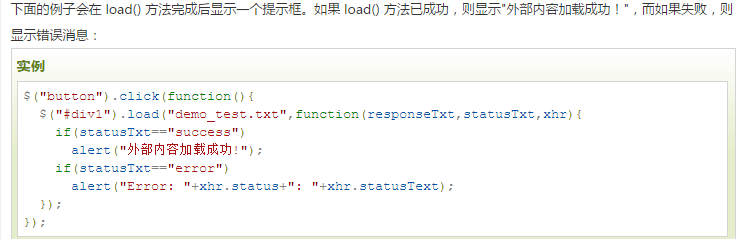


Callback参数规定load（）完成之后所允许调用的回调函数，回调函数可以设置不同的参数：

ResponseTxt——包含调用成功时的结果内容；

StatusTXT——包含地哦啊哟个的状态

Xhr——包含XMLHttpRequest对象；



1. AJAX get（）和post（）方法——从服务器请求数据

两种在客户端和服务器端进行请求和响应的方法分别是get和post。

GET——从指定资源请求数据；

POST——向指定资源提交要处理的数据；

GET基本上用于从服务器取回数据（（也可能返回缓存数据）

POST也可以从服务器获取数据，但是post方法不会缓存数据，并且常用于连同请求一起发送数据

1. 、$.get()

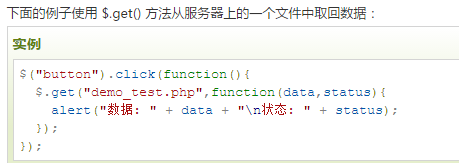
通过HTTP GET 请求从服务器请求数据。

语法：

$.get（URL，callback）；

URL：必需，规定希望请求的URL

Callback：可选，规定请求成功后将要指向的函数；



1. 、$.post()——通过http post从服务器请求数据

语法：

$.post（URL，data，callback）；

必需的 URL 参数规定您希望请求的 URL。

可选的 data 参数规定连同请求发送的数据。

可选的 callback 参数是请求成功后所执行的函数名。

下面的例子使用 $.post() 连同请求一起发送数据：



$.post() 的第一个参数是我们希望请求的 URL ("demo\_test\_post.php")。

然后我们连同请求（name 和 url）一起发送数据。

"demo\_test\_post.php" 中的 PHP 脚本读取这些参数，对它们进行处理，然后返回结果。

第三个参数是回调函数。第一个回调参数存有被请求页面的内容，而第二个参数存有请求的状态。

第六部分 jQuery其他方法

一、jQuery—noConflict()方法

这个方法帮助我们解决怎样在页面上同时使用jQuery和其他框架。

1. jQuery和其他JavaScript框架

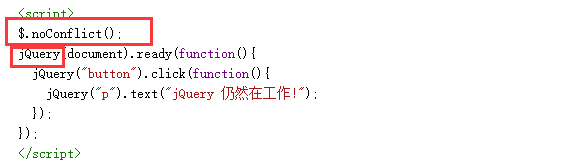
其他一些 JavaScript 框架包括：MooTools、Backbone、Sammy、Cappuccino、Knockout、JavaScript MVC、Google Web Toolkit、Google Closure、Ember、Batman 以及 Ext JS。

其中某些框架也使用 $ 符号作为简写（就像 jQuery），如果您在用的两种不同的框架正在使用相同的简写符号，有可能导致脚本停止运行。

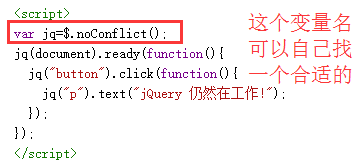
jQuery 的团队考虑到了这个问题，并实现了 noConflict() 方法。

2、noConflict（）方法————三种解决冲突的办法

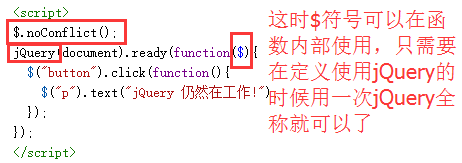
noConflict（）方法会释放对$标识符的控制，这样其他脚本就能够使用$标识符了。同时，仍然可以通过全名代替简写的方式来使用jQuery。（用jQuery代替$标识符）



也可以创建自己的简写。noConflict（）返回对jQuery的引用，我们可以将它存入变量，以供稍后使用。



如果自己的jQuery代码块使用$简写，并且不愿意放弃使用这个快捷方式，那么就可以将$符号作为变量传递给ready函数。这样就可以在函数内部使用$符号了，只是这里会出现一个使用范围的问题，在刚才那个函数外部依旧需要使用jQuery这个全称。



1. JSONP

Jsonp(JSON with Padding) 是 json 的一种"使用模式"，可以让网页从别的域名（网站）那获取资料，即跨域读取数据。

为什么我们从不同的域（网站）访问数据需要一个特殊的技术(JSONP )呢？这是因为同源策略。

同源策略，它是由Netscape提出的一个著名的安全策略，现在所有支持JavaScript 的浏览器都会使用这个策略。