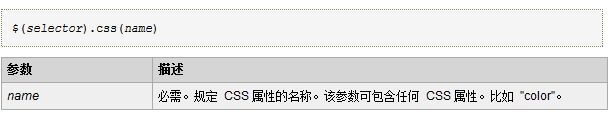
**样式与动画**

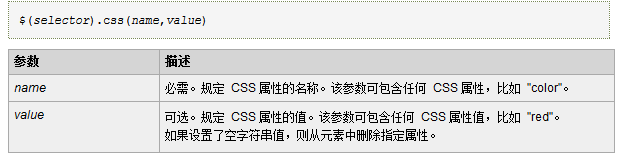
* 修改内联CSS
* 基本的隐藏和显示
* 指定显示速度
* 淡入和淡出
* 滑上和滑下
* 创建自定义动画

**修改内联CSS**

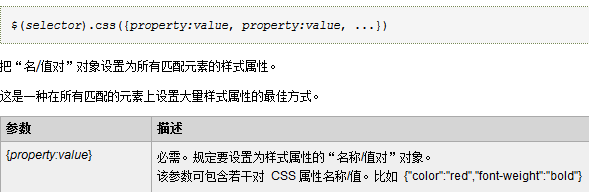
* 为了修改文档的外观，我们都是先在单独的样式表中为类定义好样式，然后再通过jQuery来添加或者移除这些类。一般而言，这都 是为HTML应用CSS的首先方式。但是，在有些情况下，使用的样式没有在样式表中 定义，或者能过样式表声明不是那么容易。针对这种情况，jQuery提供了.css()方法。
* css() 方法返回或设置匹配的元素的一个或多个样式属性。
* **返回 CSS 属性值**

****

* **实例**
* 取得第一个段落的 color 样式属性的值：
* $("p").css("color");
* **设置 CSS 属性**

****

* **实例**
* 将所有段落的颜色设为红色：
* $("p").css("color","red");



$("p").css({

"color":"white",

"background-color":"#98bf21",

"font-family":"Arial",

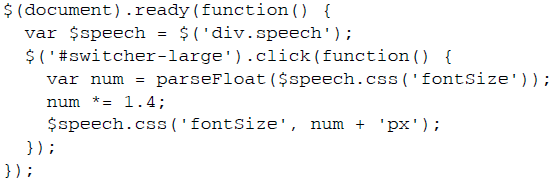
"font-size":"20px",

"padding":"5px"

});

**例**

* 例4-3——例4-5使用css()改变字体大小。



* 例题4.4

$(document).ready(function() {

var $speech = $('div.speech');

$('#switcher button').click(function() {

var num = parseFloat($speech.css('fontSize'));

if (this.id== 'switcher-large') {

num \*= 1.4;

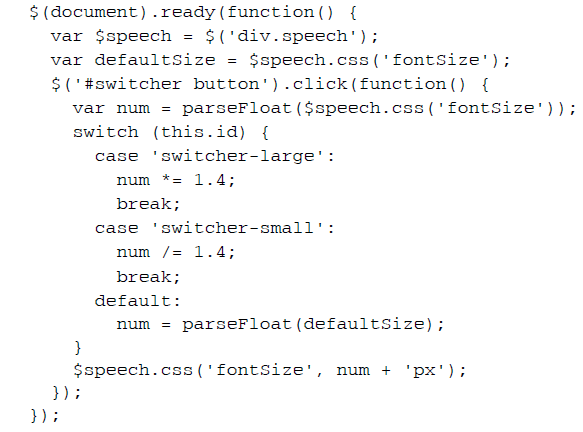
} else if (this.id== 'switcher-small') {

num /= 1.4;

}

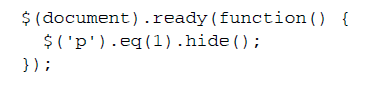
$speech.css('fontSize', num + 'px');

});

});

**基本的隐藏和显示**

* hide(): 在 HTML 文档中, 为一个元素调用 hide() 方法会将该元素的 display 样式改为 none. 代码功能同 css(“display”, “none”);
* show(): 将元素的 display 样式改为先前的显示状态.
* 以上两个方法在不带任何参数的情况下, 作用是立即隐藏或显示匹配的元素, 不会有任何动画. 可以通过制定速度参数使元素动起来.
* 以上两个方法会同时减少(增大)内容的高度, 宽度和不透明度.

**例4-6**

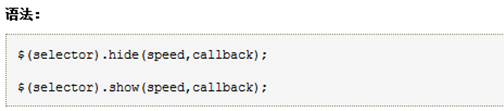
* 为一段文本添加一个read more链接，并隐藏第二段文本

**代码清单4-7**

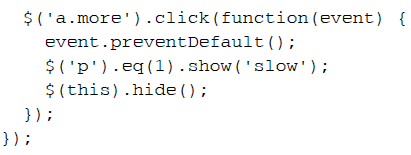
* 当用户单击第一段末尾read more链接时，就会隐藏该链接同时显示第二个段落

**指定显示速度**

* 可以在.show()或.hide()中指定显示速度。

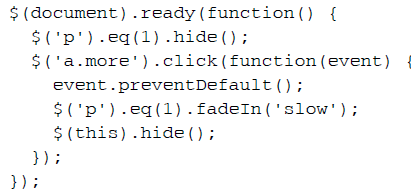


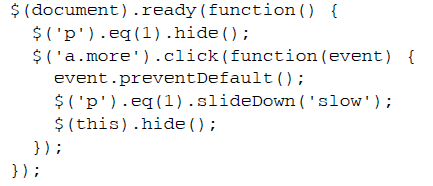
* 可选的 speed 参数规定隐藏/显示的速度，可以取以下值：“slow”、“normal”、"fast" 或毫秒。
* 可选的 callback 参数是隐藏或显示完成后所执行的函数名称。

**代码4-8**

**淡入和淡出**

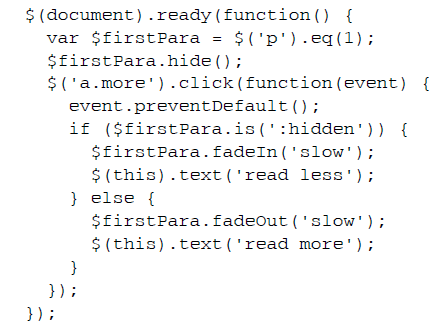
* fadeIn(), fadeOut(): 只改变元素的透明度. fadeOut() 会在指定的一段时间内降低元素的不透明度, 直到元素完全消失. fadeIn() 则相反.
* slideDown(), slideUp(): 只会改变元素的高度. 如果一个元素的 display 属性为 none, 当调用 slideDown() 方法时, 这个元素将由上至下延伸显示. slideUp() 方法正好相反, 元素由下至上缩短隐藏.

**代码4-9**

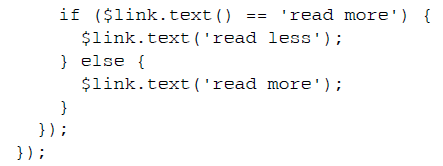
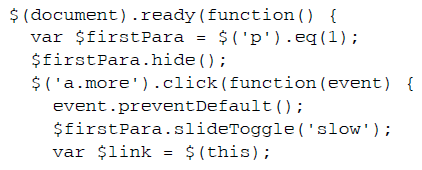
**代码清单4-10**

**切换可见性**

* 有时候，我们需要切换某些元素的可见性，而不像前面例子中那样只把它们显示出来



* 。jQuery提供了一个.toggle()方法，该方法的作用类似于.show()和.hide()方法，而且与它们一样的是.toggle()方法时长参数也是可选的。
* 另一个复合方法是.slideToggle()，该方法通过逐渐增加或减少元素高度来显示或隐藏元素。

**代码清单4-12**

**创建自定义动画**

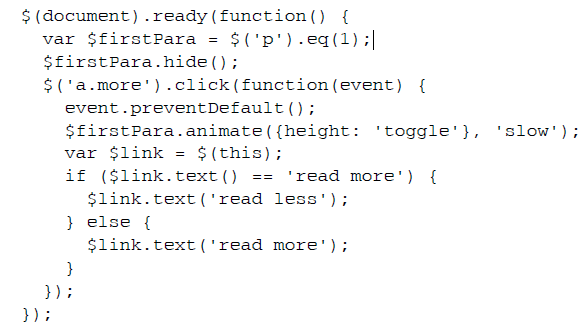
* 除了预置的效果方法外，jQuery还提供了一个强大的.animate()方法，用于创建控制更加精细的自定义动画。

**语法 1**

* $(selector).animate(styles,speed,easing,callback)
* 一个包含样式属性及值的对象
* 可选的时长参数
* 可选的缓动（easing）类型
* 选的回调函数：将在本章后面讨论。
* 其中只有styles是必需的，其他参数都是可选的

**语法2**

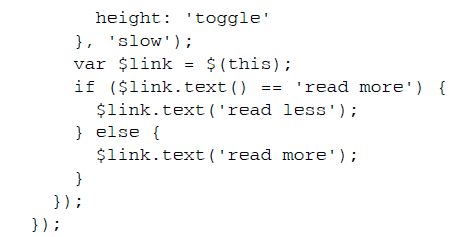
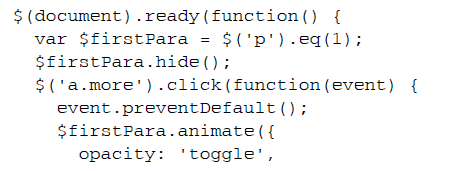
* $(selector).animate(styles,options)
* styles 必需。规定产生动画效果的 CSS 样式和值（同上）。
* options 可选。规定动画的额外选项。
* **可能的值：**
* **speed - 设置动画的速度**
* **callback - 规定动画完成之后要执行的函数**
* **queue - 指示是否在效果队列中放置动画。如果为 false，则动画将立即开始**
* 看一看怎么通过这个低级接口来实现与调用slideToggle()相同的效果



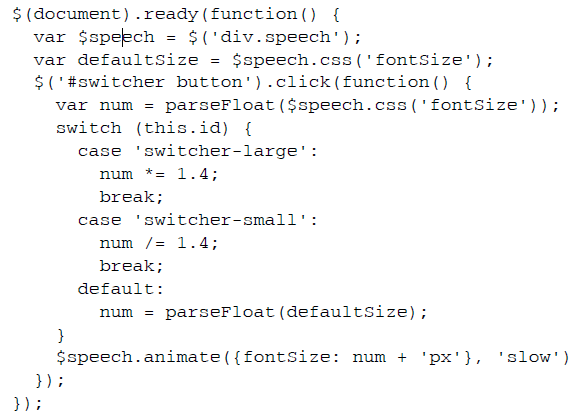
* 通过这个例子可以看出，.animate()方法针对CSS属性提供了方便简写值：'show'、'hide'和'toggle'

**一次给多个属性添加动画效果**

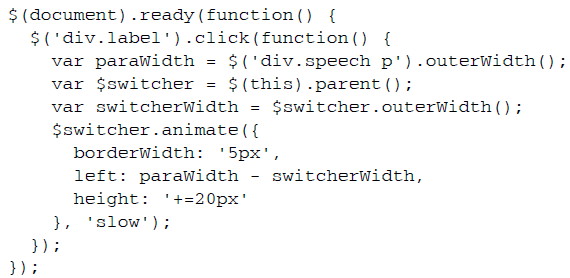
* 使用.animate()方法可以同时修改多个CSS属性。
* 代码清单4-14，要在切换第二个段落时，创建一
* 个同时具有滑动和淡入淡出效果的动画



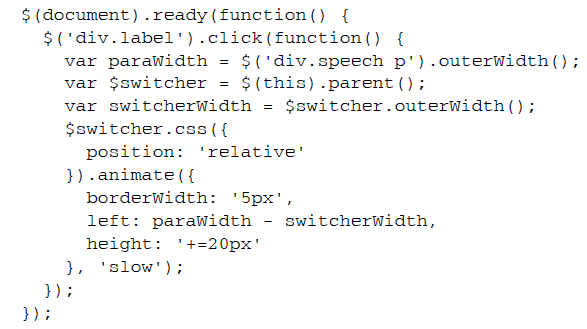
* 不仅可以在简写效果方法中使用样式属性，也可以使用其他CSS属性，如：left，top、fontSize、margin、padding和borderWidth。
* 代码清单4-15
* 要动画实现同样的文本大小变化



* 更复杂的效果，把DIV从页面左侧移动到右侧的同时，让该项的高度增加20像素并使其边框宽度增加到5像素.
* 我们使用jQuery的.**outWidth()**方法来计算宽度，包括内边距及边框宽度.



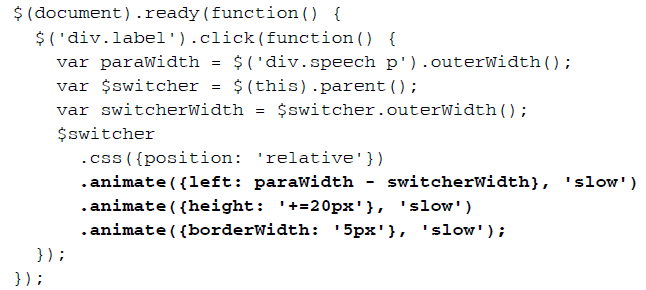
* 在使用.animate()方法时，必须明确CSS对我们要改变的元素所施加的限制。例如，在元素的CSS定位没有设置成relative或absolute的情况下，调整left属性对于匹配的元素毫无作用。



**并发与排队效果**

* 可以看出.animate()方法在为一组特定的元素创建并发效果时非常有用。然而，有的时候我们需要的则是排队效果，即让效果一个接一个地发生。
* 当为同一组元素应用多重效果时，可以通过连缀这些效果轻易地实现排队。

**代码清单4-18**



* 通过使用连缀，可以对其他任何jQuery效果进行排队，而并不限于.animate()方法。
* 我们可以按照下列顺序对<div id="switcher">上的效果进行排队。

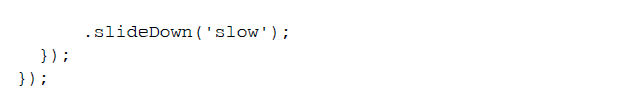
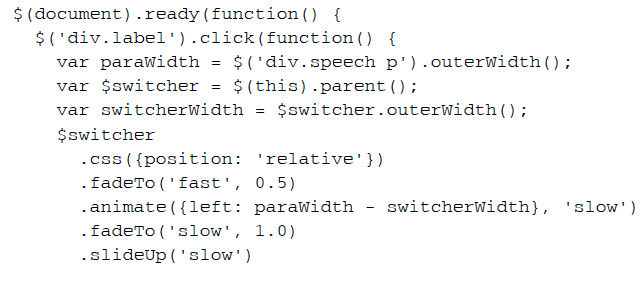
(1) 通过.fadeTo()将其不透明度减退为0.5。

(2) 通过.animate()将其移动到右侧。

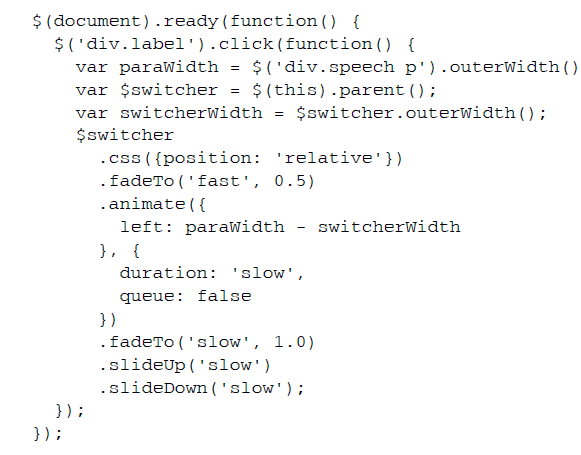
(3) 通过.fadeTo()将其渐增回完全不透明。

(4) 通过.slideUp()隐藏它。

(5) 通过.slideDown()再将其显示出来。

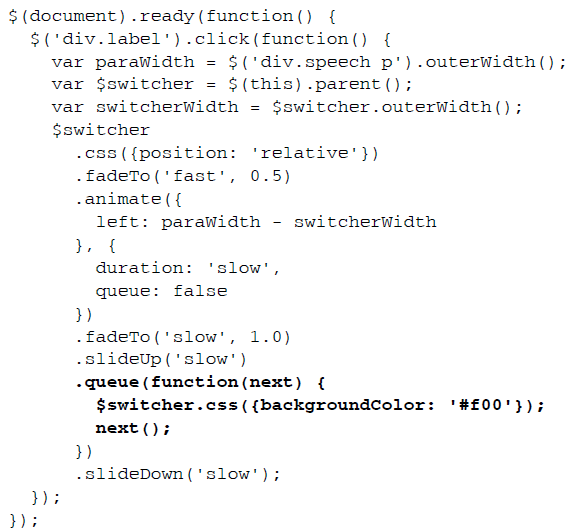


* 越过队列
* 要是想在这个<div>不透明度减退至一半的同时，把它移动到右侧应该怎么办呢？
* queue选项设置为false即可让当前动画与前一个动画同时开始。
* 代码清单4-20

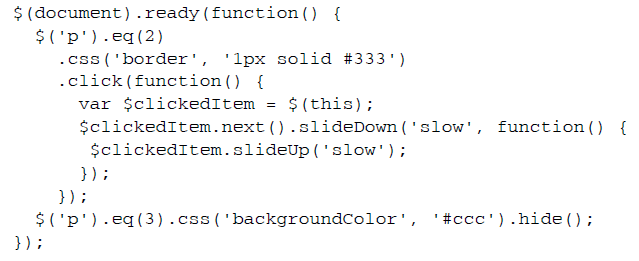


**手工队列**

* 有关为一组元素应用排队效果的最后一个需要注意的问题，就是排队不能自动应用到其他的非效果方法，如.css()。
* 把非效果方法添加到队列中的一种方式，就是使用.queue()方法。
* 代码清单4-22



* 处理多组元素队列效果
* 为了对不同元素上的效果实现排队，jQuery为每个效果方法都提供了回调函数。
* 代码清单4-26 ，在向上滑出一个段落后，向下滑入另一个段落



* 代码清单4-27，<div id="switcher">的背景颜色在它滑上之后滑下之前，变成了红色。
* 使用了交互的完成回调函数而不是.queue()

