jQuery获取父、子、同级标签

[**jquery选择器 之 获取父级元素、同级元素、子元素**](http://www.cnblogs.com/Tally/archive/2013/01/04/2844042.html)

**一、获取父级元素**

1、 **parent([expr]):**

**获取指定元素的所有父级元素**

<div id="par\_div"><a id="href\_fir" href="#">href\_fir</a>  
<a id="href\_sec" href="#">href\_sec</a>  
<a id="href\_thr" href="#">href\_thr</a></div>  
<span id="par\_span">  
<a id="href\_fiv" href="#">href\_fiv</a>  
</span>

$(document).ready(function(){  
$("a").parent().addClass('a\_par');  
});

**二、获取同级元素：**

**1、next([expr])：**

**获取指定元素的下一个同级元素（注意是下一个同级元素哦）**

<!DOCTYPE html>  
<html>  
<head>  
  <script type="text/javascript" src="/jquery/jquery.js"></script>  
</head>

<body>  
<ul>  
   <li>list item 1</li>  
   <li>list item 2</li>  
   <li class="third-item">list item 3</li>  
   <li>list item 4</li>  
   <li>list item 5</li>  
</ul>

<script>  
$('li.third-item').next().css('background-color', 'red');  
</script>

</body>  
</html>

这个例子的结果是，只有list item 4背景色变为红色

**2、nextAll([expr]):**

**获取指定元素后边的所有同级元素**

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

var p\_nex = $("p").nextAll();  
p\_nex.addClass('p\_next\_all');

注意看最后一个”<p&gt”标签哦，也被加上了’p\_next\_all’这个类名哦~~

**3、andSelf():**

**获取指定元素后边的所有同级元素，之后加上指定的元素**

我感觉这个函数是最有意思的一个函数了，什么意思？直译过来就是”还有我“,”还有自己”,没错，还有自己。

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

var p\_nex = $("p").nextAll().andSelf();  
p\_nex.addClass('p\_next\_all');

注意看第一个“<p>”标签啊，这句话的意思是选择p标签后面的所有同级标签，以及自己。。。

**以下这两个不举具体的例子了，实际上就是next()和nextAll()的相反喽**

**4、prev()：获取指定元素的上一个同级元素（是上一个哦）。**

**5、prevAll()：获取指定元素的前边所有的同级元素。**

**三、获取子元素**

**1、查找子元素方式1：>**

例如：var aNods = $("ul > a");查找ul下的所有a标签

**2、查找子元素方式2：children()**

**3、查找子元素方式3：find()**

**这里再简单介绍以下children()和find()的异同：**

1> children及find方法都用是用来获得element的子elements的，两者都不会返回 text node，就像大多数的jQuery方法一样。   
2> children方法获得的仅仅是元素一下级的子元素，即：immediate children。   
3> find方法获得所有下级元素，即：descendants of these elements in the DOM tree   
4> children方法的参数selector 是可选的（optionally），用来过滤子元素，

但find方法的参数selector方法是必选的。   
5> find方法事实上可以通过使用 jQuery( selector, context )来实现。即$('li.item-ii').find('li')等同于$('li', 'li.item-ii').

例：

<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>  
使用：$('ul.level-2').children().css('border', '1px solid green'); 的效果是：

使用 $('ul.level-2').find('li').css('border', '1px solid green'); 的效果是： 