**Jquery中的筛选部分讲解**

## **eq(index|-index)**

### 概述

获取第N个元素

### 参数

#### ****index****IntegerV1.1.2

一个整数，指示元素基于0的位置,这个元素的位置是从0算起。

#### ****-index****IntegerV1.4

一个整数，指示元素的位置，从集合中的最后一个元素开始倒数。(1算起)

### 示例

#### 参数index描述:

获取匹配的第二个元素

##### HTML 代码:

<p> This is just a test.</p> <p> So is this</p>

##### jQuery 代码:

$("p").eq(1)

##### 结果:

[ <p> So is this</p> ]

#### 参数-index描述:

获取匹配的第二个元素

##### HTML 代码:

<p> This is just a test.</p> <p> So is this</p>

##### jQuery 代码:

$("p").eq(-2)

##### 结果:

[ <p> This is just a test.</p> ]

## **first()**

### V1.4概述

获取第一个元素

### 示例

#### 描述:

获取匹配的第一个元素

##### HTML 代码:

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

##### jQuery 代码:

$('li').first()

##### 结果:

[ <li>list item 1</li> ]

## **last()**

### V1.4概述

获取最后个元素

### 示例

#### 描述:

获取匹配的最后个元素

##### HTML 代码:

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

##### jQuery 代码:

$('li').last()

##### 结果:

[ <li>list item 5</li> ]

## **hasClass(class)**

### 概述

检查当前的元素是否含有某个特定的类，如果有，则返回true。

这其实就是 is("." + class)。

### 参数

#### ****class****StringV1.2

用于匹配的类名

### 示例

#### 描述:

给包含有某个类的元素进行一个动画。

##### HTML 代码:

<div class="protected"></div><div></div>

##### jQuery 代码:

$("div").click(function(){

if ( $(this).hasClass("protected") )

$(this)

.animate({ left: -10 })

.animate({ left: 10 })

.animate({ left: -10 })

.animate({ left: 10 })

.animate({ left: 0 });

});

## **filter(expr|obj|ele|fn)**

### 概述

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

### 参数

#### ****expr****StringV1.0

字符串值，包含供匹配当前元素集合的选择器表达式。

#### ****jQuery object****objectV1.0

现有的jQuery对象，以匹配当前的元素。

#### ****element****ExpressionV1.4

一个用于匹配元素的DOM元素。

#### ****function(index)****FunctionV1.4

一个函数用来作为测试元素的集合。它接受一个参数index，这是元素在jQuery集合的索引。在函数， this指的是当前的DOM元素。

### 示例

#### 参数selector描述:

保留带有select类的元素

##### HTML 代码:

<p>Hello</p><p>Hello Again</p><p class="selected">And Again</p>

##### jQuery 代码:

$("p").filter(".selected")

##### 结果:

[ <p class="selected">And Again</p> ]

#### 参数selector描述:

保留第一个以及带有select类的元素

##### HTML 代码:

<p>Hello</p><p>Hello Again</p><p class="selected">And Again</p>

##### jQuery 代码:

$("p").filter(".selected, :first")

##### 结果:

[ <p>Hello</p>, <p class="selected">And Again</p> ]

#### 回调函数 描述:

保留子元素中不含有ol的元素。

##### HTML 代码:

<p><ol><li>Hello</li></ol></p><p>How are you?</p>

##### jQuery 代码:

$("p").filter(function(index) {

return $("ol", this).length == 0;

});

##### 结果:

[ <p>How are you?</p> ]

## **is(expr|obj|ele|fn)**

### 概述

根据选择器、DOM元素或 jQuery 对象来检测匹配元素集合，如果其中至少有一个元素符合这个给定的表达式就返回true。

如果没有元素符合，或者表达式无效，都返回'false'。 '''注意：'''在jQuery 1.3中才对所有表达式提供了支持。在先前版本中，如果提供了复杂的表达式，比如层级选择器（比如 + , ~ 和 > ），始终会返回true

### 参数

#### ****expr****StringV1.0

字符串值，包含供匹配当前元素集合的选择器表达式。

#### ****jQuery object****objectV1.6

现有的jQuery对象，以匹配当前的元素。

#### ****element****ExpressionV1.6

一个用于匹配元素的DOM元素。

#### ****function(index)****FunctionV1.6

一个函数用来作为测试元素的集合。它接受一个参数index，这是元素在jQuery集合的索引。在函数， this指的是当前的DOM元素。

### 示例

#### 参数expr 描述:

由于input元素的父元素是一个表单元素，所以返回true。

##### HTML 代码:

<form><input type="checkbox" /></form>

##### jQuery 代码:

$("input[type='checkbox']").parent().is("form")

##### 结果:

true

#### 回调函数 描述:

判断点击li标签增加背景色为红色，如果点击的是第2个strong,当前的li增加背景色为绿色，

##### HTML 代码:

<ul>

<li><strong>list</strong> item 1 - one strong tag</li>

<li><strong>list</strong> item <strong>2</strong> - two <span>strong tags</span></li>

<li>list item 3</li>

</ul>

##### jQuery 代码:

$("li").click(function() {

var $li = $(this),

isWithTwo = $li.is(function() {

return $('strong', this).length === 2;

});

if ( isWithTwo ) {

$li.css("background-color", "green");

} else {

$li.css("background-color", "red");

}

});

## **map(callback)**

### 概述

将一组元素转换成其他数组（不论是否是元素数组）

你可以用这个函数来建立一个列表，不论是值、属性还是CSS样式，或者其他特别形式。这都可以用'$.map()'来方便的建立。

### 参数

#### ****callback****FunctionV1.2

给每个元素执行的函数

### 示例

#### 描述:

把form中的每个input元素的值建立一个列表。

##### HTML 代码:

<p><b>Values: </b></p>

<form>

<input type="text" name="name" value="John"/>

<input type="text" name="password" value="password"/>

<input type="text" name="url" value="http://ejohn.org/"/>

</form>

##### jQuery 代码:

$("p").append( $("input").map(function(){

return $(this).val();

}).get().join(", ") );

##### 结果:

[ <p>John, password, http://ejohn.org/</p> ]

## **has(expr|ele)**

### 概述

保留包含特定后代的元素，去掉那些不含有指定后代的元素。

.has()方法将会从给定的jQuery对象中重新创建一组匹配的对象。提供的选择器会一一测试原先那些对象的后代，含有匹配后代的对象将得以保留。

### 参数

#### ****expr****StringV1.4

一个选择器字符串。

#### ****element****DOMElementV1.4

一个DOM元素

### 示例

#### 描述:

给含有ul的li加上背景色

##### HTML 代码:

<ul>

<li>list item 1</li>

<li>list item 2

<ul>

<li>list item 2-a</li>

<li>list item 2-b</li>

</ul>

</li>

<li>list item 3</li>

<li>list item 4</li>

</ul>

##### jQuery 代码:

$('li').has('ul').css('background-color', 'red');

## **not(expr|ele|fn)**

### 概述

删除与指定表达式匹配的元素

### 参数

#### ****expr****StringV1.0

一个选择器字符串。

#### ****element****DOMElementV1.0

一个DOM元素

#### ****function(index)****FunctionV1.4

一个用来检查集合中每个元素的函数。this是当前的元素。

### 示例

#### 描述:

从p元素中删除带有 select 的ID的元素

##### HTML 代码:

<p>Hello</p><p id="selected">Hello Again</p>

##### jQuery 代码:

$("p").not( $("#selected")[0] )

##### 结果:

[ <p>Hello</p> ]

## **slice(start, [end])**

### 概述

选取一个匹配的子集

与原来的slice方法类似

### 参数

#### ****start****IntegerV1.1.4

开始选取子集的位置。第一个元素是0.如果是负数，则可以从集合的尾部开始选起。

#### ****end****IntegerV1.1.4

结束选取自己的位置，如果不指定，则就是本身的结尾。

### 示例

#### 描述:

选择第一个p元素

##### HTML 代码:

<p>Hello</p><p>cruel</p><p>World</p>

##### jQuery 代码:

$("p").slice(0, 1).wrapInner("<b></b>");

##### 结果:

[ <p><b>Hello</b></p> ]

#### 描述:

选择前两个p元素

##### HTML 代码:

<p>Hello</p><p>cruel</p><p>World</p>

##### jQuery 代码:

$("p").slice(0, 2).wrapInner("<b></b>");

##### 结果:

[ <p><b>Hello</b></p>,<p><b>cruel</b></p> ]

#### 描述:

只选取第二个p元素

##### HTML 代码:

<p>Hello</p><p>cruel</p><p>World</p>

##### jQuery 代码:

$("p").slice(1, 2).wrapInner("<b></b>");

##### 结果:

[ <p><b>cruel</b></p> ]

#### 描述:

只选取第二第三个p元素

##### HTML 代码:

<p>Hello</p><p>cruel</p><p>World</p>

##### jQuery 代码:

$("p").slice(1).wrapInner("<b></b>");

##### 结果:

[ <p><b>cruel</b></p>, <p><b>World</b></p> ]

#### 描述:

选取第最后一个p元素

##### HTML 代码:

<p>Hello</p><p>cruel</p><p>World</p>

##### jQuery 代码:

$("p").slice(-1).wrapInner("<b></b>");

##### 结果:

[ <p><b>World</b></p> ]

## **children([expr])**

### 概述

取得一个包含匹配的元素集合中每一个元素的所有子元素的元素集合。

可以通过可选的表达式来过滤所匹配的子元素。注意：parents()将查找所有祖辈元素，而children()只考虑子元素而不考虑所有后代元素。

### 参数

#### ****expr****StringV1.0

用以过滤子元素的表达式

### 示例

#### 描述:

查找DIV中的每个子元素。

##### HTML 代码:

<p>Hello</p><div><span>Hello Again</span></div><p>And Again</p>

##### jQuery 代码:

$("div").children()

##### 结果:

[ <span>Hello Again</span> ]

#### 描述:

在每个div中查找 .selected 的类。

##### HTML 代码:

<div><span>Hello</span><p class="selected">Hello Again</p><p>And Again</p></div>

##### jQuery 代码:

$("div").children(".selected")

##### 结果:

[ <p class="selected">Hello Again</p> ]

## jQuery**closest(expr,[con]|obj|ele)**

### 概述

jQuery 1.3新增。从元素本身开始，逐级向上级元素匹配，并返回最先匹配的元素。。

closest会首先检查当前元素是否匹配，如果匹配则直接返回元素本身。如果不匹配则向上查找父元素，一层一层往上，直到找到匹配选择器的元素。如果什么都没找到则返回一个空的jQuery对象。

closest和parents的主要区别是：1，前者从当前元素开始匹配寻找，后者从父元素开始匹配寻找；2，前者逐级向上查找，直到发现匹配的元素后就停止了，后者一直向上查找直到根元素，然后把这些元素放进一个临时集合中，再用给定的选择器表达式去过滤；3，前者返回0或1个元素，后者可能包含0个，1个，或者多个元素。

closest对于处理事件委托非常有用。

### 参数

#### ****expr****String,ArrayV1.3

用以过滤元素的表达式。jQuery 1.4开始，也可以传递一个字符串数组，用于查找多个元素。

#### ****expr,[context]****StringV1.4

expr：用以过滤子元素的表达式

context：DOM元素在其中一个匹配的元素可以被发现。如果没有上下文在当时的情况下通过了jQuery设置将被使用。

#### ****jQuery object****objectV1.6

一个用于匹配元素的jQuery对象

#### ****element****DOMElementV1.6

一个用于匹配元素的DOM元素。

### 示例

#### 描述:

展示如何使用clostest查找多个元素

##### HTML 代码:

<ul><li></li><li></li></ul>

##### jQuery 代码:

$("li:first").closest(["ul", "body"]);

##### 结果:

[ul, body]

#### 描述:

展示如何使用clostest来完成事件委托。

##### HTML 代码:

<ul>

<li><b>Click me!</b></li>

<li>You can also <b>Click me!</b></li>

</ul>

##### jQuery 代码:

$(document).bind("click", function (e) {

$(e.target).closest("li").toggleClass("hilight");

});

## **find(expr|obj|ele)**

### 概述

搜索所有与指定表达式匹配的元素。这个函数是找出正在处理的元素的后代元素的好方法。

所有搜索都依靠jQuery表达式来完成。这个表达式可以使用CSS1-3的选择器语法来写。

### 参数

#### ****expr****StringV1.0

用于查找的表达式

#### ****jQuery object****objectV1.6

一个用于匹配元素的jQuery对象

#### ****element****DOMElementV1.6

一个DOM元素

### 示例

#### 描述:

从所有的段落开始，进一步搜索下面的span元素。与$("p span")相同。

##### HTML 代码:

<p><span>Hello</span>, how are you?</p>

##### jQuery 代码:

$("p").find("span")

##### 结果:

[ <span>Hello</span> ]

## **next([expr])**

### 概述

取得一个包含匹配的元素集合中每一个元素紧邻的后面同辈元素的元素集合。

这个函数只返回后面那个紧邻的同辈元素，而不是后面所有的同辈元素（可以使用nextAll）。可以用一个可选的表达式进行筛选。

### 参数

#### ****expr****StringV1.0

用于筛选的表达式

### 示例

#### 描述:

找到每个段落的后面紧邻的同辈元素。

##### HTML 代码:

<p>Hello</p><p>Hello Again</p><div><span>And Again</span></div>

##### jQuery 代码:

$("p").next()

##### 结果:

[ <p>Hello Again</p>, <div><span>And Again</span></div> ]

#### 描述:

找到每个段落的后面紧邻的同辈元素中类名为selected的元素。

##### HTML 代码:

<p>Hello</p><p class="selected">Hello Again</p><div><span>And Again</span></div>

##### jQuery 代码:

$("p").next(".selected")

##### 结果:

[ <p class="selected">Hello Again</p> ]

## **nextAll([expr])**

### 概述

查找当前元素之后所有的同辈元素。

可以用表达式过滤

### 参数

#### ****expr****StringV1.2

用来过滤的表达式

### 示例

#### 描述:

给第一个div之后的所有元素加个类

##### HTML 代码:

<div></div><div></div><div></div><div></div>

##### jQuery 代码:

$("div:first").nextAll().addClass("after");

##### 结果:

[ <div class="after"></div>, <div class="after"></div>, <div class="after"></div> ]

## **nextUntil([exp|ele][,fil])**

### 概述

查找当前元素之后所有的同辈元素，直到遇到匹配的那个元素为止。

如果提供的jQuery代表了一组DOM元素，.nextUntil()方法也能让我们找遍所有元素所在的DOM树，直到遇到了一个跟提供的参数匹配的元素的时候才会停下来。这个新jQuery对象里包含了下面所有找到的同辈元素，但不包括那个选择器匹配到的元素。

如果没有选择器匹配到，或者没有提供参数，那么跟在后面的所有同辈元素都会被选中。这就跟用没有提供参数的 .nextAll()效果一样。

### 参数

#### ****[expr][,filter]****String,StringV1.4

****expr****: 用于筛选祖先元素的表达式。

****filter****: 一个字符串，其中包含一个选择表达式匹配元素。

#### ****[element][,filter]****DOMElement,StringV1.6

****element****: 用于筛选祖先元素的DOM元素。

****filter****: 一个字符串，其中包含一个选择表达式匹配元素。

### 示例

#### 描述:

给#term-2后面直到dt前的元素加上红色背景

##### HTML 代码:

<dl>

<dt>term 1</dt>

<dd>definition 1-a</dd>

<dd>definition 1-b</dd>

<dd>definition 1-c</dd>

<dd>definition 1-d</dd>

<dt id="term-2">term 2</dt>

<dd>definition 2-a</dd>

<dd>definition 2-b</dd>

<dd>definition 2-c</dd>

<dt>term 3</dt>

<dd>definition 3-a</dd>

<dd>definition 3-b</dd>

</dl>

##### jQuery 代码:

$('#term-2').nextUntil('dt').css('background-color', 'red');

var term3 = document.getElementById("term-3");

$("#term-1").nextUntil(term3, "dd").css("color", "green");

## **offsetParent()**

### V1.2.6概述

返回第一个匹配元素用于定位的父节点。

这返回父元素中第一个其position设为relative或者absolute的元素。此方法仅对可见元素有效。

### 示例

#### 描述:

设置最近的祖先定位元素的背景颜色

##### HTML 代码:

<div style="width:70%;position:absolute;left:100px;top:100px">

<div style="margin:50px;background-color:yellow">

<p>点击下面的按钮可以设置本段落的最近的父（祖先）元素的背景色。</p>

<div>

</div>

<button>点击这里</button>

##### jQuery 代码:

$("button").click(function(){

$("p").offsetParent().css("background-color","red");

});

## **parent([expr])**

### 概述

取得一个包含着所有匹配元素的唯一父元素的元素集合。

你可以使用可选的表达式来筛选。

### 参数

#### ****expr****StringV1.0

用来筛选的表达式

### 示例

#### 描述:

查找每个段落的父元素

##### HTML 代码:

<div><p>Hello</p><p>Hello</p></div>

##### jQuery 代码:

$("p").parent()

##### 结果:

[ <div><p>Hello</p><p>Hello</p></div>]

#### 描述:

查找段落的父元素中每个类名为selected的父元素。

##### HTML 代码:

<div><p>Hello</p></div><div class="selected"><p>Hello Again</p></div>

##### jQuery 代码:

$("p").parent(".selected")

##### 结果:

[ <div class="selected"><p>Hello Again</p></div> ]

## **parents([expr])**

### 概述

取得一个包含着所有匹配元素的祖先元素的元素集合（不包含根元素）。可以通过一个可选的表达式进行筛选。

### 参数

#### ****expr****StringV1.0

用于筛选祖先元素的表达式

### 示例

#### 描述:

找到每个span元素的所有祖先元素。

##### HTML 代码:

<html><body><div><p><span>Hello</span></p><span>Hello Again</span></div></body></html>

##### jQuery 代码:

$("span").parents()

#### 描述:

找到每个span的所有是p元素的祖先元素。

##### jQuery 代码:

$("span").parents("p")

## **parentsUntil([exp|ele][,fil])**

### 概述

查找当前元素的所有的父辈元素，直到遇到匹配的那个元素为止。

如果提供的jQuery代表了一组DOM元素，.parentsUntil()方法也能让我们找遍所有元素的祖先元素，直到遇到了一个跟提供的参数匹配的元素的时候才会停下来。这个返回的jQuery对象里包含了下面所有找到的父辈元素，但不包括那个选择器匹配到的元素。

### 参数

#### ****[expr][,filter]****String,StringV1.4

****expr****: 用于筛选祖先元素的表达式

****filter****: 一个字符串，其中包含一个选择表达式匹配元素。

#### ****[element][,filter]****DOMElement,StringV1.6

****element****:用于筛选祖先元素的DOM元素

****filter****: 一个字符串，其中包含一个选择表达式匹配元素。

### 示例

#### 描述:

查找item-a的祖先，但不包括level-1

##### HTML 代码:

<ul class="level-1">

<li class="item-i">I</li>

<li class="item-ii">II

<ul class="level-2">

<li class="item-a">A</li>

<li class="item-b">B

<ul class="level-3">

<li class="item-1">1</li>

<li class="item-2">2</li>

<li class="item-3">3</li>

</ul>

</li>

<li class="item-c">C</li>

</ul>

</li>

<li class="item-iii">III</li>

</ul>

##### jQuery 代码:

$('li.item-a').parentsUntil('.level-1')

.css('background-color', 'red');

## **prev([expr])**

### 概述

取得一个包含匹配的元素集合中每一个元素紧邻的前一个同辈元素的元素集合。

可以用一个可选的表达式进行筛选。只有紧邻的同辈元素会被匹配到，而不是前面所有的同辈元素。

### 参数

#### ****expr****StringV1.0

用于筛选前一个同辈元素的表达式

### 示例

#### 描述:

找到每个段落紧邻的前一个同辈元素。

##### HTML 代码:

<p>Hello</p><div><span>Hello Again</span></div><p>And Again</p>

##### jQuery 代码:

$("p").prev()

##### 结果:

[ <div><span>Hello Again</span></div> ]

#### 描述:

找到每个段落紧邻的前一个同辈元素中类名为selected的元素。

##### HTML 代码:

<div><span>Hello</span></div><p class="selected">Hello Again</p><p>And Again</p>

##### jQuery 代码:

$("p").prev(".selected")

##### 结果:

[ <p class="selected">Hello Again</p> ]

## **prevAll([expr])**

### 概述

查找当前元素之前所有的同辈元素

可以用表达式过滤。

### 参数

#### ****expr****StringV1.2

用于过滤的表达式

### 示例

#### 描述:

给最后一个之前的所有div加上一个类

##### HTML 代码:

<div></div><div></div><div></div><div></div>

##### jQuery 代码:

$("div:last").prevAll().addClass("before");

##### 结果:

[ <div class="before"></div>, <div class="before"></div>, <div class="before"></div> ]

## **prevUntil([exp|ele][,fil])**

### 概述

查找当前元素之前所有的同辈元素，直到遇到匹配的那个元素为止。

如果提供的jQuery代表了一组DOM元素，.prevUntil()方法也能让我们找遍所有元素所在的DOM树，直到遇到了一个跟提供的参数匹配的元素的时候才会停下来。这个新jQuery对象里包含了前面所有找到的同辈元素，但不包括那个选择器匹配到的元素。

如果没有选择器匹配到，或者没有提供参数，那么排在前面的所有同辈元素都会被选中。这就跟用没有提供参数的 .prevAll()效果一样。

### 参数

#### ****[expr][,filter]****String,StringV1.4

****expr****: 用于筛选祖先元素的表达式

****filter****: 一个字符串，其中包含一个选择表达式匹配元素。

#### ****[element][,filter]****DOMElement,StringV1.6

****element****:用于筛选祖先元素的DOM元素

****filter****: 一个字符串，其中包含一个选择表达式匹配元素。

### 示例

#### 描述:

给#term-2前面直到dt前的元素加上红色背景

##### HTML 代码:

<dl>

<dt>term 1</dt>

<dd>definition 1-a</dd>

<dd>definition 1-b</dd>

<dd>definition 1-c</dd>

<dd>definition 1-d</dd>

<dt id="term-2">term 2</dt>

<dd>definition 2-a</dd>

<dd>definition 2-b</dd>

<dd>definition 2-c</dd>

<dt>term 3</dt>

<dd>definition 3-a</dd>

<dd>definition 3-b</dd>

</dl>

##### jQuery 代码:

$('#term-2').prevUntil('dt').css('background-color', 'red');

## **siblings([expr])**

### 概述

取得一个包含匹配的元素集合中每一个元素的所有唯一同辈元素的元素集合。可以用可选的表达式进行筛选。

### 参数

#### ****expr****StringV1.0

用于筛选同辈元素的表达式

### 示例

#### 描述:

找到每个div的所有同辈元素。

##### HTML 代码:

<p>Hello</p><div><span>Hello Again</span></div><p>And Again</p>

##### jQuery 代码:

$("div").siblings()

##### 结果:

[ <p>Hello</p>, <p>And Again</p> ]

#### 描述:

找到每个div的所有同辈元素中带有类名为selected的元素。

##### HTML 代码:

<div><span>Hello</span></div><p class="selected">Hello Again</p><p>And Again</p>

##### jQuery 代码:

$("div").siblings(".selected")

##### 结果:

[ <p class="selected">Hello Again</p> ]

## **add(expr|ele|html|obj[,con])**

### 概述

把与表达式匹配的元素添加到jQuery对象中。这个函数可以用于连接分别与两个表达式匹配的元素结果集。

jQuery 1.4 中， .add()方法返回的结果将始终以元素在HTML文档中出现的顺序来排序，而不再是简单的添加。

### 参数

#### ****expr****StringV1.0

一个用于匹配元素的选择器字符串。

#### ****elements****DOMElementV1.0

DOM元素。

#### ****html****StringV1.0

HTML片段添加到匹配的元素。

#### ****jQuery object****objectV1.3.2

一个jqeruy对象增加到匹配的元素

#### ****expr,context****Element, jQueryV1.4

****expr****:用于匹配元素并添加的表达式字符串，或者用于动态生成的HTML代码，如果是一个字符串数组则返回多个元素

****context****:作为待查找的 DOM 元素集、文档或 jQuery 对象。

### 示例

#### 描述:

添加一个新元素到一组匹配的元素中，并且这个新元素能匹配给定的表达式。

##### HTML 代码:

<p>Hello</p><span>Hello Again</span>

##### jQuery 代码:

$("p").add("span")

##### 结果:

[ <p>Hello</p>, <span>Hello Again</span> ]

#### 描述:

动态生成一个元素并添加至匹配的元素中

##### HTML 代码:

<p>Hello</p>

##### jQuery 代码:

$("p").add("<span>Again</span>")

##### 结果:

[ <p>Hello</p>, <span>Hello Again</span> ]

#### 描述:

为匹配的元素添加一个或者多个元素

##### HTML 代码:

<p>Hello</p><p><span id="a">Hello Again</span></p>

##### jQuery 代码:

$("p").add(document.getElementById("a"))

##### 结果:

[ <p>Hello</p>, <p><span id="a">Hello Again</span></p>, <span id="a">Hello Again</span> ]

## **andSelf()**

### V1.2概述

加入先前所选的加入当前元素中

对于筛选或查找后的元素，要加入先前所选元素时将会很有用。

### 示例

#### 描述:

选取所有div以及内部的p，并加上border类

##### HTML 代码:

<div><p>First Paragraph</p><p>Second Paragraph</p></div>

##### jQuery 代码:

$("div").find("p").andSelf().addClass("border");

##### 结果:

<div class="border">

<p class="border">First Paragraph</p>

<p class="border">Second Paragraph</p>

</div>

## **contents()**

### V1.2概述

查找匹配元素内部所有的子节点（包括文本节点）。如果元素是一个iframe，则查找文档内容

### 示例

#### 描述:

查找所有文本节点并加粗

##### HTML 代码:

<p>Hello <a href="http://ejohn.org/">John</a>, how are you doing?</p>

##### jQuery 代码:

$("p").contents().not("[nodeType=1]").wrap("<b/>");

##### 结果:

<p><b>Hello</b> <a href="http://ejohn.org/">John</a>, <b>how are you doing?</b></p>

#### 描述:

往一个空框架中加些内容

##### HTML 代码:

<iframe src="/index-blank.html" width="300" height="100"></iframe>

##### jQuery 代码:

$("iframe").contents().find("body")

.append("I'm in an iframe!");

## **end()**

### V1.0概述

回到最近的一个"破坏性"操作之前。即，将匹配的元素列表变为前一次的状态。

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。这包括在 Traversing 中任何返回一个jQuery对象的函数--'add', 'andSelf', 'children', 'filter', 'find', 'map', 'next', 'nextAll', 'not', 'parent', 'parents', 'prev', 'prevAll', 'siblings' and 'slice'--再加上 Manipulation 中的 'clone'。

### 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

##### HTML 代码:

<p><span>Hello</span>,how are you?</p>

##### jQuery 代码:

$("p").find("span").end()

##### 结果:

[ <p><span>Hello</span> how are you?</p> ]