目录

[1. DOM 1](#_Toc508878479)

[1) Attribute 1](#_Toc508878480)

[2) Manipulation 1](#_Toc508878481)

[3) Traversing 1](#_Toc508878482)

[4) Core 1](#_Toc508878483)

## DOM

### Attribute

$("p").addClass(css中定义的样式类型); 给某个元素添加样式

$("img").attr({src:"test.jpg",alt:"test Image"}); 给某个元素添加属性/值，参数是map

$("img").attr("src","test.jpg"); 给某个元素添加属性/值

$("img").attr("title", function() { return this.src }); 给某个元素添加属性/值

$("元素名称").html(); 获得该元素内的内容（元素，文本等）

$("元素名称").html("<b>new stuff</b>"); 给某元素设置内容

$("元素名称").removeAttr("属性名称") 给某元素删除指定的属性以及该属性的值

$("元素名称").removeClass("class") 给某元素删除指定的样式

$("元素名称").text(); 获得该元素的文本

$("元素名称").text(value); 设置该元素的文本值为value

$("元素名称").toggleClass(class) 当元素存在参数中的样式的时候取消,如果不存在就设置此样式

$("input元素名称").val(); 获取input元素的值

$("input元素名称").val(value); 设置input元素的值为value

### Manipulation

$("元素名称").after(content); 在匹配元素后面添加内容  
$("元素名称").append(content); 将content作为元素的内容插入到该元素的后面  
$("元素名称").appendTo(content); 在content后接元素  
$("元素名称").before(content); 与after方法相反  
$("元素名称").clone(布尔表达式) 当布尔表达式为真时，克隆元素（无参时，当作true处理）  
$("元素名称").empty() 将该元素的内容设置为空  
$("元素名称").insertAfter(content); 将该元素插入到content之后  
$("元素名称").insertBefore(content); 将该元素插入到content之前  
$("元素").prepend(content); 将content作为该元素的一部分，放到该元素的最前面  
$("元素").prependTo(content); 将该元素作为content的一部分，放content的最前面  
$("元素").remove(); 删除所有的指定元素  
$("元素").remove("exp"); 删除所有含有exp的元素  
$("元素").wrap("html"); 用html来包围该元素  
$("元素").wrap(element); 用element来包围该元素

### Traversing

add(expr)  
add(html)  
add(elements)  
children(expr)  
contains(str)  
end()  
filter(expression)  
filter(filter)  
find(expr)  
is(expr)  
next(expr)  
not(el)  
not(expr)  
not(elems)  
parent(expr)  
parents(expr)  
prev(expr)  
siblings(expr)

### Core

$(html).appendTo("body") 相当于在body中写了一段html代码  
$(elems) 获得DOM上的某个元素  
$(function(){……..}); 执行一个函数  
$("div > p").css("border", "1px solid gray"); 查找所有div的子节点p，添加样式  
$("input:radio", document.forms[0]) 在当前页面的第一个表单中查找所有的单选按钮  
$.extend(prop) prop是一个jQuery对象，  
举例：

|  |
| --- |
| 1. jQuery.extend({ 2. min: function(a, b) { return a **<** **b** ? a : b; }, 3. max: function(a, b) { return a **>** b ? a : b; } 4. }); |

jQuery( expression, [context] ) ---$( expression, [context]); 在默认情况下，$()查询的是当前HTML文档中的DOM元素。

each( callback ) 以每一个匹配的元素作为上下文来执行一个函数

|  |
| --- |
| 1. 举例：1 2. $("span").click(function(){ 3. $("li").each(function(){ 4. $(this).toggleClass("example"); 5. }); 6. }); 7. 举例：2 8. $("button").click(function () { 9. $("div").each(function (index, domEle) { 10. // domEle == this 11. $(domEle).css("backgroundColor", "yellow"); 12. if ($(this).is("#stop")) { 13. $("span").text("Stopped at div index #" + index); 14. return false; 15. } 16. }); 17. }); |

## jQuery Event

ready(fn); $(document).ready() 注意在body中没有onload事件，否则该函数不能执行。在每个页面中可以有很多个函数被加载执行，按照fn的顺序来执行。

bind( type, [data], fn ) 为每一个匹配元素的特定事件（像click）绑定一个或多个事件处理器函数。可能的事件属性有：blur, focus, load, resize, scroll, unload, click, dblclick, mousedown, mouseup, mousemove,mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress,keyup, error。

one( type, [data], fn ) 为每一个匹配元素的特定事件（像click）绑定一个或多个事件处理器函数。在每个对象上，这个事件处理函数只会被执行一次。其他规则与bind()函数相同。

trigger( type, [data] ) 在每一个匹配的元素上触发某类事件。

triggerHandler( type, [data] ) 这一特定方法会触发一个元素上特定的事件(指定一个事件类型)，同时取消浏览器对此事件的默认行动。

unbind( [type], [data] ) 反绑定，从每一个匹配的元素中删除绑定的事件。

$("p").unbind() 移除所有段落上的所有绑定的事件。

$("p").unbind( "click" ) 移除所有段落上的click事件。

hover( over, out ) over,out 都是方法, 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。

|  |
| --- |
| 1. ("p").hover(function(){ 2. $(this).addClass("over"); 3. }, 4. function(){ 5. $(this).addClass("out"); 6. } 7. ); |

|  |
| --- |
| **[html]** [view plain](http://blog.csdn.net/tanga842428/article/details/52432026) [copy](http://blog.csdn.net/tanga842428/article/details/52432026)   1. $("p").toggle(function(){ 2. $(this).addClass("selected"); 3. }, 4. function(){ 5. $(this).removeClass("selected"); 6. } 7. ); |