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

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为了更方便开发者使用，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 |

$('li:first').css('background', '#ccc');//第一个元素

$('li:last).css('background', '#ccc');//最后一个元素

$('li:not(.red)).css('background', '#ccc');//非 class为 red的元素

$('li:even').css('background', '#ccc');//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

Query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的父元素（作用于ul,不是li） | 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。

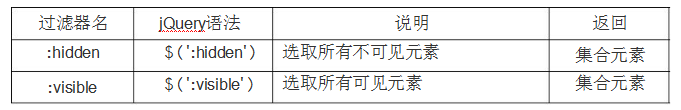
$('li').parent().css('background', '#ccc');//选择当前元素的父元素

$('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。



$('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。

面试题：visible与地

请说出文本框readonly属性与disabled属性的区别。

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(in  dex)) | $('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(3)').css('background', '#ccc'); //每个父元素第三个 li元素

作业：自己查询以下方法，并且注以相应的案例：

is(s/o/e/f)

hasClass(class)

slice(start, end)

filter(s/o/e/f)

end()

contents()

