**例：**通过id，标签，class获取元素

<div id="div1" class="box"></div>

原生选择元素：

document.getElementById('div1');

document.getElementsByTagName('div');

document.getElementsByClassName('box');// 还不兼容，要自己实现兼容的写法

jq选择元素

注意它有三种方式：模拟CSS选择元素、独有表达式选择、多种筛选方法

**一、模拟CSS选择元素**

<div id="div1" class="box"></div>

$('#div1');

$('div');

$('.box');

特别注意：

这时，如果页面中还有一个<p class="box"></p>，还是使用$('.box').css('background', '#f00');

这时，会发现div和p都加上背景色了。

会发现这跟原生有一个很大的不同，就是省略了循环。

原生中，我们选择了一组原生，是需要for循环来对每一个进行操作的。但这里，我们不用循环就可以。

**二、独有表达式选择**

<ul>

<li></li>

<li></li>

<li></li>

<li></li>

<li></li>

</ul>

$('li').css('background', '#f00');

$('li:first').css('background', '#f00');

$('li:last').css('background', '#f00');

$('li:eq(2)').css('background', '#f00');

$('li:even').css('background', '#f00');

$('li:odd').css('background', '#f00');

eq(index) 匹配列表中的第index个元素（index 从 0 开始）

gt(index) 匹配列表索引 大于 index 的元素

lt(index) 匹配列表索引 小于 index 的元素

选取范围

$("ul li:lt(5):gt(2)") 注意：这里:lt(5):gt(2) 的顺序不能变，必须是lt在前，gt在后。

$('ul li:even') 它返回的是奇数（相反）

$('ul li:nth-child(even)') 它返回的是偶数（恢复本来面目）

**三、多种筛选方法**

<ul>

<li></li>

<li></li>

<li title="hello"></li>

<li class="box"></li>

<li class="box"></li>

</ul>

$('li.box').css('background', '#f00');

$('li').filter('.box').css('background', '#f00');

$('li').filter('[title=hello]').css('background', '#f00');