**DOM操作**

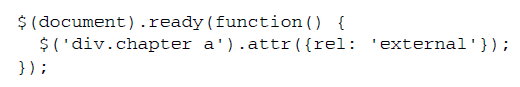
* 操作属性
* 创建新元素
* 插入新元素
* 移动元素
* 包装元素
* 反向插入新元素
* 复制元素

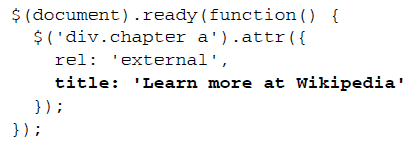
**操作属性**

* 操作class属性
* 追加样式: addClass()
* 移除样式: removeClass() --- 从匹配的元素中删除全部或指定的 class
* 切换样式: toggleClass() --- 控制样式上的重复切换.如果类名存在则删除它, 如果类名不存在则添加它.
* 判断是否含有某个样式: hasClass() --- 判断元素中是否含有某个 class, 如果有, 则返回 true; 否则返回 false

**操作其他属性（非类属性）**

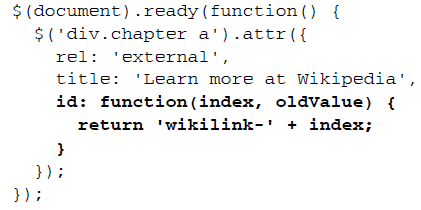
* attr(): 获取html属性和设置属性
* **当为该方法传递一个参数时, 即为某元素的获取指定属性**
* **当为该方法传递两个参数时, 即为某元素设置指定属性的值**
* removeAttr():删除属性值
* **$(‘div.chapter a’).attr('rel'， 'external');**
* attr（）可以传入一个包含键值对的映射
* 例题5-1



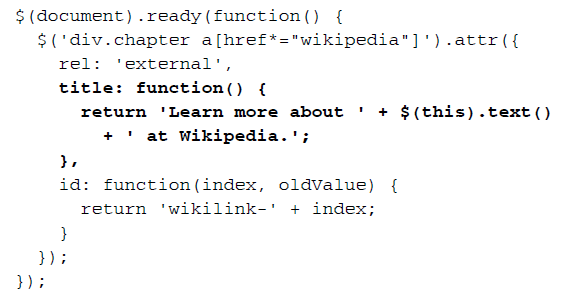
* 例题5-2

**值回调**

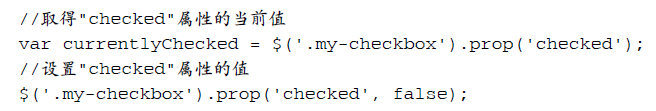
* 值回调其实就是给参数传入一个函数，而不是传入具体的值。这个函数会针对匹配的元素集
* 中的每个元素都调用一次，调用后的返回值将作为属性的值
* 例题5-3 为每个元素生成唯一id



* 例题5-5包含wikipedia的链接，添加更具体的title属性

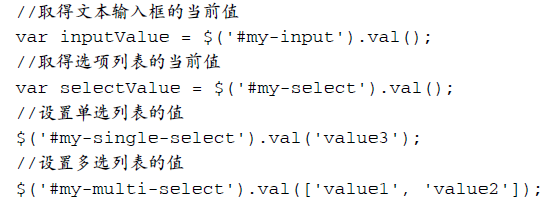


* 使用prop（）方法可以获取和设置DOM元素的属性。
* 大部分情况下，HTML属性与DOM属性名称和值是一样的，但个别元素略有区别。，比如HTML中的checked属性是一个字符串，而DOM中的checked属性则是一个布尔值。对于布尔值属性



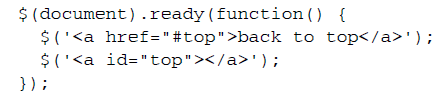
**表单控件的值**

* 在取得和设置表单控件的值时，最好不要使用.attr()方法。而对于选项列表，最好连.prop()方法也不要使用，建议使用jQuery提供的.val()方法



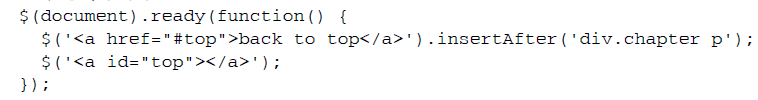
**创建新元素**

* 使用$()可以生成一个jQuery对象。
* 例5-6添加回到顶端的链接

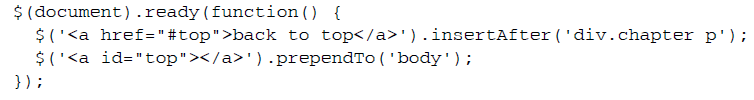


**插入新元素**

* 用insertAfter()方法在元素外部，之后添加内容。
* 用insertBefore()方法在元素外部，之前添加内容。
* 例题5-7

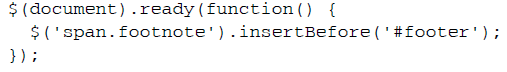


* 用appendTo()方法在元素内部，之后添加内容。
* 用prependTo()方法在元素内部，之前添加内容。
* 例题5-8



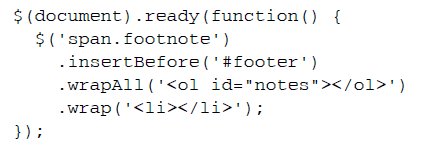
**移动元素**

* 我们可以用$()取得页面中某个位置上的元素，将他们插入到另一个位置
* 例5-9将文章中的注脚移动到文档的底部

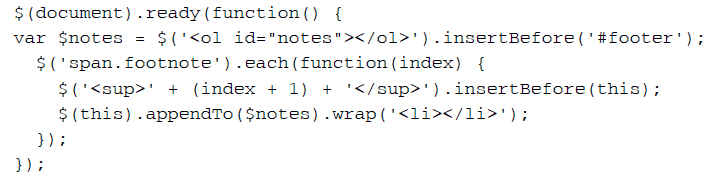


**包装元素**

* wrap() 方法把每个被选元素放置在指定的 HTML 内容或元素中。
* wrapAll() 在指定的 HTML 内容或元素中放置所有被选的元素
* 例5-10添加编号

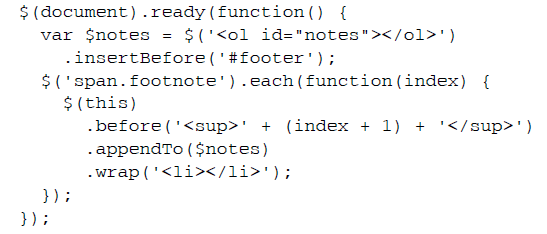


* 使用foreach遍历
* 5-12提取编号，添加到对应链接



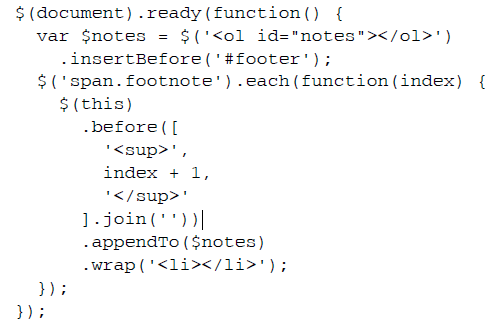
**反向插入方法**

* 像.insertBefore()和.appendTo()这样的插入方法，一般都 有一个对应的反向方法。反向方法也执行相同的操作，只不过“目标”和“内容”正好相反。
* 例
* $(“<p>Hello</p>”). appendTo(“#container”);
* $(“#container”). append(“<p>Hello</p>”);
* .before()是.insertBefore()对应的反向方法
* 例5-13



* 使用join创建字符串
* var arr = [ "a", "b", "c", "d", "e" ]
* arr.join("-");
* =============================
* 结果:a-b-c-d-e

**例4-14使用join创建字符串**



* 例4-15为脚注标签添加一个指向页面底部的链接和一个唯一的id值。

原来的上标

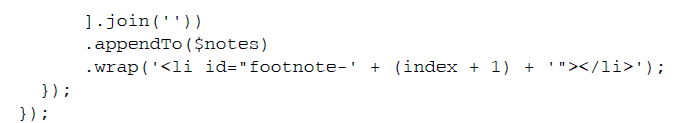
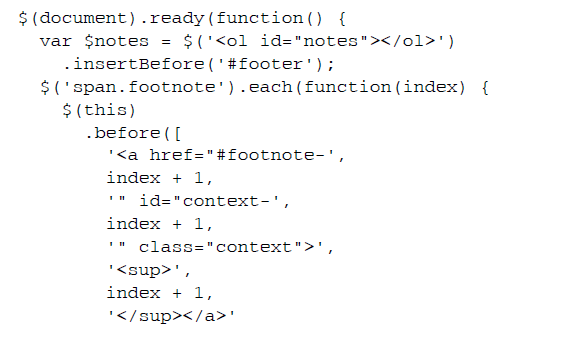
**<sup>2</sup>**

更改后的上标

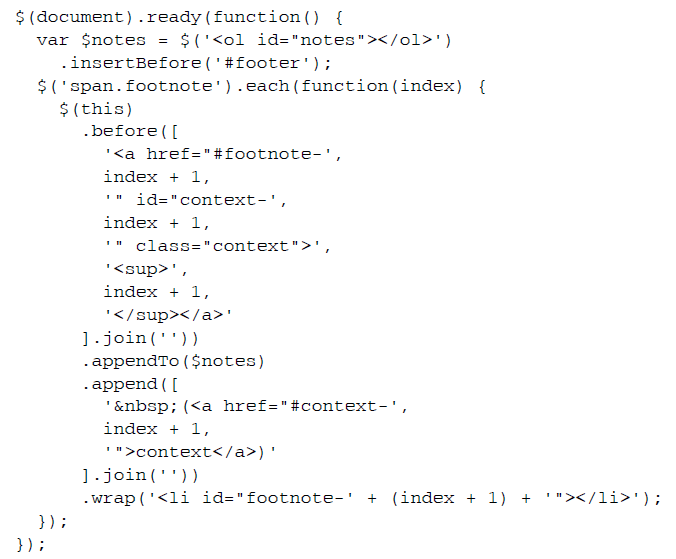
**<a href="#footnote-2" id="context-2“ class="context“>**

**<sup>2</sup>**

**</a>**



* 例4-16脚注标签中创建一个指向其上下文的链接

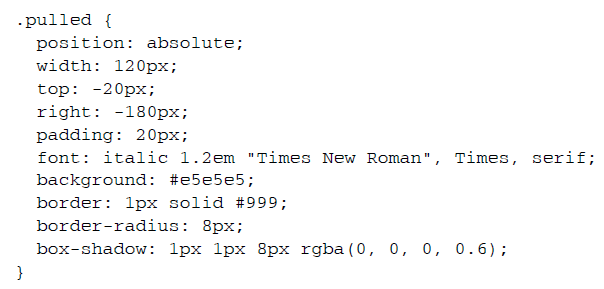


**复制元素**

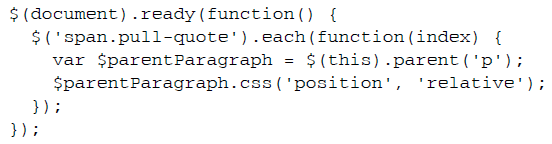
* clone() 方法生成被选元素的副本，包含子节点、文本和属性

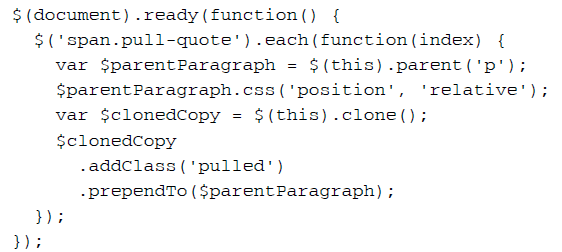


* 例如，下面这行代码将创建<div class="chapter">中第一段落的副本：
* **$('div.chapter p:eq(0)').clone();**
* 通过复制创建突出引用



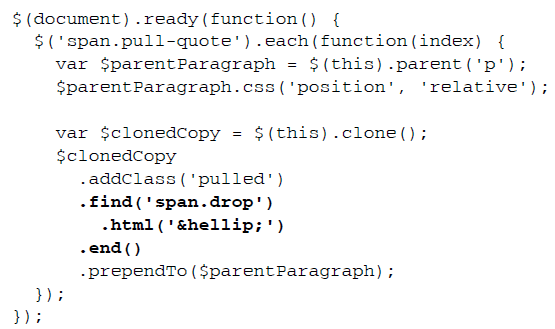
* 代码清单5-17，匹配所有<span class=
* "pull-quote">元素的选择符表达式，然后为选择的元素应用position:relative样式，



* 代码清单5-18，先复制每个<span>元素，然后为得到的副本添加pulled类，最后再把这个副本插入到其父段落的开始处，

**内容setter 和getter 方法**

* 如果能够对突出引用稍作修改，去掉一些文本并代之以省略号，那么效果会更好。
* 实现这种替换的最简便方式，就是直接用新的HTML代替旧的内容。此时，就要用到.html()方法了，参见代码清单5-19。



* 与.html()方法类似，.text()也可以取得匹配元素的内容，或者用新字符串替换匹配元素的内容。但是，与.html()不同的是，.text()始终会取得或设置纯文本内容。
* 在使用.text()取得内容时，所有HTML标签都将被忽略，而所有HTML实体也会被转换成对应的字符。而在通过它设置内容时，诸如<这样的特殊字符，都会被转换成等价的HTML实体，
* 代码清单5-20，在引用中显示的是去掉了<span class="pull-qoute">中的<strong>、<em>、<a href>及其他行内标签之后的文本。

