阮一峰：jQuery官方基础教程笔记

jQuery是目前使用最广泛的javascript函数库。

据统计，全世界排名前100万的网站，有46%使用jQuery，远远超过其他库。微软公司甚至把jQuery作为他们的官方库。

对于网页开发者来说，学会jQuery是必要的。因为它让你了解业界最通用的技术，为将来学习更高级的库打下基础，并且确实可以很轻松地做出许多复杂的效果。

虽然jQuery上手简单，比其他库容易学会，但是要全面掌握，却不轻松。因为它涉及到网页开发的方方面面，提供的各种方法和内部变化有上千种之多。初学者常常感到，入门很方便，提高很困难。

目前，互联网上最好的jQuery入门教材，是Rebecca Murphey写的《[jQuery基础教程](http://www.amazon.cn/mn/detailApp/ref=as_li_qf_sp_asin_tl?_encoding=UTF8&tag=vastwork-23&linkCode=as2&asin=B001ASJS2W&camp=536&creative=3200&creativeASIN=B001ASJS2W" \t "_blank)》（jQuery Fundamentals）。在Google里搜索”jQuery培训”，此书排在第一位。jQuery官方团队已经同意，把此书作为官方教程的基础。

这本书虽然是入门教材，但也足足有100多页。我对它做了一个详细的笔记，试图理清jQuery的设计思想，找出学习的脉络。我的目标是全面掌握jQuery，遇到问题的时候，心里有底，基本知道使用它的哪一个功能，然后可以迅速从手册中找到具体的写法。

下面就是我的笔记，它应该是目前网上不多的jQuery中文教程之一。你只需要一点javascript语言的基本知识，就能看懂它，在最短的时间里，掌握jQuery的所有主要方面（除了ajax和插件开发）。

===========================================

**jQuery设计思想**

原文网址：<http://jqfundamentals.com/book/>

阮一峰翻译整理

【目录】

一、选择网页元素

二、改变结果集

三、链式操作

四、元素的操作：取值和赋值

五、元素的操作：移动

六、元素的操作：复制、删除和创建

七、工具方法

八、事件操作

九、特殊效果

【正文】

# 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](http://www.ruanyifeng.com/blog/2009/03/css_selectors.html)：

$(document) //选择整个文档对象

|  |  |
| --- | --- |
| 1  2  3 | $('#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](http://api.jquery.com/category/selectors/)：

$('a:first') //选择网页中第一个a元素

|  |  |
| --- | --- |
| 1  2  3  4  5 | $('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](http://api.jquery.com/category/traversing/filtering/)，可以缩小结果集：

$('div').has('p'); //选择包含p元素的div元素

|  |  |
| --- | --- |
| 1  2  3  4 | $('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](http://api.jquery.com/category/traversing/tree-traversal/)：

$('div').next('p'); //选择div元素后面的第一个p元素

|  |  |
| --- | --- |
| 1  2  3  4 | $('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：

|  |  |
| --- | --- |
| 1 | $('div').find('h3').eq(2).html('Hello'); |

分解开来，就是下面这样：

$('div') //找到div元素

|  |  |
| --- | --- |
| 1  2  3 | .find('h3') //选择其中的h3元素  .eq(2) //选择第3个h3元素  .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()](http://api.jquery.com/end/)方法，使得结果集可以后退一步：

$('div')

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | .find('h3')  .eq(2)  .html('Hello')  .end() //退回到选中所有的h3元素的那一步  .eq(0) //选中第一个h3元素  .html('World'); //将它的内容改为World |

# 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

|  |  |
| --- | --- |
| 1 | $('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：  
[.html()](http://api.jquery.com/html/)取出或设置html内容

[.text()](http://api.jquery.com/text/) 取出或设置text内容

[.attr()](http://api.jquery.com/attr/) 取出或设置某个属性的值  
[.width()](http://api.jquery.com/width/) 取出或设置某个元素的宽度  
[.height()](http://api.jquery.com/height/) 取出或设置某个元素的高度  
[.val()](http://api.jquery.com/val/) 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()](http://api.jquery.com/text/)例外，它取出所有元素的text内容）。

# 五、元素的操作：移动

如果要移动选中的元素，有两种方法：一种是直接移动该元素，另一种是移动其他元素，使得目标元素达到我们想要的位置。

假定我们选中了一个div元素，需要把它移动到p元素后面。

第一种方法是使用[.insertAfter()](http://api.jquery.com/insertAfter/)，把div元素移动p元素后面：

|  |  |
| --- | --- |
| 1 | $('div').insertAfter('p'); |

第二种方法是使用[.after()](http://api.jquery.com/after/)，把p元素加到div元素前面：

|  |  |
| --- | --- |
| 1 | $('p').after('div'); |

表面上看，这两种方法的效果是一样的，唯一的不同似乎只是操作视角的不同。但是实际上，它们有一个重大差别，那就是返回的元素不一样。第一种方法返回div元素，第二种方法返回p元素。你可以根据需要，选择到底使用哪一种方法。

使用这种模式的操作方法，一共有四对：

[.insertAfter()](http://api.jquery.com/insertAfter/)和[.after()](http://api.jquery.com/after/)：在现存元素的外部，从后面插入元素  
[.insertBefore()](http://api.jquery.com/insertBefore/)和[.before()](http://api.jquery.com/before)：在现存元素的外部，从前面插入元素  
[.appendTo()](http://api.jquery.com/appendTo/)和[.append()](http://api.jquery.com/append)：在现存元素的内部，从后面插入元素  
[.prependTo()](http://api.jquery.com/prependTo/)和[.prepend()](http://api.jquery.com/prepend)：在现存元素的内部，从前面插入元素

# 六、元素的操作：复制、删除和创建

复制元素使用[.clone()](http://api.jquery.com/clone/)。

删除元素使用[.remove()](http://api.jquery.com/remove/)和[.detach()](http://api.jquery.com/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()](http://api.jquery.com/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：

|  |  |
| --- | --- |
| 1  2  3  4  5 | $('Hello  ');   $('<li class="new">new list item');   $('ul').append('list item  '); |

# 七、工具方法

jQuery设计思想之六：除了对选中的元素进行操作以外，还提供一些与元素无关的[工具方法](http://api.jquery.com/category/utilities/" \t "_blank)（utility）。不必选中元素，就可以直接使用这些方法。

如果你懂得Javascript语言的[继承原理](http://www.ruanyifeng.com/blog/2011/06/designing_ideas_of_inheritance_mechanism_in_javascript.html" \t "_blank)，那么就能理解工具方法的实质。它是定义在jQuery构造函数上的方法，即jQuery.method()，所以可以直接使用。而那些操作元素的方法，是定义在构造函数的prototype对象上的方法，即jQuery.prototype.method()，所以必须生成实例（即选中元素）后使用。如果不理解这种区别，问题也不大，只要把工具方法理解成，是像javascript原生函数那样，可以直接使用的方法就行了。

常用的工具方法有以下几种：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | $.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设计思想之七，就是把[事件](http://api.jquery.com/category/events/" \t "_blank)直接绑定在网页元素之上。

|  |  |
| --- | --- |
| 1  2  3 | $('p').click(function(){  alert('Hello');  }); |

目前，jQuery主要支持以下事件：

[.blur()](http://api.jquery.com/blur/)?表单元素失去焦点。  
[.change()](http://api.jquery.com/change/)?表单元素的值发生变化  
[.click()](http://api.jquery.com/click/)?鼠标单击  
[.dblclick()](http://api.jquery.com/dblclick/)?鼠标双击  
[.focus()](http://api.jquery.com/focus/)?表单元素获得焦点  
[.focusin()](http://api.jquery.com/focusin/)?子元素获得焦点  
[.focusout()](http://api.jquery.com/focusout/)?子元素失去焦点  
[.hover()](http://api.jquery.com/hover/)?同时为mouseenter和mouseleave事件指定处理函数  
[.keydown()](http://api.jquery.com/keydown/)?按下键盘（长时间按键，只返回一个事件）  
[.keypress()](http://api.jquery.com/keypress/)?按下键盘（长时间按键，将返回多个事件）  
[.keyup()](http://api.jquery.com/keyup/)?松开键盘  
[.load()](http://api.jquery.com/load-event/)?元素加载完毕  
[.mousedown()](http://api.jquery.com/mousedown/)?按下鼠标  
[.mouseenter()](http://api.jquery.com/mouseenter/)?鼠标进入（进入子元素不触发）  
[.mouseleave()](http://api.jquery.com/mouseleave/)?鼠标离开（离开子元素不触发）  
[.mousemove()](http://api.jquery.com/mousemove/)?鼠标在元素内部移动  
[.mouseout()](http://api.jquery.com/mouseleave/)?鼠标离开（离开子元素也触发）  
[.mouseover()](http://api.jquery.com/mouseover/)?鼠标进入（进入子元素也触发）  
[.mouseup()](http://api.jquery.com/mouseup/)?松开鼠标  
[.ready()](http://api.jquery.com/ready/)?DOM加载完成  
[.resize()](http://api.jquery.com/resize/)?浏览器窗口的大小发生改变  
[.scroll()](http://api.jquery.com/scroll/)?滚动条的位置发生变化  
[.select()](http://api.jquery.com/select/)?用户选中文本框中的内容  
[.submit()](http://api.jquery.com/submit/)?用户递交表单  
[.toggle()](http://api.jquery.com/toggle-event/)?根据鼠标点击的次数，依次运行多个函数  
[.unload()](http://api.jquery.com/unload/)?用户离开页面

以上这些事件在jQuery内部，都是[.bind()](http://api.jquery.com/bind/" \t "_blank)的便捷方式。使用[.bind()](http://api.jquery.com/bind/" \t "_blank)可以更灵活地控制事件，比如为多个事件绑定同一个函数：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $('input').bind(  'click change', //同时绑定click和change事件  function() {  alert('Hello');  }  ); |

有时，你只想让事件运行一次，这时可以使用[.one()](http://api.jquery.com/one/" \t "_blank)方法。  
$(“p”).one(“click”, function() {  
alert(“Hello”); //只运行一次，以后的点击不会运行  
});

|  |  |
| --- | --- |
| 1 | <a href="http://api.jquery.com/unbind/" target="\_blank">.unbind()</a>用来解除事件绑定。 |

$(‘p’).unbind(‘click’);  
所有的事件处理函数，都可以接受一个[事件对象](http://api.jquery.com/category/events/event-object/" \t "_blank)（event object）作为参数，比如下面例子中的e：

|  |  |
| --- | --- |
| 1  2  3 | $("p").click(function(e) {  alert(e.type); // "click"  }); |

这个事件对象有一些很有用的属性和方法：

[event.pageX](http://api.jquery.com/event.pageX/)?事件发生时，鼠标距离网页左上角的水平距离  
[event.pageY](http://api.jquery.com/event.pageY/)?事件发生时，鼠标距离网页左上角的垂直距离  
[event.type](http://api.jquery.com/event.type/)?事件的类型（比如click）  
[event.which](http://api.jquery.com/event.which/)?按下了哪一个键  
[event.data](http://api.jquery.com/event.data/)?在事件对象上绑定数据，然后传入事件处理函数  
[event.target](http://api.jquery.com/event.target/)?事件针对的网页元素  
[event.preventDefault()](http://api.jquery.com/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()](http://api.jquery.com/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：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $('a').click(function(e) {  if ($(this).attr('href').match('evil')) { //如果确认为有害链接  e.preventDefault(); //阻止打开  $(this).addClass('evil'); //加上表示有害的class  }  }); |

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()](http://api.jquery.com/trigger/" \t "_blank)或[.triggerHandler()](http://api.jquery.com/triggerHandler/" \t "_blank)。

|  |  |
| --- | --- |
| 1  2 | $('a').click();  $('a').trigger('click'); |

# 九、特殊效果

最后，jQuery允许对象呈现某些[特殊效果](http://api.jquery.com/category/effects/" \t "_blank)。

|  |  |
| --- | --- |
| 1 | $('h1').show(); //展现一个h1标题 |

常用的特殊效果如下：  
[.fadeIn()](http://api.jquery.com/fadeIn/)?淡入  
[.fadeOut()](http://api.jquery.com/fadeOut/)?淡出  
[.fadeTo()](http://api.jquery.com/fadeTo/)?调整透明度  
[.hide()](http://api.jquery.com/hide/)?隐藏元素  
[.show()](http://api.jquery.com/show/)?显示元素  
[.slideDown()](http://api.jquery.com/slideDown/)?向下展开  
[.slideUp()](http://api.jquery.com/slideUp/)?向上卷起  
[.slideToggle()](http://api.jquery.com/slideToggle/)?依次展开或卷起某个元素  
[.toggle()](http://api.jquery.com/toggle/)?依次展示或隐藏某个元素  
除了[.show()](http://api.jquery.com/show/)和[.hide()](http://api.jquery.com/hide/" \t "_blank)，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。

|  |  |
| --- | --- |
| 1 | $('p').fadeOut(300, function() { $(this).remove(); }); |

更复杂的特效，可以用[.animate()](http://api.jquery.com/animate/)自定义。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | $('div').animate(  {  left : "+=50", //不断右移  opacity : 0.25 //指定透明度  },  300, // 持续时间  function() { alert('done!'); } //回调函数  ); |

[.stop()](http://api.jquery.com/stop/)和[.delay()](http://api.jquery.com/delay/)用来停止或延缓特效的执行。

[$.fx.off](http://api.jquery.com/jQuery.fx.off/)如果设置为true，则关闭所有网页特效。

（完）

作者：[阮一峰](http://www.ruanyifeng.com/blog/2011/07/jquery_fundamentals.html)