

jQuery 常用功能大全

1、关于页面元素的引用

通过 jquery 的 `$()` 引用元素包括通过 id、class、元素名以及元素的层级关系及 dom 或者 xpath 条件等方法，且返回的对象为 jquery 对象（集合对象），不能直接调用 dom 定义的方法。

2、jQuery 对象与 dom 对象的转换

只有 jquery 对象才能使用 jquery 定义的方法。注意 dom 对象和 jquery 对象是有区别的，调用方法时要注意操作的是 dom 对象还是 jquery 对象。

普通的 dom 对象一般可以通过 `$()` 转换成 jquery 对象。

如：`$(document.getElementById_x("msg"))` 则为 jquery 对象，可以使用 jquery 的方法。

由于 jquery 对象本身是一个集合。所以如果 jquery 对象要转换为 dom 对象则必须取出其中的某一项，一般可通过索引取出。

如：`$("#msg")[0]`，`$("#div").eq(1)[0]`，`$("#div").get()[1]`，`$("#td")[5]` 这些都是 dom 对象，可以使用 dom 中的方法，但不能再使用 JQuery 的方法。

以下几种写法都是正确的：

```
$("#msg").html();  
$("#msg")[0].innerHTML;  
$("#msg").eq(0)[0].innerHTML;  
$("#msg").get(0).innerHTML;
```

3、如何获取 jQuery 集合的某一项

对于获取的元素集合，获取其中的某一项（通过索引指定）可以使用 `eq` 或 `get(n)` 方法或者索引号获取，要注意，`eq` 返回的是 jquery 对象，而 `get(n)` 和索引返回的是 dom 元素对象。对于 jquery 对象只能使用 jquery 的方法，而 dom 对象只能使用 dom 的方法，如要获取第三个 `<div>` 元素的内容。有如下两种方法：

```
$("#div").eq(2).html(); //调用 jquery 对象的方法  
$("#div").get(2).innerHTML; //调用 dom 的方法属性
```

4、同一函数实现 set 和 get

Jquery 中的很多方法都是如此，主要包括如下几个：

```
$("#msg").html(); //返回 id 为 msg 的元素节点的 html 内容。
```

```
$("#msg").html("<b>new content</b>");//将 “<b>new  
content</b>” 作为 html 串写入 id 为 msg 的元素节点内容中, 页  
面显示粗体的 new content
```

```
$("#msg").text(); //返回 id 为 msg 的元素节点的文本内容。
```

```
$("#msg").text("<b>new content</b>");//将 “<b>new  
content</b>” 作为普通文本串写入 id 为 msg 的元素节点内容中,  
页面显示<b>new content</b>
```

```
$("#msg").height(); //返回 id 为 msg 的元素的高度
```

```
$("#msg").height("300"); //将 id 为 msg 的元素的高度设为 300
```

```
$("#msg").width(); //返回 id 为 msg 的元素的宽度
```

```
$("#msg").width("300"); //将 id 为 msg 的元素的宽度设为 300
```

```
$("#input").val(""); //返回表单输入框的 value 值
```

```
$("#input").val("test"); //将表单输入框的 value 值设为 test
```

```
$("#msg").click(); //触发 id 为 msg 的元素的单击事件
```

```
$("#msg").click(fn); //为 id 为 msg 的元素单击事件添加函数  
同样 blur, focus, Select, submit 事件都可以有着两种调用方法
```

5、集合处理功能

对于 jquery 返回的集合内容无需我们自己循环遍历并对每个对象分别做处理，jquery 已经为我们提供的很方便的方法进行集合的处理。

包括两种形式：

```
$("#p").each(function(i){this.style.color=['#f00','#0f0','#  
00f'][ i ]})//为索引分别为 0, 1, 2 的 p 元素分别设定不同的字  
体颜色。
```

```
$("#tr").each(function(i){this.style.backgroundColor=['#ccc  
, '#fff'][i%2]})//实现表格的隔行换色效果
```

`$("p").click(function(){alert($(this).html())})` //为每个 p 元素增加了 click 事件，单击某个 p 元素则弹出其内容

6、扩展我们需要的功能

```
$.extend({
min: function(a, b){return a < b?a:b; },
max: function(a, b){return a > b?a:b; }
}); //为 jquery 扩展了 min,max 两个方法
使用扩展的方法（通过“$.方法名”调用）：
alert("a=10,b=20,max="+$.max(10,20)+" ,min="+$.min(10,20));
```

7、支持方法的连写

所谓连写，即可以对一个 jquery 对象连续调用各种不同的方法。

例如：

```
$("p").click(function(){alert($(this).html())})
.mouseover(function(){alert('mouse over event')})
.each(function(i){this.style.color=['#f00','#0f0','#00f'][
i ]});
```

8、操作元素的样式

主要包括以下几种方式：

```
$("#msg").css("background"); //返回元素的背景颜色
$("#msg").css("background","#ccc") //设定元素背景为灰色
$("#msg").height(300); $("#msg").width("200"); //设定宽高
$("#msg").css({ color: "red", background: "blue" }); //以名
值对的形式设定样式
$("#msg").addClass("S elect"); //为元素增加名称为 S elect 的
class
$("#msg").removeClass("S elect"); //删除元素名称为 S elect
的 class
$("#msg").toggleClass("S elect"); //如果存在（不存在）就删
除（添加）名称为 S elect 的 class
```

9、完善的事件处理功能

Jquery 已经为我们提供了各种事件处理方法，我们无需在 html 元

素上直接写事件，而可以直接为通过 jquery 获取的对象添加事件。
如：

```
$("#msg").click(function() {alert("good");}) //为元素添加了  
单击事件  
$("p").click(function(i) {this.style.color=['#f00','#0f0','#00f'][i]})//为三个不同的 p 元素单击事件分别设定不同的处理
```

jQuery 中几个自定义的事件：

(1) hover(fn1,fn2)：一个模仿悬停事件（鼠标移动到一个对象上面及移出这个对象）的方法。当鼠标移动到一个匹配的元素上面时，会触发指定的第一个函数。当鼠标移出这个元素时，会触发指定的第二个函数。

//当鼠标放在表格的某行上时将 class 置为 over，离开时置为 out。

```
$("tr").hover(function() {  
$(this).addClass("over");  
},  
function() {  
$(this).addClass("out");  
});
```

(2) ready(fn)：当 DOM 载入就绪可以查询及操纵时绑定一个要执行的函数。

```
$(document).ready(function() {alert("Load Success");})//页面  
加载完毕提示“Load Success”，相当于 onload 事件。与$(fn)等价
```

(3) toggle(evenFn, oddFn)：每次点击时切换要调用的函数。如果点击了一个匹配的元素，则触发指定的第一个函数，当再次点击同一元素时，则触发指定的第二个函数。随后的每次点击都重复对这两个函数的轮番调用。

//每次点击时轮换添加和删除名为 S elected 的 class。

```
$("p").toggle(function() {  
$(this).addClass("S elected");  
},function() {  
$(this).removeClass("S elected");  
});
```

(4) trigger(eventtype)：在每一个匹配的元素上触发某类事件。
例如：

```
$("p").trigger("click"); //触发所有 p 元素的 click 事件
```

(5) bind(eventtype,fn), unbind(eventtype)：事件的绑定与反绑定

从每一个匹配的元素中（添加）删除绑定的事件。

例如：

```
$("#p").bind("click", function(){alert($(this).text());});  
//为每个 p 元素添加单击事件  
$("#p").unbind(); //删除所有 p 元素上的所有事件  
$("#p").unbind("click") //删除所有 p 元素上的单击事件
```

10、几个实用特效功能

其中 toggle()和 slidetoggle()方法提供了状态切换功能。

如 toggle()方法包括了 hide()和 show()方法。

slideToggle()方法包括了 slideDown()和 slideUp 方法。

11、几个有用的 jQuery 方法

\$.browser. 浏览器类型：检测浏览器类型。有效参数：safari, opera, msie, mozilla。如检测是否 ie: \$.browser.isie, 是 ie 浏览器则返回 true。

\$.each(obj, fn)：通用的迭代函数。可用于近似地迭代对象和数组（代替循环）。如

```
$.each( [0,1,2], function(i, n){ alert( "Item #" + i + ": " + n ); });
```

等价于：

```
var tempArr=[0,1,2];  
for(var i=0;i<tempArr.length;i++){  
alert("Item #"+i+": "+tempArr[ i ]);  
}
```

也可以处理 json 数据，如

```
$.each( { name: "John", lang: "JS" }, function(i, n){ alert( "Name: " + i + ", Value: " + n ); });
```

结果为：

Name:name, Value:John

Name:lang, Value:JS

\$.extend(target, prop1, propN)：用一个或多个其他对象来扩展一个对象，返回这个被扩展的对象。这是 jquery 实现的继承方式如：

\$.extend(settings, options); //合并 settings 和 options, 并将合并结果返回 settings 中, 相当于 options 继承 setting 并将继承结果保存在 setting 中。

```
var settings = $.extend({}, defaults, options); // 合并
defaults 和 options, 并将合并结果返回到 setting 中而不覆盖
default 内容。
可以有多个参数 (合并多项并返回)
$.map(array, fn): 数组映射。把一个数组中的项目 (处理转换后)
保存到另一个新数组中, 并返回生成的新数组。如:
var tempArr=$.map( [0,1,2], function(i){ return i + 4; });
tempArr 内容为: [4,5,6]
var tempArr=$.map( [0,1,2], function(i){ return i > 0 ? i +
1 : null; });
tempArr 内容为: [2,3]
$.merge(arr1,arr2): 合并两个数组并删除其中重复的项目。
如: $.merge( [0,1,2], [2,3,4] ) //返回 [0,1,2,3,4]
$.trim(str): 删除字符串两端的空白字符。
如: $.trim(" hello, how are you? "); //返回 "hello,how are you?"
```

12、解决自定义方法或其他类库与 jQuery 的冲突

很多时候我们自己定义了\$(id)方法来获取一个元素,或者其他的一些 js 类库如 prototype 也都定义了\$方法, 如果同时把这些内容放在一起就会引起变量方法定义冲突, JQuery 对此专门提供了方法用于解决此问题。

使用 jquery 中的 jQuery.noConflict(); 方法即可把变量\$的控制权让渡给第一个实现它的那个库或之前自定义的\$方法。之后应用 JQuery 的时候只要将所有的\$换成 jQuery 即可, 如原来引用对象方法\$("#msg")改为 jQuery("#msg")。如:

```
jQuery.noConflict(); // 开始使用 jQuery
jQuery("div p").hide(); // 使用其他库的 $()
$("content").style.display = 'none';
```

转载请自觉注明原文: <http://www.jq-school.com/Show.aspx?id=6>