# JQuery基础教程 简明笔记

# 第二章 选择符

CSS文件：

**.poem {**

**margin: 0.2em;**

**}**

**.emphasized {**

**font-style:italic;**

**border:1px solid #888;**

**padding: 0.5em;**

**}**

**jQuery:**

**$(document).ready(function () {**

**$('.poem').addClass('emphasized');**

**});**

说明：用匿名函数的方式把所有文档中带“.poem-stanza”类的文本添加“emphasized”样式。当DOM加载完成后（不必等待图像加载完成）触发。

**$(‘p’):** 取得文档中所有的段落。

**$(‘#some-id’) :** 会取得文档中具有对应some-id ID的一个元素。

**$(‘.some-class’):** 会取得文档中带有some-class类中的所有元素。

**$(document).ready(function () {**

**$('#selected-plays > li').addClass('horizontal');**

**$('#selected-plays li:not(.horizontal)').addClass('horizontal');**

**});**

说明：改代码使用了子元素组合符（>）将horizontal类只添加到位于顶级的项中。实际上，位于$()函数中的选择符的含义是，查找ID 为selected-plays的元素（#selected-plays）的子元素（>）中的所有的列表项（li）。下面那行是为其他项（非顶级的项）添加样式。

**$('a[@title]') :**选择所有带title属性的链接。jQuery通过将属性前置一个@符号并放在一对方括号中。如果不带前置@符号的情况下，可以用来指定包含在另一个元素中的元素。

**$('div[ol]'):** 取得包含一个ol元素的所有div元素。注意，属性选择符方括号中的字符串仅作为筛选标准存在，用来辅助选择目标对象，方括号左侧的元素才是属性选择符真正要选择的目标对象。

属性选择符允许以类似正则表达式的语法来标识字符串的开始（^），和结尾（$）。而且也可以使用星号（\*）表示字符串中的任意位置。

**$(document).ready(function() {**

**$(‘a[@href^=”mailto:”]’).addClass(‘mailto’);** //取得所有带href属性并以.pdf结尾的链接。

**$(‘a[@href$=”.pdf”]’).addClass(‘pdfline’);**

**$(‘a[@href\*=”mysite.com”]’).addClass(‘mysite’);**

**});**

自定义选择符（选择符以一个冒号（：）开头）：

**$(‘div.horizontal:eq(1)’):** 从匹配的带哟有horizontal类得div集合中，选择第2个项（注意，JavaScript数组采用从0开始的编号方式。）。

jQuery库中有两个十分有用的自定义选择符是：odd和even。

**.odd {**

**Background-color:#ffc;**  /\*奇数行的背景颜色为浅黄色\*/

**}**

**.even {**

**Background-color:#cef;**  /\*偶数行的背景颜色为浅蓝色\*/

**}**

jQuery代码：

**$(document).ready(function () {**

**$(‘tr:odd’).addClass(‘odd’);**

**$(‘tr:even’).addClass(‘even’);**

**$(‘td:contains(“Henry”)’).addClass(‘highlight’);** /\*我们还希望突出显示提到任何一种Henry游戏的所有表格单元\*/

}

**);**

为标题行添加样式

**$(document.ready(function() {**

**$(‘th’).parent().addClass(‘table-heading’);**

**$(‘tr:not([th]):even’).addClass(‘even’);**

**$(‘td:contains(“Henry”)’).addClass(‘highlight’);**

**});**

为标题行添加不同的样式的任务，可以通过找到<th>标签并取得它们的父元素来完成。而其他的行则可以通过组合使用CSS、Xpath以及自定义选择符过滤<tr>元素来选择并添加样式。

对于标题行，我们通过parent()方法取得了一个普通的父元素，没有在圆括号中包含任何参数。因为我们知道这个父亲元素就是<tr>.而且结果集合中只有这一个元素。虽然我们会认为由于存在两个<th>元素，所以<tr>中最终会包含两个table-heading类，但jQuery很聪明，它不会向元素中重复添加同一个类名。其中，.next()方法只取得最接近的下一个同辈元素。

1. **$(‘td:contains(“Henry”)’).siblings().addClass(‘highlight’);**

取得包含Henry的单元格，然后取得改单元格的所有同辈元素（不仅仅是下一个同辈元素），然后添加类。

1. **$(‘td:contains(“Henry”)’).parent().find(‘td:gt(0)’).addClass(‘highlight’);**

取得包含Henry的单元格，再取得它的父元素，然后找到该元素中包含的所有（0是第1个单元格）的单元格。最后添加类。

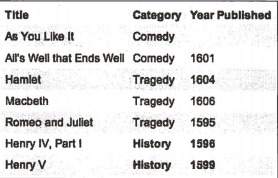
1. **$(‘td:contains(“Henry”)’).parent().find(‘td’).not(‘:contains(“Henry”)’)).addClass(‘highlight’);**

取得包含Henry的单元格，再取得它的父元素，找到该元素中包含的所有单元格，然后过滤这些单元格排除包含Henry的那一个。最后添加类。

1. **$(‘td:contains(“Henry”)’).parent().find(‘td:eq(1)’).addClass(‘highlight’).end().find(‘td:eq(2)’).addClass(‘highlight’);**

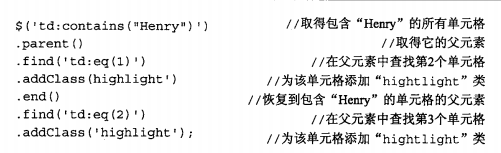
取得包含Henry的单元格，再取得它的父元素，找到该元素包含的子元素中的第2个单元格，添加类；取消最后一次**.find()**，再查找该元素包含的子元素中的第3个单元格，添加类。

以上四种方法都会得到相同的结果，如图：

****

为了代码清晰起见，以上组合选择符表达式和方法的这几种方式并不全都是我们所推荐的，实际上，第4种转弯抹角的方式几乎达到了荒谬的程度。不过，通过这些方式应该能够体现出jQuery的DOM便利方式的确极具灵活性。

上面的第4中方法可以写成多行：



**var myTag = $(‘#my-element’).get(0).tagName;**

等价于

**var myTag = $(‘#my-element’)[0].tagName;**

获得带有id=”my-element”属性的元素的标签名。

# 第三章 事件

**$(document).ready(function() {**

//这里是代码。。。。

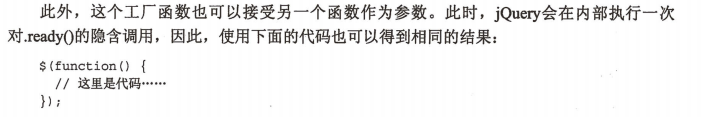
**});**

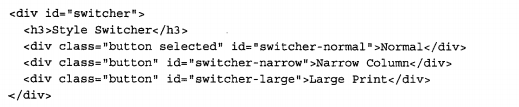
可以简写成：

**$().ready(function() {**

//这里是代码。。。。

**});**

****

****

我们希望用户点击Large Print按钮时能改变正文的显示样式，这需要一点CSS代码：

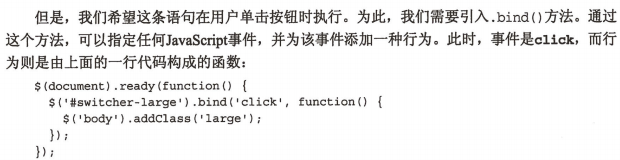
**body.large .chapter {**

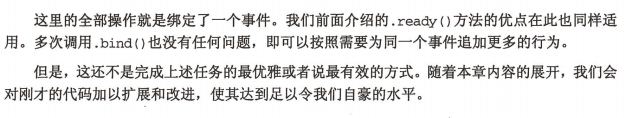
**font-size: 1.5em;**

**}**

然后我们的目标就是为body标签应用large类。添加类得语句为：

**$(‘body’).addClass(‘large’);**

****

****我们希望把用户最近单击过的按钮的文本添加粗体样式：

**.selected {**

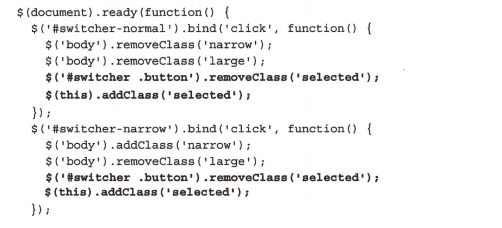
**Font-weight: bold;**

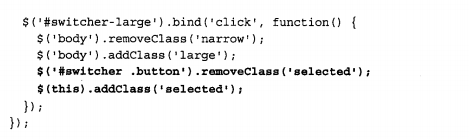
**}**

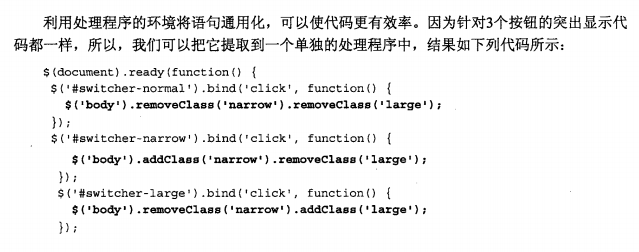
利用jQuery的隐式迭代特性，我们知道

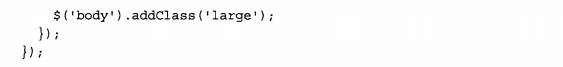
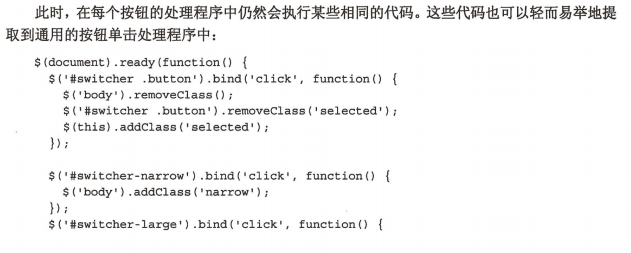
$(‘#switcher .button’).removeClass(‘selected’); 可以把从所有按钮中移除这个类。

所以我们编写如下代码：





****

****