

第4章 过滤选择器

学习要点：

- 1.基本过滤器
- 2.内容过滤器
- 3.可见性过滤器
- 4.子元素过滤器
- 5.其他方法

主讲教师：李炎恢

官方网站：<http://www.ycku.com>

合作网站：<http://www.ibeifeng.com>

过滤选择器简称：过滤器。它其实也是一种选择器，而这种选择器类似与 CSS3 (<http://t.mb5u.com/css3/>)里的伪类，可以让不支持 CSS3 的低版本浏览器也能支持。和常规选择器一样，jQuery 为了方便开发者使用，提供了很多独有的过滤器。

一. 基本过滤器

过滤器主要通过特定的过滤规则来筛选所需的 DOM 元素，和 CSS 中的伪类的语法类似：使用冒号(:)开头。

过滤器名	jQuery 语法	说明	返回
:first	\$(li:first)	选取第一个元素	单个元素
:last	\$(li:last)	选取最后一个元素	单个元素
:not(selector)	\$(li:not(.red))	选取 class 不是 red 的 li 元素	集合元素
:even	\$(li:even)	选择索引(0 开始)是偶数的所有元素	集合元素
:odd	\$(li:odd)	选择索引(0 开始)是奇数的所有元素	集合元素
:eq(index)	\$(li:eq(2))	选择索引(0 开始)等于 index 的元素	单个元素
:gt(index)	\$(li:gt(2))	选择索引(0 开始)大于 index 的元素	集合元素
:lt(index)	\$(li:lt(2))	选择索引(0 开始)小于 index 的元素	集合元素
:header	\$(header)	选择标题元素，h1 ~ h6	集合元素
:animated	\$(animated)	选择正在执行动画的元素	集合元素
:focus	\$(focus)	选择当前被焦点的元素	集合元素

<code>\$(li:first).css('background', '#ccc');</code>	//第一个元素
<code>\$(li:last).css('background', '#ccc');</code>	//最后一个元素
<code>\$(li:not(.red)).css('background', '#ccc');</code>	//非 class 为 red 的元素
<code>\$(li:even).css('background', '#ccc');</code>	//索引为偶数的元素

```
$(li:odd).css('background', '#ccc'); //索引为奇数的元素
$(li:eq(2)).css('background', '#ccc'); //指定索引值的元素
$(li:gt(2)).css('background', '#ccc'); //大于索引值的元素
$(li:lt(2)).css('background', '#ccc'); //小于索引值的元素
$(':header').css('background', '#ccc'); //页面所有 h1 ~ h6 元素
```

注意：:focus 过滤器，必须是网页初始状态的已经被激活焦点的元素才能实现元素获取。而不是鼠标点击或者 Tab 键盘敲击激活的。

```
$(input).get(0).focus(); //先初始化激活一个元素焦点
$(':focus').css('background', 'red'); //被焦点的元素
```

jQuery 为最常用的过滤器提供了专用的方法，已达到提到性能和效率的作用：

```
$(li).eq(2).css('background', '#ccc'); //元素 li 的第三个元素，负数从后开始
$(li).first().css('background', '#ccc'); //元素 li 的第一个元素
$(li).last().css('background', '#ccc'); //元素 li 的最后一个元素
$(li).not('.red').css('background', '#ccc'); //元素 li 不含 class 为 red 的元素
```

注意：:first、:last 和 first()、last()这两组过滤器和方法在出现相同元素的时候，first 会实现第一个父元素的第一个子元素，last 会实现最后一个父元素的最后一个子元素。所以，如果需要明确是哪个父元素，需要指明：

```
$('#box li:last').css('background', '#ccc'); //box 元素的最后一个 li
//或
$('#box li).last().css('background', '#ccc'); //同上
```

二. 内容过滤器

内容过滤器的过滤规则主要是包含的子元素或文本内容上。

过滤器名	jQuery 语法	说明	返回
:contains(text)	\$(':contains("ycku.com"))	选取含有"ycku.com"文本的元素	元素集合
:empty	\$(':empty')	选取不包含子元素或空文本的元素	元素集合
:has(selector)	\$(':has(.red)')	选取含有 class 是 red 的元素	元素集合
:parent	\$(':parent')	选取含有子元素或文本的元素	元素集合

```
//选择元素文本节点含有 ycku.com 文本的元素
$(div:contains("ycku.com")).css('background', '#ccc');
$(div:empty).css('background', '#ccc'); //选择空元素
$(ul:has(.red)).css('background', '#ccc'); //选择子元素含有 class 是 red 的元素
$(parent).css('background', '#ccc'); //选择非空元素
```

jQuery 提供了一个 has()方法来提高:has 过滤器的性能：

```
$(ul).has('.red').css('background', '#ccc'); //选择子元素含有 class 是 red 的元素
```

jQuery 提供了一个名称和:parent 相似的方法,但这个方法并不是选取含有子元素或文本的元素,而是获取当前元素的父元素,返回的是元素集合。

```
$(li).parent().css('background', '#ccc');           //选择当前元素的父元素
$(li).parents().css('background', '#ccc');           //选择当前元素的父元素及祖先元素
$(li).parentsUntil('div').css('background', '#ccc'); //选择当前元素遇到 div 父元素停止
```

三. 可见性过滤器

可见性过滤器根据元素的可见性和不可见性来选择相应的元素。

过滤器名	jQuery 语法	说明	返回
:hidden	\$(':hidden')	选取所有不可见元素	集合元素
:visible	\$(':visible')	选取所有可见元素	集合元素

```
$(p:hidden).size();           //元素 p 隐藏的元素
$(p:visible).size();          //元素 p 显示的元素
```

注意: :hidden 过滤器一般是包含的内容为: CSS 样式为 display:none、input 表单类型为 type="hidden"和 visibility:hidden 的元素。

四. 子元素过滤器

子元素过滤器的过滤规则是通过父元素和子元素的关系来获取相应的元素。

过滤器名	jQuery 语法	说明	返回
:first-child	\$(li:first-child)	获取每个父元素的第一个子元素	集合元素
:last-child	\$(li:last-child)	获取每个父元素的最后一个子元素	集合元素
:only-child	\$(li:only-child)	获取只有一个子元素的元素	集合元素
:nth-child(odd/even/eq(index))	\$(li:nth-child(even))	获取每个自定义子元素的元素(索引值从 1 开始计算)	集合元素

```
$(li:first-child).css('background', '#ccc');           //每个父元素第一个 li 元素
$(li:last-child).css('background', '#ccc');           //每个父元素最后一个 li 元素
$(li:only-child).css('background', '#ccc');           //每个父元素只有一个 li 元素
$(li:nth-child(odd)).css('background', '#ccc');       //每个父元素奇数 li 元素
$(li:nth-child(even)).css('background', '#ccc');       //每个父元素偶数 li 元素
$(li:nth-child(2)).css('background', '#ccc');         //每个父元素第三个 li 元素
```

五. 其他方法

jQuery 在选择器和过滤器上,还提供了一些常用的方法,方便我们开发时灵活使用。

方法名	jQuery 语法	说明	返回
is(s/o/e/f)	\$(li).is('.red')	传递选择器、DOM、jquery 对象	集合元素

		或是函数来匹配元素结合	
hasClass(class)	<code>\$(li).eq(2).hasClass('red')</code>	其实就是 <code>is("." + class)</code>	集合元素
slice(start, end)	<code>\$(li).slice(0,2)</code>	选择从 start 到 end 位置的元素，如果是负数，则从后开始	集合元素
filter(s/o/e/f)	<code>\$(li).filter('red')</code>		
end()	<code>\$(div).find('p').end()</code>	获取当前元素前一次状态	集合元素
contents()	<code>\$(div).contents()</code>	获取某元素下面所有元素节点，包括文本节点，如果是 iframe，则可以查找文本内容	集合元素

```

$.red').is('li'); //true, 选择器, 检测 class 为是否为 red
$.red').is($('li')); //true, jQuery 对象集合, 同上
$.red').is($('li').eq(2)); //true, jQuery 对象单个, 同上
$.red').is($('li').get(2)); //true, DOM 对象, 同上
$.red').is(function () { //true, 方法, 同上
    return $(this).attr('title') == '列表 3'; //可以自定义各种判断
});

```

```

$(li).eq(2).hasClass('red'); //和 is 一样, 只不过只能传递 class

```

```

$(li).slice(0,2).css('color', 'red'); //前三个变成红色

```

注意：这个参数有多种传法和 JavaScript 的 slice 方法是一样的比如：slice(2)，从第三个开始到最后选定；slice(2,4)，第三和第四被选定；slice(0,-2)，从倒数第三个位置，向前选定所有；slice(2,-2)，前两个和末尾两个未选定。

```

$("div").find("p").end().get(0); //返回 div 的原生 DOM
$(div).contents().size(); //返回子节点(包括文本)数量

```

```

$(li).filter('red').css('background','#ccc'); //选择 li 的 class 为 red 的元素
$(li).filter('red, :first, :last').css('background','#ccc'); //增加了首尾选择
//特殊要求函数返回
$(li).filter(function () {
    return $(this).attr('class') == 'red' && $(this).attr('title') == '列表 3';
}).css('background', '#ccc');

```

感谢收看本次教程！

本课程是由北风网(ibEIFENG.COM)

瓢城 Web 俱乐部(Ycku.com)联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！