

软大师的Jquery

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的基本设计思想和主要用法，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Javascript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

jQuery设计思想之二，就是提供各种强大的过滤器，对结果集进行筛选，缩小选择结果。

```
$('#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

jQuery设计思想之三，就是最终选中网页元素以后，可以对它进行一系列操作，并且所有操作可以连接在一起，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：

```
$( 'div' ) //找到div元素
    .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()方法，使得结果集可以后退一步：

```
$( 'div' )
    .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设计思想之四，就是使用同一个函数，来完成取值（getter）和赋值（setter），即“取值器”与“赋值器”合一。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：

- .html() 取出或设置html内容
- .text() 取出或设置text内容
- .attr() 取出或设置某个属性的值
- .width() 取出或设置某个元素的宽度
- .height() 取出或设置某个元素的高度
- .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()例外，它取出所有元素的text内容）。

五、元素的操作：移动

jQuery设计思想之五，就是提供两组方法，来操作元素在网页中的位置移动。一组方法是直接移动该元素，另一组方法是移动其他元素，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()，把div元素移动p元素后面：

```
$( 'div' ).insertAfter( $( 'p' ) );
```

第二种方法是使用.after()，把p元素加到div元素前面：

```
$( '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()和.after()：在现存元素的外部，从后面插入元素
- .insertBefore()和.before()：在现存元素的外部，从前面插入元素
- .appendTo()和.append()：在现存元素的内部，从后面插入元素
- .prependTo()和.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

除了元素的位置移动之外，jQuery还提供其他几种操作元素的重要方法。

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：

```
$('#<p>Hello</p>');  
$('#<li class="new">new list item</li>');  
$('#ul').append('<li>list item</li>');
```

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（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，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：

- `$.trim()` 去除字符串两端的空格。
- `$.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，就是把事件直接绑定在网页元素之上。

```
$('#p').click(function(){  
    alert('Hello');  
});
```

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：

```
$('#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()方法。

```
$("#p").one("click", function() {
    alert("Hello"); //只运行一次，以后的点击不会运行
});
```

.unbind()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（event object）作为参数，比如下面例子中的e：

```
$("#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 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：

```
$('#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()或.triggerHandler()。

```
$('#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。

```
$('#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。

```
$('#hl').fadeIn(300); // 300毫秒内淡入
$('#hl').fadeOut('slow'); // 缓慢地淡出
```

在特效结束后，可以指定执行某个函数。

```
$('#p').fadeOut(300, function() { $(this).remove(); });
```

更复杂的特效，可以用[.animate\(\)](#)自定义。

```
$('#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\(\)](#)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。