Jquery知识点整理

1、Jquery和原生js的区别

|  | **window.onload** | **$(document).ready()** |
| --- | --- | --- |
| 执行时机 | 必须等待网页中所有的内容加载完毕后（包括图片）才能执行 | 网页中所有DOM结构绘制完毕后就执行，可能DOM元素关联的东西并没有加载完 |
| 编写个数 | 不能同时编写多个，以下代码无法正确执行：window.onload =function({alert("test1");}window.onload = function(){alert("test2");}结果只会输出"test2" | 能同时编写多个 |
| 简化写法 | 无 | $(document).ready(function(){});可以简写成$(function(){}); |

2、一、选择网页元素

　　jQuery的基本设计和主要用法，就是"选择某个网页元素，然后对其进行某种操作"。这是它区别于其他函数库的根本特点。  
　　使用jQuery的第一步，往往就是将一个选择表达式，放进构造函数jQuery()（简写为$），然后得到被选中的元素。

选择表达式可以是CSS选择器：

 $(document)//选择整个文档对象    
 $('#myId')//选择ID为myId的网页元素    
 $('div.myClass')//选择class为myClass的div元素    
 $('input[name=first]')//选择name属性等于first的input元素

也可以是jQuery特有的表达式：

 $('a:first')//选择网页中第一个a元素    
 $('tr:odd')//选择表格的奇数行    
 $('#myForm :input')//选择表单中的input元素    
 $('div:visible') //选择可见的div元素    
 $('div:gt(2)')//选择所有的div元素，除了前三个    
 $('div:animated')//选择当前处于动画状态的div元素    
　　  
 二、改变结果集  
　　如果选中多个元素，jQuery提供过滤器，可以缩小结果集：

\* $('div').has('p'); //选择包含p元素的div元素    
\* $('div').not('.myClass'); //选择class不等于myClass的div元素    
\* $('div').filter('.myClass'); //选择class等于myClass的div元素    
\* $('div').first(); //选择第1个div元素    
\* $('div').eq(5); //选择第6个div元素    
有时候，我们需要从结果集出发，移动到附近的相关元素，jQuery也提供了在DOM树上的移动方法：

 $('div').next('p'); //选择div元素后面的第一个p元素    
 $('div').parent(); //选择div元素的父元素    
 $('div').closest('form'); //选择离div最近的那个form父元素    
 $('div').children(); //选择div的所有子元素    
 $('div').siblings(); //选择div的同级元素

 三、链式操作  
　　选中网页元素以后，就可以对它进行某种操作。  
　　jQuery允许将所有操作连接在一起，以链条的形式写出来，比如：

 $('div').find('h3').eq(2).html('Hello');    
　分解开来，就是下面这样：

1. $('div')                 //找到div元素

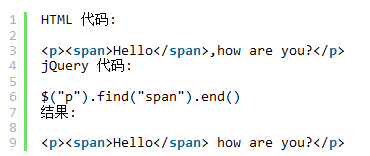
2. .find('h3')           //选择其中的h3元素

3. .eq(2)               //选择第3个h3元素

4. .html('Hello');  //将它的内容改为Hello    
　　这是jQuery最令人称道、最方便的特点。它的原理在于每一步的jQuery操作，返回的都是一个jQuery对象，所以不同操作可以连在一起。  
　jQuery还提供了.end()方法，使得结果集可以后退一步：

1. $('div')    
2. .find('h3')    
3. .eq(2)    
4. .html('Hello')    
5. .end()//退回到选中所有的h3元素的那一步    
6. .eq(0)//选中第一个h3元素    
7. .html('World'); //将它的内容改为World    
.end()：回到最近的一个"破坏性"操作之前。如果之前没有破坏性操作，则返回一个空集。所谓的"破坏性"就是指任何改变所匹配的jQuery元素的操作。

示例  
描述：选取所有的p元素，查找并选取span子元素，然后再回过来选取p元素



-四、元素的操作：取值和赋值

　　操作网页元素，最常见的需求是取得它们的值，或者对它们进行赋值。  
　　jQuery使用同一个函数，来完成取值（getter）和赋值（setter）。到底是取值还是赋值，由函数的参数决定。

 $('h1').html(); //html()没有参数，表示取出h1的值   
 $('h1').html('Hello'); //html()有参数Hello，表示对h1进行赋值    
常见的取值和赋值函数如下：

1. .html() 返回或设置被选元素的内容 (inner HTML)      
2. .text() 取出或设置text内容        
3. .attr() 取出或设置某个属性的值      
4. .width() 取出或设置某个元素的宽度      
5. .height() 取出或设置某个元素的高度    
6. .val() 取出或设置html内容 取出某个表单元素的值    
　　需要注意的是，如果结果集包含多个元素，那么赋值的时候，将对其中所有的元素赋值；取值的时候，则是只取出第一个元素的值（.text()例外，它取出所有元素的text内容）。

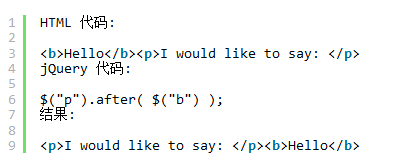
  五、元素的操作：移动  
.insertAfter()，把div元素移动p元素后面：

$('div').insertAfter('p');  
.after()，把p元素加到div元素前面：

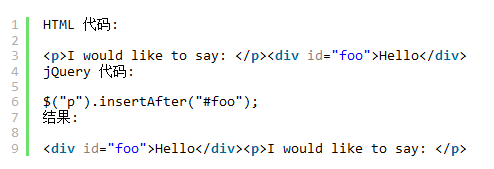
 $('p').after('div');  
使用这种模式的操作方法，一共有四对

1. .insertAfter()和.after()：在现存元素的外部，从后面插入元素    
2. .insertBefore()和.before()：在现存元素的外部，从前面插入元素    
3. .appendTo()和.append()：在现存元素的内部，从后面插入元素    
4. .prependTo()和.prepend()  ：在现存元素的内部，从前面插入元素

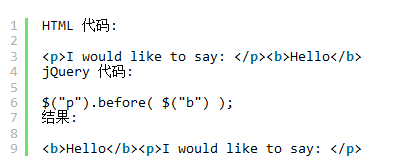
1.after():  
描述:  
在所有段落中后插入一个jQuery对象(类似于一个DOM元素数组)。

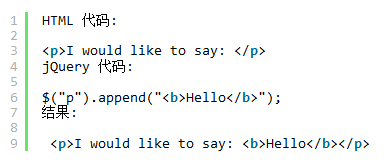


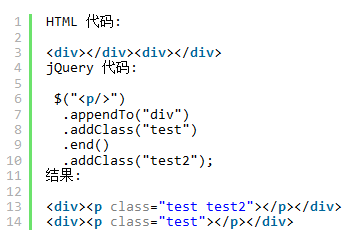
1. insertAfter()：  
   描述:  
   把所有段落插入到一个元素之后。与 $("#foo").after("p")相同

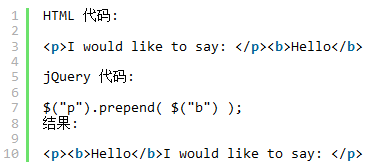


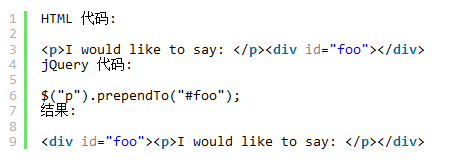
3.before()：  
描述:  
在所有段落中前插入一个jQuery对象(类似于一个DOM元素数组)。

  
4.append()：  
描述：向所有段落中追加一些HTML标记。

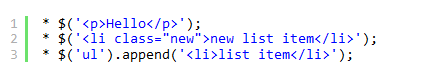
  
5.appendTo()  
描述：新建段落追加div中并加上一个class

  
6.prepend()  
描述：向所有段落中前置一个jQuery对象(类似于一个DOM元素数组)。

  
7.prependTo()  
描述：把所有段落追加到ID值为foo的元素中。

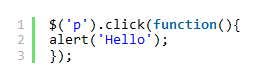


\*\*六\*\*、元素的操作：复制、删除和创建  
　　复制元素使用.clone()  
　　删除元素使用.remove()和.detach()。两者的区别在于，前者不保留被删除元素的事件，后者保留，有利于重新插入文档时使用。  
　　清空元素内容（但是不删除该元素）使用.empty()。  
　　创建新元素的方法非常简单，只要把新元素直接传入jQuery的构造函数就行了：

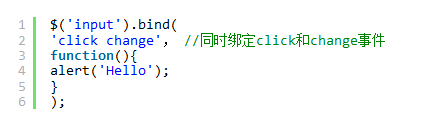
  
七、工具方法  
　　除了对选中的元素进行操作以外，jQuery还提供一些工具方法（utility），不必选中元素，就可以直接使用。  
常用的工具方法有以下几种：

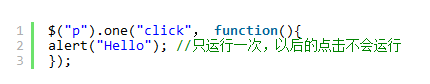
 $.trim() 去除字符串两端的空格。  
 $.each() 遍历一个数组或对象。    
 $.inArray() 返回一个值在数组中的索引位置。如果该值不在数组中，则返回-1。    
 $.grep() 返回数组中符合某种标准的元素。     
 $.extend() 将多个对象，合并到第一个对象。     
 $.makeArray() 将对象转化为数组。    
 $.type() 判断对象的类别（函数对象、日期对象、数组对象、正则对象等等）。    
 $.isArray() 判断某个参数是否为数组。    
 $.isEmptyObject() 判断某个对象是否为空（不含有任何属性）。    
 $.isFunction() 判断某个参数是否为函数。    
 $.isPlainObject() 判断某个参数是否为用"{}"或"new Object"建立的对象。    
 $.support() 判断浏览器是否支持某个特性。

八、事件操作  
　jQuery可以对网页元素绑定事件。根据不同的事件，运行相应的函数。

  
目前，jQuery主要支持以下事件：

 .blur() 表单元素失去焦点。    
 .change() 表单元素的值发生变化    
 .click() 鼠标单击    
 .dblclick() 鼠标双击    
 .focus() 表单元素获得焦点    
 .focusin() 子元素获得焦点    
 .focusout() 子元素失去焦点    
 .hover() 同时为mouseenter和mouseleave事件指定处理函数    
 .keydown() 按下键盘（长时间按键，只返回一个事件）    
 .keypress() 按下键盘（长时间按键，将返回多个事件）    
 .keyup() 松开键盘    
 .load() 元素加载完毕    
 .mousedown() 按下鼠标    
 .mouseenter() 鼠标进入（进入子元素不触发）    
 .mouseleave() 鼠标离开（离开子元素不触发）    
 .mousemove() 鼠标在元素内部移动    
 .mouseout() 鼠标离开（离开子元素也触发）    
 .mouseover() 鼠标进入（进入子元素也触发）    
 .mouseup() 松开鼠标    
 .ready() DOM加载完成    
 .resize() 浏览器窗口的大小发生改变    
 .scroll() 滚动条的位置发生变化    
 .select() 用户选中文本框中的内容    
 .submit() 用户递交表单    
 .toggle() 根据鼠标点击的次数，依次运行多个函数    
 .unload()    
 用户离开页面    
　　以上这些事件在jQuery内部，都是.bind()的便捷方式。使用.bind()可以更灵活地控制事件，比如为多个事件绑定同一个函数：

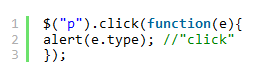
  
有时，你只想让事件运行一次，这时可以使用.one()方法。



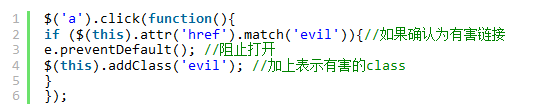
unbind()用来解除事件绑定。



所有的事件处理函数，都可以接受一个事件对象(event object)作为参数，比如下面例子中的e：

   
　　这个事件对象有一些很有用的属性和方法：

 event.pageX 事件发生时，鼠标距离网页左上角的水平距离 　　  
event.pageY 事件发生时，鼠标距离网页左上角的垂直距离  
event.type 事件的类型（比如click） 　　  
event.which 按下了哪一个键 　　  
event.data在事件对象上绑定数据，然后传入事件处理函数 　　  
event.target 事件针对的网页元素  
event.preventDefault() 阻止事件的默认行为（比如点击链接，会自动打开新页面）  
event.stopPropagation() 停止事件向上层元素冒泡  
　　在事件处理函数中，可以用this关键字，返回事件针对的DOM元素：



　　有两种方法，可以自动触发一个事件。一种是直接使用事件函数，另一种是使用.trigger()或.triggerHandler()。



九、特殊效果

　　jQuery允许对象呈现某些特殊效果。

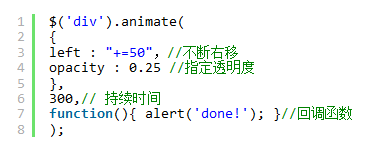
$('h1').show(); //展现一个h1标题  
常用的特殊效果如下：

.fadeIn() 淡入  
.fadeOut() 淡出  
.fadeTo() 调整透明度  
.hide() 隐藏元素  
.show() 显示元素  
.slideDown() 向下展开  
.slideUp() 向上卷起  
.slideToggle() 依次展开或卷起某个元素  
.toggle() 依次展示或隐藏某个元素  
除了.show()和.hide()，所有其他特效的默认执行时间都是400ms(毫秒)，但是你可以改变这个设置。

\* $('h1').fadeIn(300); // 300毫秒内淡入    
\* $('h1').fadeOut('slow'); //缓慢地淡出   
　　在特效结束后，可以指定执行某个函数。



　更复杂的特效，可以用.animate()自定义。

  
　　.stop()和.delay()用来停止或延缓特效的执行。  
　　$.fx.off如果设置为true，则关闭所有网页特效。

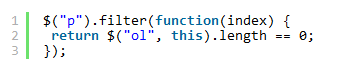
几个常见的筛选选择器：

filter()：筛选出与指定表达式匹配的元素集合。这个方法用于缩小匹配的范围。用逗号分隔多个表达式  
描述：保留子元素中不含有ol的元素。

HTML 代码:



jQuery 代码:



结果:



silce()：选取一个匹配的子集

描述：选择第一个p元素

HTML 代码:



jQuery 代码:



结果:



问题一：$(this)和this关键字在jQuery中有何异同???

这是一个简单却很好的问题。

$(this)返回的是一个jQuery对象，我们可以对它调用jQuery的特有方法，例如：text() 获取文本，val() 获取值，html()获取标签中的内容（如果有html语言，一并获取）。

this 代表当前元素，是javascript中关键字中的一个，表示上下文中的当前DOM元素。this不可以调用jQuery的方法，所以需要使用 $() 函数将this包裹，转换为jQuery对象---$(this).

问题二：jQuery 库中的 $() 是什么???

$()函数是JQuery函数的别称，就是一种标志，符合其思想：write less，do more。第一次接触可能有些不适应，是jQuery代码佶屈聱牙，晦涩难懂。我们需要适应一段时间，过一段时间，就会爱上其简洁，方便，飞起来。

$()函数用于将任何对象包裹成jQuery对象，接着就可以被允许调用定义在jQuery对象上的多个不同方法。甚至可以将一个选择器字符串传入 $()函数，它会返回一个包含所有匹配的DOM 元素数组的jQuery对象。可以用each()方法进行遍历里面的对象。