

第21章 行 为

现在单纯的静态网页已经很少了，为了更好的吸引用户，网页创作者们采用了各种手段，通过增强网页本身的动态效果和人机交互特性，以更好地突出网页主题，体现网页风格。在网页中创建轮替按钮，就是最常使用的手段，利用轮替按钮，可以将浏览者的操作同页面本身的行为结合起来，从而引起浏览者的注意。

然而，仅仅只有动态轮替按钮的网页还远远不是一个成功的网页，实际上，为了充分体现用户同网页的交互性，我们需要更多的技术和更完善的实现手段。很多人通过在网页中包含JavaScript脚本来实现用户同页面的交互，然而编写脚本既复杂，也需要专门的学习。有鉴于此，Fireworks提供了一种称作“Behavior”（行为）的机制，允许您在构建图片时直接构建各种复杂的交互行为，在将图像导出后，同时也生成相应的代码。这些代码不用进行任何修改，就能够直接在HTML网页中使用。

可以将行为看作是Fireworks的内置JavaScript库，它不仅可以帮助您构建脚本，还可以对现有脚本进行自动的管理。如果您使用过 Dreamweaver，您一定了解行为的概念，Fireworks中的行为特性虽然没有 Dreamweaver强大，但是原理是一样的。

在为按钮设置轮替效果时，实际上也是一种对行为的应用，按钮编辑器将这种行为封装起来，使用户感受不到它。因此可以说，这一章实际上是上一章“按钮和导航条”的延续，是对轮替按钮特性在技术上的延伸和深入。

21.1 概述

在第20章，我们介绍了最简单的行为：轮替按钮。这一章我们将介绍如何在 Fireworks中构建更为复杂的行为。为了帮助读者更好地了解行为，我们首先介绍一下行为的基本概念。

21.1.1 行为的概念

所谓行为（Behavior），顾名思义，就是在网页中进行的一系列动作。通过这些动作，可以实现用户同网页的交互，也可以导致某个任务的执行。

一般来说，一个行为应该由一个事件（event）和一系列动作（action）所组成。例如，当用户将鼠标移动到一幅图像上时，这就产生了一个事件，如果这时候图像发生变化（这就是我们前面介绍过的轮替特性），则实际上就导致了一个动作发生。

在网页中，动作通常由一段 JavaScript代码所实现，利用这段代码可以完成相应的任务，例如切换图像的显示等。在 Fireworks中，您可以使用内置的行为来往文档中添加 JavaScript代码，而不用自己书写。在将文档导出时，不仅会生成图像文件，还会生成相应的 HTML代码，而将代码放入网页相应的位置上，就可以在网页中实现动作。

事件则通常由浏览器所定义，它可以附加到各种页面元素上，也可以附加到 HTML标记中。通常，一个事件总是针对页面元素或标记而言的。例如，在大多数浏览器中，要实现轮替按钮，在按钮图片上会发生 onMouseOver、onMouseOut和onClick这三种事件。当鼠标移动

到按钮上，就发生 `onMouseOver` 事件，而鼠标移动到按钮之外，就发生 `onMouseOut` 事件，而单击按钮，就发生 `onClick` 事件。另一个常见的事件是 `onLoad`，它同图像或是文档正文相关联，当载入文档时，就发生文档的 `onLoad` 事件，而载入图像时，就发生图像的 `onLoad` 事件。

将事件和动作组合起来就构成了行为，例如，将onClick行为同一段JavaScript代码相关联，在单击鼠标时就可以执行相应的 JavaScript代码。通常，我们将事件产生的这个过程称作触发。

不是所有的动作都需要用户的干涉才会发生，例如我们可以在网页中指定某个动作每隔10秒运行一次，当然，这实际上还是由事件触发的，只是这种事件不是通过用户本身的行为而产生的罢了。

21.1.2 Fireworks中的行为特性

在Fireworks中，主要可以创建同轮替操作紧密相关的行为。例如，可以通过构建简单轮替行为来实现简单轮替按钮，或是通过设置增强轮替行为来实现在文档中的“未关联轮替”（Disjoint Rollover）。

所谓未关联轮替，指的是这样一种行为：当用户的鼠标移动或单击文档中的某个区域时，会改变另一处区域的显示。它同轮替按钮的区别在于，轮替按钮生成的动态效果发生在按钮本身上面，而未关联轮替动作发生在文档中其他的区域中。

图21-1中显示了未关联轮替行为的一个例子。将鼠标移动到“儒学探讨”按钮上时，在文档中的另外一个地方就显示一段《论语》文字；而将鼠标移动到“道家寻疑”按钮上时，原先的《论语》文字就变为《老子》文字。也可以将普通的轮替行为同未关联轮替行为相结合，例如在这个例子里，经典文字的变化构成未关联轮替，而按钮本身则是普通轮替。

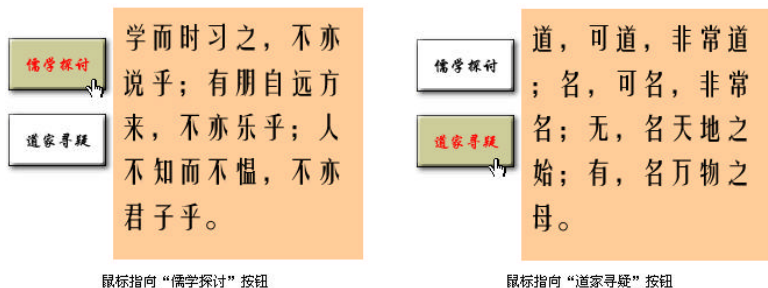


图21-1 未关联轮替

可以看出，在 Fireworks 中，可设置的行为是与图像（包括文本对象）动作密切相关的。这很显然，因为毕竟 Fireworks 是 Web 图像处理程序，而不是一个网页编辑程序。如果希望构建专业的 HTML 行为，最好的方法是使用 Dreamweaver。利用 Fireworks 可以在很大程度上减少我们在 Dreamweaver 中的操作，但其作用范围毕竟有限。需要明白，在 Dreamweaver 中对行为的控制，才是真实可靠的丰富网页的手段。

21.2 创建轮替效果

利用按钮编辑器可以创建轮替按钮。实际上，按钮编辑器的作用就是将应用到按钮图片

上的轮替行为进行封装，以简化相关的操作。如果希望在文档中实现广义的轮替效果，也可以直接在文档窗口中进行。

21.2.1 构建轮替的常规操作

一般来说，要在文档中实现轮替效果，无论是仅包含 2 个状态（弹起状态和移入状态）的简单轮替，还是包含 4 个状态的（弹起状态、移入状态、按下状态和按下时移入状态）的复杂轮替，乃至未关联轮替，都可以采用类似的方法，如下所示：

1) 在文档中，根据轮替的要求创建需要的帧。例如，对于只包含 2 个状态的轮替，需要创建 2 个帧；对于完整的包含 4 个状态的轮替，需要创建 4 个帧。

2) 根据需要，在各个帧中放入相应的图片，这些图片用于表示每帧对应的状态。

3) 在文档中，利用切片工具绘制切片对象，以设置轮替行为的活动区域，也即触发轮替行为的区域。切片对象会位于文档中的 Web Layer 图层中，并被所有的帧所共享。活动区域可以设置在发生改变的图片区域内，也可以在文档中的任意位置。如果活动区域在发生改变的图片区域内，就生成普通的轮替；如果活动区域在文档中其他地方，就可以生成未关联轮替。

4) 如果活动区域在发生改变的图片区域之外，则可以在文档中绘制一个热区对象，以触发这种轮替行为。热区对象同样位于文档的 Web Layer 图层中，并被所有的帧共享。

5) 利用行为面板可以在文档中设置轮替行为。通过行为面板，可以将状态图片的改变动作同鼠标事件相关联，最终形成整体行为。例如，可以通过行为面板来设计如下的行为：在浏览器中，将鼠标指向某个热区时，自动用第 3 帧的图像来替换第 2 帧的图像。我们会在后面介绍行为面板。

6) 按照需要，为每个图像切片（也即活动区域）分派链接，并设置相应的替换文字，或是设置链接目标类型。

7) 按照需要，对图像和切片进行优化。

8) 将文档导出。通常导出操作会生成背景图像、切片以及相应的 HTML 代码。

9) 按照需要，将生成的 HTML 代码放入网页中需要的地方。根据导出操作时设置的目标浏览器类型的不同，可以在相应的 HTML 编辑器中直接使用相应的代码。例如，可以在导出时设置针对 Dreamweaver 3 生成代码，然后在 Dreamweaver 3 中直接将代码插入到需要的地方。

21.2.2 了解轮替状态和帧的关系

在文档窗口中，如果希望构建轮替效果，首先需要了解如何生成轮替元素，也即那些根据不同状态而改变显示的图片，注意，这里的图片指的是广义的图片，它可以包含 Fireworks 文档中可以出现的所有类型的对象，包括路径对象、位图对象以及文本对象等。

通常，在不同的状态下，每种图片内容都被放入到相应的帧中。例如，对于按钮来说，它包含弹起、移入、按下和按下时移入 4 个状态，如果希望在文档窗口中不使用按钮编辑器而是直接创建按钮轮替效果，可以按照如下方法，将相应的按钮图片放入到指定的帧中：

将描述按钮弹起状态的图片放入到 Frame 1 帧中。

将描述按钮移入状态的图片放入到 Frame 2 帧中。

将描述按钮按下状态的图片放入到 Frame 3 帧中。

将描述按钮按下时移入状态的图片放入到 Frame 4 帧中。

当然，上面的这种规律同样适合于其他类型的轮替，例如图像轮替甚至文本轮替等。实际上按钮也就是图片，它可以由任何类型的对象所创建，包括路径对象、位图对象和文本对象等，所以没有什么特殊之处。

21.2.3 定义轮替的活动区域

如果希望将某种行为应用到对象上，除了要设置相应的动作以及触发事件之外，还必须在文档中指定可以触发动作的区域。例如，如果希望在文档中设置轮替效果，必须指定一个区域作为触发轮替动作的区域，当鼠标移入这个区域时就触发轮替动作，而将鼠标移离这个区域时，就恢复图片的默认状态。通常，我们将触发轮替动作的区域称作活动区域，而将轮替效果发生的区域称作轮替元素区域。

对于轮替按钮来说，通常将整个按钮图片所覆盖的区域作为活动区域。换句话说，它的轮替元素区域和活动区域是合二为一的。而对于较复杂的轮替行为来说，活动区域和轮替元素区域可能并不在一起。将鼠标移入到某个区域时，会改变文档中另外某个区域的状态，这就能够实现更为复杂多变，也是更为丰富多彩的动态效果。这也就是所谓的未关联轮替。例如，在图21-1中，两个按钮所在的区域就是活动区域，而显示有经典文字的区域就是所谓的轮替元素区域。当然，按钮本身也具有轮替行为。对于按钮来说，它既是活动区域，也是轮替元素区域。

在Fireworks中，需要使用切片或热区来定义活动区域。具体的选择可以根据需要的不同而不同，通常可以根据如下的规则进行设置：

- 如果活动区域和轮替元素合二为一，则可以根据整个轮替元素覆盖的范围制作切片，并将之设置为活动区域。
- 如果希望在一个切片中通过行为触发一个较小区域的轮替效果，可以在切片中定义热区，然后将热区设置为活动区域。
- 如果希望在文档中，触发另外一个切片中的轮替效果，则可以随意使用切片或热区作为活动区域。

通常使用热区作为活动区域较好，因为这样在最终导出时生成的结果会更简单一些。

21.2.4 认识行为面板

为图像设置行为，主要是通过行为面板来实现的。要显示行为面板，可以打开“Window”菜单，选择“Behaviors”(行为)命令，或是按下Ctrl+Alt+H组合键，图21-2显示了一个典型的行为面板。

1. 添加行为

通过单击行为面板上的添加行为按钮，即可往文档中添加行为。通常，行为必须应用到热区或切片上，因此，应该首先选中切片或热区，然后再添加行为。具体的方法如下：

- 1) 在文档中选择要应用行为的热区或切片，



图21-2 行为面板

也即选中活动区域。

2) 在行为面板上单击“添加行为”按钮，这时会出现如图 21-3所示的菜单，提示您选择行为种类。其中包含如下一些菜单项，相应的行为我们会在后面予以介绍：

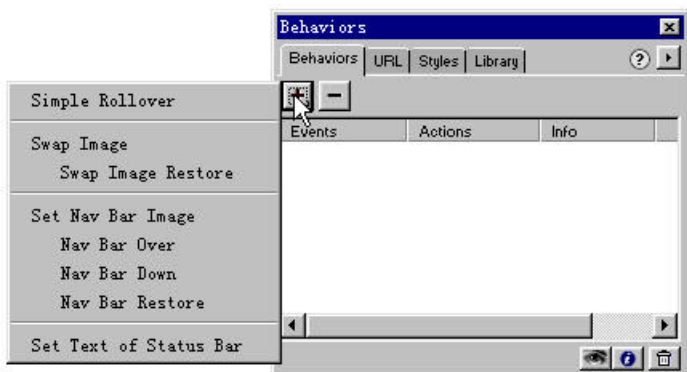


图21-3 添加行为

- Simple Rollover（简单轮替） 该项用于构建只包含弹起和移入状态的简单轮替。
- Swap Image（交互图像） 该项用于创建各种复杂的轮替，包括未关联轮替等。
- Swap Image Restore（交互图像恢复） 该项用于创建使轮替图像恢复为默认状态的行为。
- Set Nav Bar Image（设置导航条图像） 该项用于创建导航条上的导航按钮，其中包含了按钮的按下状态。
- Nav Bar Over（导航条弹起） 该项用于创建导航条按钮的弹起行为。
- Nav Bar Down（导航条按下） 该项用于创建导航条按钮的按下行为。
- Nav Bar Restore（导航条恢复） 该项用于创建导航条按钮恢复默认状态的行为。
- Set Text of Status Bar（设置状态行文本） 该项用于设置触发事件时，浏览器状态行上显示的状态行文字。

3) 在选择了某种行为之后，通常会出现一个对话框，提示您进行相关设置。

4) 设置完毕，确定操作后，相应的行为动作、事件以及相应的操作信息就会出现在行为面板中。其中行为的事件出现在事件列上，行为的动作出现在动作列上，而行为的操作信息出现在信息列中。

例如，在图21-4所示的面板中，我们添加了一个“Set Text of Status Bar”行为。在动作列中，显示动作的名称“Set Text of Status Bar”；在事件列中，显示“OnMouseOver”，表明当鼠标移动到相应的切片或热区之上时，即可触发该动作；而在信息列中，显示“欢迎访问微文工作室主页”字样，表明当鼠标移动到相应的切片或热区上时，在浏览器的状态行上显示“欢迎访问微文工作室主页”的欢迎文字。为了便于观察，我们将面板拉宽了一些。

注意 在添加行为菜单中，有些行为项是往右方缩进的，这表明这些行为通常作为某种行为中的子行为而存在，一般来说，不用我们亲自设置。我们会在后面介绍子行为的概念。

2. 修改触发事件

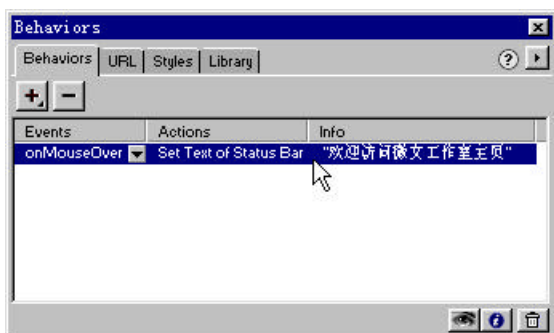


图21-4 一个添加了行为的面板

在添加了行为之后，Fireworks会根据行为的含义自动设置默认的触发事件，但是有时候这种触发事件并不一定是真正需要的，Fireworks允许您根据需要改变触发事件。您可以按照如下方法进行操作：

- 1) 首先在文档窗口中，选中应用了要修改行为的对象，这时在行为面板上就会显示所有应用到该对象上的行为。
- 2) 在行为面板上的行为列表中，选中要修改触发事件的行为项。
- 3) 在事件列的右方有一个箭头按钮，单击该箭头按钮，即可打开如图 21-5所示的下拉菜单，其中包含如下一些事件：

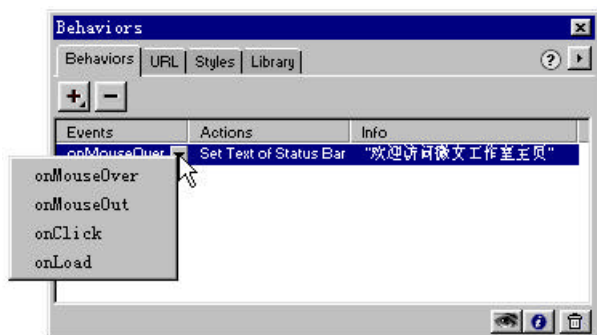


图21-5 修改触发事件

- onMouseOver 当鼠标指针移入应用行为的对象（如切片）范围时，就会触发该事件。
- onMouseOut 当鼠标指针移出应用行为的对象（如切片）范围时，就会触发该事件。
- onClick 当用户单击应用行为的对象（如切片）范围时，就会触发该事件。
- onLoad 当应用了行为的对象（如切片）被完全载入到网页中后，就会触发该事件。

- 4) 根据需要，从下拉菜单中选择合适的事件，即完成了对行为触发事件的修改。

例如，假设这里我们将原先的 onMouseOver事件修改为onMouseOut，则只有当鼠标移离应用了行为的切片区域时，在浏览器的状态行上才会显示欢迎文字。

注意 对于有些行为来说是无法修改其触发事件的。对于这种行为，在事件列的右方会出现一个“锁”形图案，表明事件被锁住，无法修改。例如，对于简单轮替行为来说，就无法修改其事件，如图 21-6所示。因为所谓简单轮替，就是根据鼠标的移入和移出

操作来切换图片显示，如果允许修改事件，也就改变了整个行为的含义，这当然是不行的。

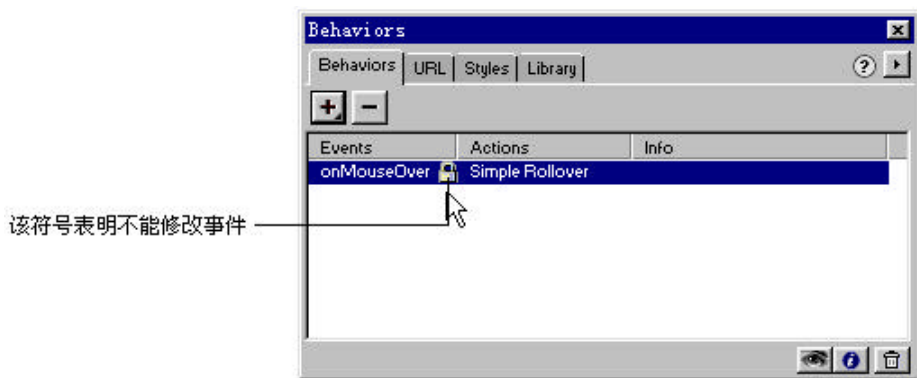


图21-6 无法修改事件的行为

3. 修改动作

Fireworks 允许您修改触发事件，也允许您修改动作。当然，这种对动作的修改，不是改变动作的类型，而只是改变动作操作的对象。您可以按照如下方法进行操作：

1) 首先在文档窗口中，选中应用了要修改行为的对象，这时在行为面板上就会显示所有应用到该对象上的行为。

2) 在行为面板上，双击要修改的行为项或是选中该行为，再打开行为面板菜单选择“Edit”命令；另一个方法是在选中行为项之后，单击行为面板上的“行为属性”按钮。这时会根据行为的不同，出现不同的对话框，允许您对行为所操作的内容进行修改。

3) 修改完毕，确定操作，即可完成修改。修改的结果可以反映到行为面板的信息列上。

4. 显示组合行为

很有可能在行为面板上显示的一个行为项中，实际上包含多个行为。例如，对于简单轮替行为，实际上应该包含两种子行为，一个行为用于设计当鼠标移入对象时的操作，而另一个行为用于设置当鼠标移出对象时的操作。通常，我们将这种包含多个子行为的行为称作组合行为。

默认状态下，在行为面板中只显示组合行为本身，而不显示其中具体包含那些子行为，但是您也可以根据需要，设置在行为面板上同时显示这些子行为，方法如下：

1) 首先在文档窗口中，选中应用了要查看行为的对象，这时在行为面板上，就会显示所有应用到该对象上的行为。

2) 打开行为面板菜单，选中“Show All”（显示全部）命令，或是单击面板上的“显示组合行为”按钮，使之被按下，这时在行为面板中，就会以虚线缩进的形式，在组合行为项之下显示其中包含的子行为，如图 21-7 所示。

3) 如果希望再次隐藏这些子行为，可以重新打开面板菜单，清除对“Show All”菜单项的选中，或是单击面板上的“显示组合行为”按钮，使之被弹起。

注意 可以看到，子行为中通常会包含那些在“添加行为”菜单中以右缩进方式显示的行为。

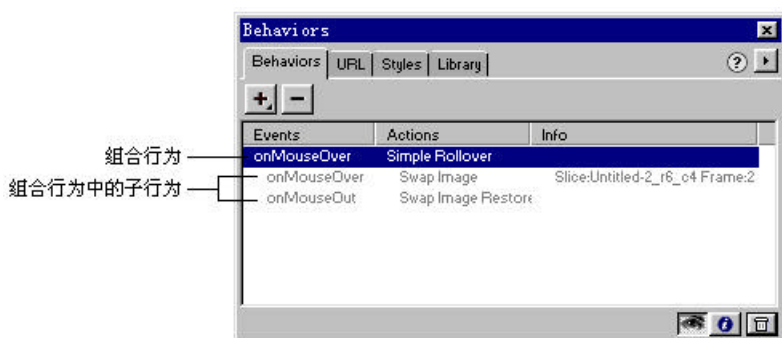


图21-7 显示组合行为

5. 取消行为组合

组合行为中包含的子行为，实际上也是真正的行为，只是它们隐藏于组合行为项之中，无法直接进行编辑。如果需要，可以取消行为组合将这些子行为释放出来，使它们成为真正的独立行为，然后按照需要进行修改。您可以按照如下方法进行操作：

- 1) 首先在文档窗口中选中应用了要查看行为的对象，这时在行为面板上就会显示所有应用到该对象上的行为。
- 2) 选中要取消组合的行为。
- 3) 打开行为面板菜单，选择“Ungroup”(取消组合)命令。

图21-8显示了将图21-7的组合行为取消组合的结果，可以看出，在将行为取消组合后，原先的子行为变为真正的独立行为，同时原先的组合行为项会消失，因为子行为被提取而出之后，它就不复存在了。

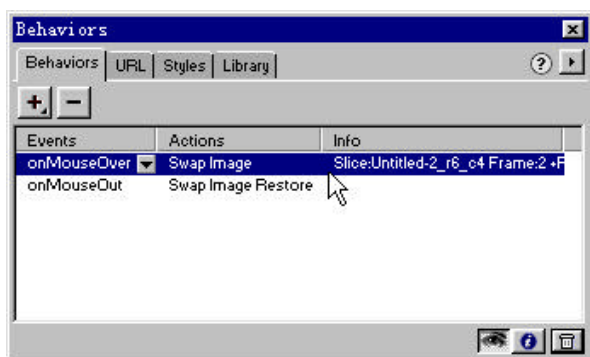


图21-8 取消行为组合

6. 删除行为

如果某种行为不再需要，可以将之从文档中删除，方法如下：

- 1) 首先在文档窗口中选中应用了要删除行为的对象，这时在行为面板上就会显示所有应用到该对象上的行为。
- 2) 在行为面板上选中要删除的行为。
- 3) 打开面板菜单，选择“Delete”命令，或是直接在面板上单击“删除行为”按钮，即可将之删除。

注意 可以单击位于面板左上角的“-”按钮来删除行为，也可以单击位于面板右下角的带有垃圾桶图案的按钮来删除行为。

如果希望删除当前出现在行为面板中的所有行为，可以打开行为面板菜单，选择“Delete All”(删除全部)命令。

21.2.5 为轮替指派链接

可以为轮替指派链接，实际上就是为活动区域指派链接。通常需要首先选中要指派链接的活动区域，然后在URL面板或对象面板上进行设置。图21-9显示了这两个面板。

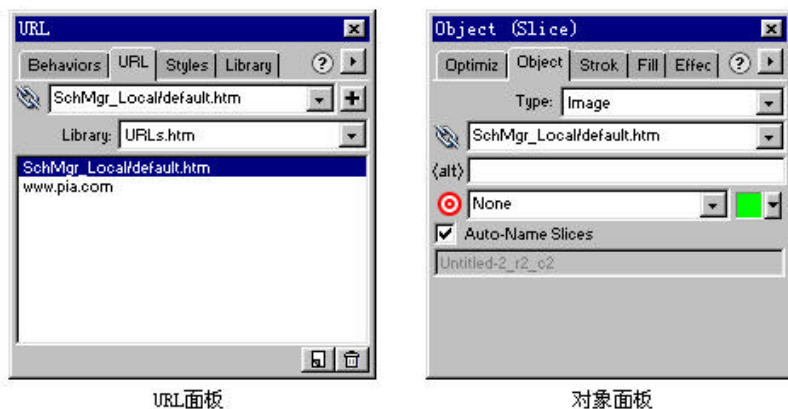


图21-9 URL面板和对象面板

21.2.6 将轮替转换为按钮

严格地说，根据在这一章中介绍的操作所生成的结果并不是按钮，而只是按钮形状的轮替图像。在Fireworks中，按钮的概念主要是指其符号类型而言。通过在文档中生成多个按钮符号的实例，即可生成按钮，而修改符号之后，会对所有的按钮造成影响。

Fireworks允许您将普通的图像轮替转换为按钮，其实质就是将所有帧中的内容作为一个符号来保存。您可以按照如下方法进行操作：

- 1) 在帧面板中，单击“Onion Skinning”(洋葱皮)按钮，然后选中其中的“Show All Frames”和“Multi-Frame Editing”命令。这样所有帧中的内容都能够在文档窗口中显示出来，并且可以在一个帧中同时编辑所有帧中的内容。
- 2) 在文档窗口中，选中所有的轮替内容。由于激活了相应的洋葱皮选项，因此该选中操作同时也选中其他帧中的内容。
- 3) 打开“Insert”菜单，选择“Convert to Symbol”(转换为符号)命令。这时会出现一个对话框，提示您输入符号的名称，并设置符号类型，如图21-10所示。
- 4) 输入名称，并选中“Button”单选按钮。
- 5) 按下“OK”按钮，确定操作。

在将轮替中的各帧转换为按钮之后，实际上是生成了新的按钮符号，并且它会出现在库面板中。同时还会生成一个新的共享层，并将一个按钮的实例放入该层中。原先的轮替中所

包含的对象，例如活动区域中的对象和轮替元素区域中的各个对象，仍然保留在文档中，没有被改变，因此，您还可以在文档中继续使用这些轮替图像资源。



图21-10 将轮替转换为按钮符号

21.3 简单轮替行为

简单轮替行为指的是只包含弹起状态和移入状态的轮替行为。利用按钮编辑器，可以很方便地构建这种轮替效果，但是您也可以直接在文档窗口中构建这种简单轮替。

我们以制作图 21-1 所示的“儒学探讨”按钮为例，介绍生成简单轮替的方法。您可以按照如下方法进行操作：

1) 在文档窗口中，首先在 Frame 1 帧中绘制默认状态的图片，然后在 Frame 2 帧中绘制鼠标移入活动区域后显示的图片。在我们这个例子里，需要绘制如图 21-11 所示的两个按钮图像。

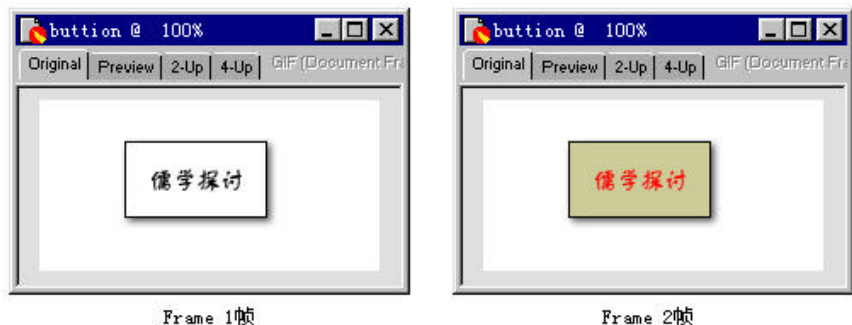


图21-11 位于不同帧的轮替元素

2) 在文档窗口中，绘制一个切片对象或热区对象作为活动区域。具体使用热区还是切片，可以根据自己的需要而定。在我们这个例子里，应该将整个按钮所在的位置设置为切片，如图 21-12 所示。您可以通过选中矩形对象，打开“Insert”菜单，选择“Slice”命令的方法，从路径对象中生成切片。

3) 在文档窗口中，选中活动区域对象，然后在行为面板上单击“添加行为”按钮，打开添加行为菜单，然后选择“Simple Rollover”命令。这时就完成了对这种简单轮替按钮的创建。同时在行为面板上出现相应的行为项，如图 21-13 所示。

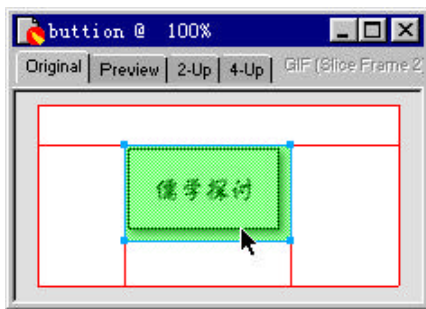


图21-12 设置活动区域

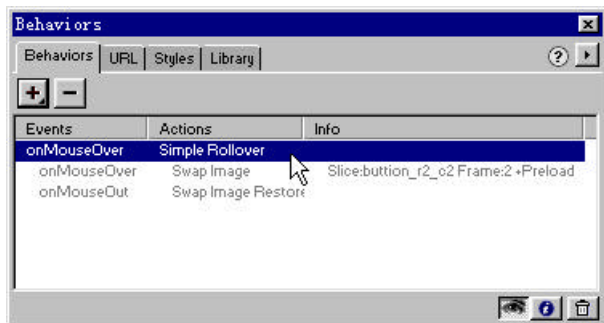


图21-13 显示有简单轮替行为的行为面板

在这个例子里，最终得到的按钮将带有如图 21-14所示的效果。这种只有两个状态的按钮称作简单轮替按钮。

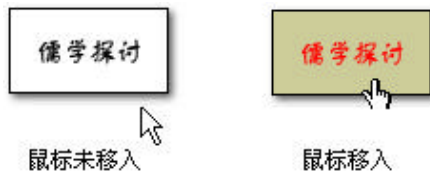


图21-14 简单轮替按钮

21.4 交互图像行为

利用交互图像行为，不仅可以构建简单轮替，也可以构建未关联轮替，假设我们希望构建图21-1中的部分效果，也即当鼠标移动到“儒学探讨”按钮上时，在文档的另外一个区域显示《论语》文字，而鼠标移出“儒学探讨”按钮时，在原先显示《论语》文字的地方显示“欢迎访问”字样，则可以按照如下方法进行的操作：

1) 在前一节中的例子的基础上，将文档的画布尺寸重新调大，并将按钮图像移动到文档中的左上角。

2) 在Frame 1帧中添加一个矩形对象，并在其中放入一个文本对象，且包含“欢迎访问”等字样。在Frame 2帧中，在该矩形对象相应的位置上也放入一个矩形对象，并在其中放入包含《论语》文字的文本对象。最终的结果如图 21-15所示。



图21-15 设置未关联轮替元素

3) 在文档窗口中，将显示有文字的区域（也即轮替元素区域）设置为切片，同时设置触发轮替行为的活动区域。这里活动区域就是按钮所在的位置，在上一节的例子里，我们已经将它设置为切片了。这时的结果如图 21-16所示。其中左方的切片就是活动区域，而右方的切片就是轮替元素区域。

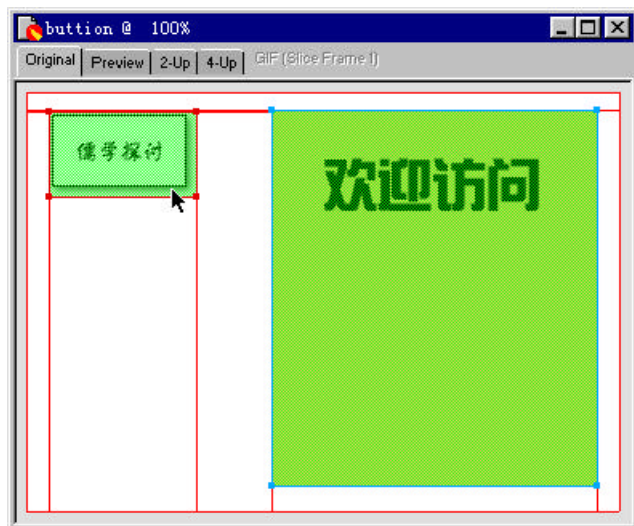


图21-16 设置轮替元素区域和活动区域

4) 在文档窗口中选中活动区域。在这个例子里，我们应该选中左方的按钮切片。

5) 在行为面板上单击“添加行为”按钮，打开添加行为菜单，然后选择“Swap Image”命令。这时会出现如图 21-17所示的对话框，提示您选择轮替元素所在的位置。

6) 选择轮替元素区域，可以采用如下的多种方法：

- 在左方的列表中选择轮替元素区域对应的切片。
- 在右方的预览区域中直接单击要作为轮替元素的切片。

7) 指定在相应的轮替元素区域中用什么图像来作为移入状态的图像。通常，我们将这种移入状态的图像称作轮替图像。您可以采用如下的方法：

- 选中“Frame No”(帧号)单选按钮，然后在后面的下拉列表中设置替换图像所在的帧。例如，这里选择2，表明使用位于Frame 2帧中对应轮替元素区域的图像作为鼠标移入时

显示的图像。

- 选中“Image File”(图像文件)单选按钮,则可以从磁盘上选择需要的图像。您可以在右方的文本框中输入图像路径和名称,也可以单击文件夹按钮,打开一个对话框,然后在磁盘上选择图像文件。

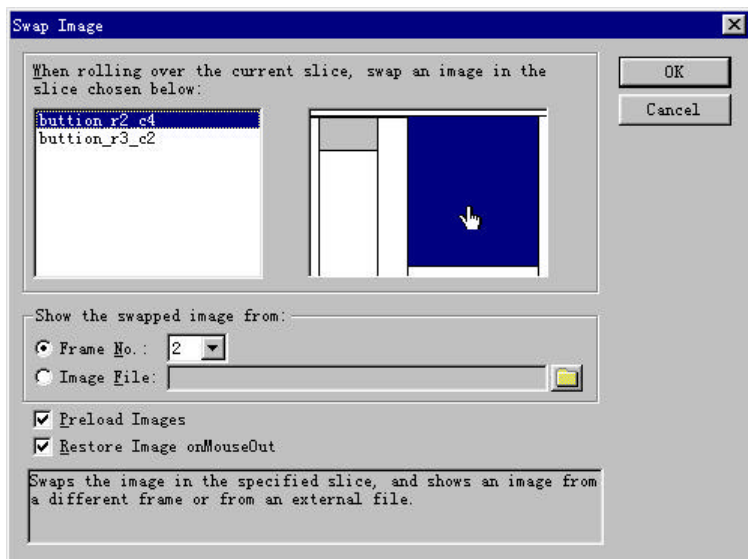


图21-17 设置轮替图像

8) 选中“Preload Images”(预载图像),则在导出的HTML代码中设置预载标记。我们在后面介绍预载的含义。

9) 选中“Restore Image onMouseOut”(在onMouseOut事件发生时恢复图像)复选框,则当鼠标移离活动区域时,轮替区域中的图像将恢复默认状态。如果清除该项,则即便鼠标移离活动区域,轮替元素区域中的图像将不再变化。

10) 设置完毕,单击“OK”按钮,完成操作,在行为面板上就可以看到新添加的行为,如图21-18所示。注意,在行为面板上现在显示有两个行为,其中一个应用于按钮本身的简单轮替行为,它是由上一节设置的;而另外一个行为才是本节设置的交互图像行为。

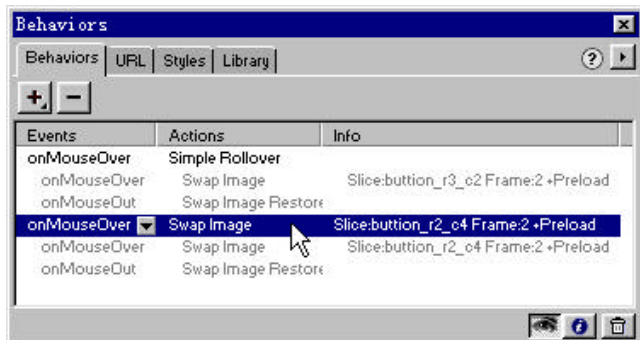


图21-18 包含两个行为的行行为面板

该操作将得到如图21-19所示的结果。



图21-19 生成的未关联轮替效果

注意 在图21-17所示的对话框中，通过选中“Preload Images”复选框，可以确保图像的预载特性。所谓预载，就是指在浏览器中联机下载页面时，一次将所有状态下的图像都下载到本地缓存中，而无论这些状态图片是否被显示过。预载的好处在于将鼠标移动到活动区域时，就可以立刻看到轮替效果，甚至离线的时候也能正确浏览网页，因为网页一旦被完全显示，所有的图片，包括那些隐藏在背后的图片，就会保存在硬盘上了。

如果没有设置预载属性，在网页中的图像只有被显示时才下载。换句话说，在联机时，如果鼠标没有指向活动区域，则只下载默认状态的图像，而只有在网页中将鼠标移动到活动区域时，系统才会开始下载相应状态的图像。这样可能会造成阅读网页的障碍，因为第一次显示轮替图像时，效果会比较缓慢，因为正在进行下载。如果希望离线浏览网页，将很难保证网页的完整性，因为您或许没有看出或显示这些轮替元素的所有状态，对于那些尚未下载的图像就不能正确显示。

在图21-17所示的对话框中，选中“Restore Image onMouseOut”复选框可以确保鼠标移离活动区域时，在轮替元素区域重新显示默认状态的图像。否则，则轮替元素区域将一直保持当前状态。在添加行为的菜单中，在“Swap Image”菜单项下有一项“Swap Image Restore”，实际上就是恢复默认轮替元素状态的行为。所以，您可以手工为活动区域添加这种行为，以确保轮替元素恢复到默认状态。

21.5 设置导航条图像行为

导航条是一系列按钮的集合，用于实现在多个网页间的快速跳转。利用 Fireworks 的设置导航条图像行为，可以非常方便地构建导航条图像。

假设我们希望构建一个包含4个按钮的导航条，您可以按照如下方法进行操作：

- 1) 在文档中，构建4个帧。
- 2) 在Frame 1帧中，绘制4个按钮的弹起状态；在Frame 2中，绘制4个按钮的移入状态；在Frame 3中，绘制4个按钮的按下状态，在Frame 4中，绘制4个按钮的按下时移入状态，如图21-20所示。
- 3) 在文档窗口中，可以根据这四个按钮的轮廓制作切片，如图21-21所示。这些切片将被

作为活动区域。

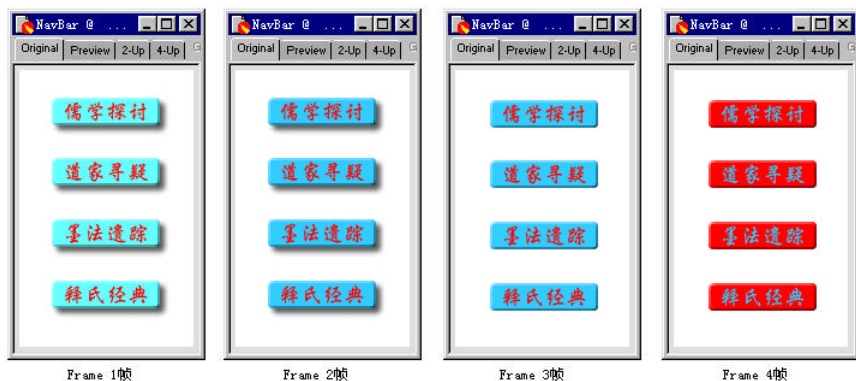


图21-20 导航条按钮的4种状态

4) 选中第一个活动区域，然后在行为面板上单击“添加行为”按钮，打开添加行为菜单，然后选择“Set Nav Bar Image”命令。这时会出现如图21-22所示的对话框，提示您进行相关设置。

5) 在“Nav Bar Preview”（导航条预览）区域，显示当前您选中的切片位置。

6) 选中“Include Over While Down State (Frame4)”（包括按下时移入状态（Frame 4））复选框，则在导航条按钮中包含按下时移入状态。如果清除该复选框，则导航条按钮中只包含前3种状态。

7) 如果希望该按钮在初始状态下时就处于按下状态，则可以选中“Show Down Image Upon Load”（载入时显示为按下图像）复选框。通常，在打开网页时，当前的默认页面会对应一个导航按钮，可以将该导航按钮设置为初始时按下状态。

8) 设置完毕，按下“OK”按钮，确定操作，即可完成了对一个导航条按钮的设置。

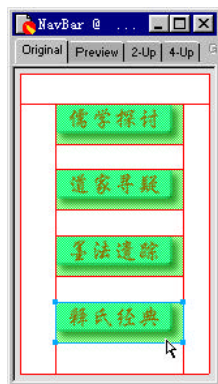


图21-21 设置导航按钮的活动区域

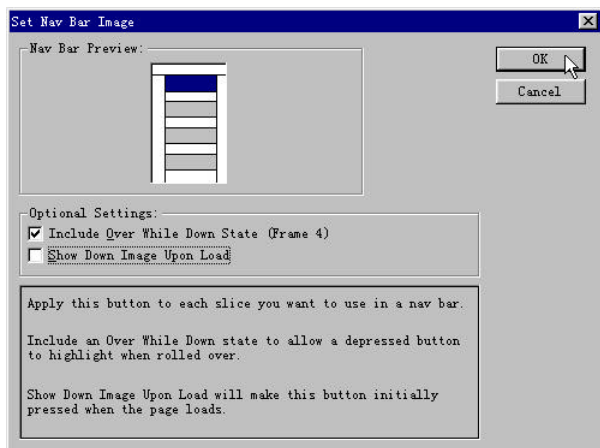


图21-22 设置导航按钮属性

9) 重复上面操作, 对其他的活动区域进行设置, 最终完成所有导航按钮的设置。

在我们这个例子里, 最终可以得到如图21-23所示的结果。图中请注意, 当单击一个按钮时, 按钮保持按下状态, 即使鼠标移离该按钮。按钮的按下状态将保持到单击另外的按钮为止。

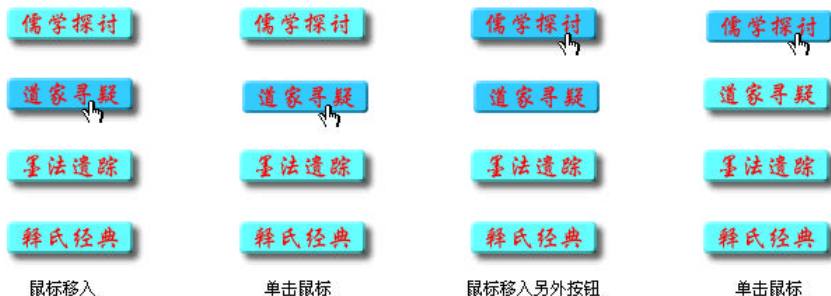


图21-23 生成的导航按钮

21.6 设置状态行文本行为

默认状态下, 当鼠标移动到一个包含链接的区域中时, 在浏览器状态行上会显示该链接的地址。例如, 如果我们为一个按钮指派 <http://www.microtext.com.cn> 的URL地址, 则默认状态下, 将鼠标移动到该按钮上时, 会得到如图 21-24所示的结果。

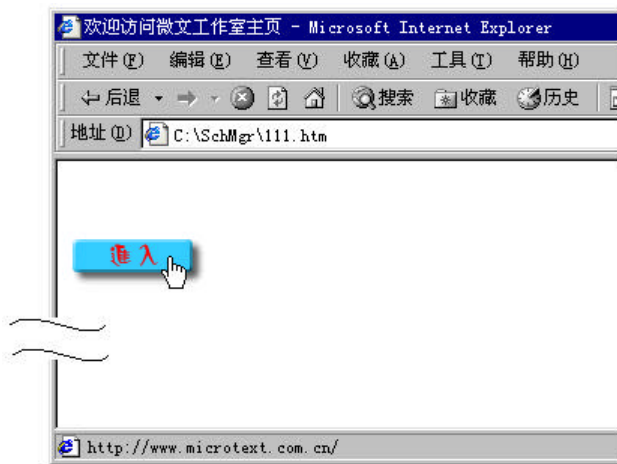


图21-24 在浏览器状态行上显示链接地址

有时候这显得不够人性化, 枯燥的地址不能说明单击按钮之后到底会跳转到什么地方。所以, 在Fireworks中, 允许您定制在浏览器状态行上显示的文本, 以便于对链接位置进行说明。您可以按照如下方法进行操作:

- 1) 在文档窗口中, 选中要设置状态行地址的活动区域。
- 2) 在行为面板上, 单击“添加行为”按钮来添加行为菜单, 然后选择“Set Text of Status Bar”命令, 这时会出现如图 21-25所示的对话框。
- 3) 在“Message”(信息)区域, 输入希望出现在状态行上的文字。
- 4) 单击“OK”按钮, 确定操作。



图21-25 设置状态行文字

经过上面的设置，在浏览器中将鼠标指向按钮时，就会看到，在浏览器的状态行上显示我们这里输入的文字，如图21-26所示。

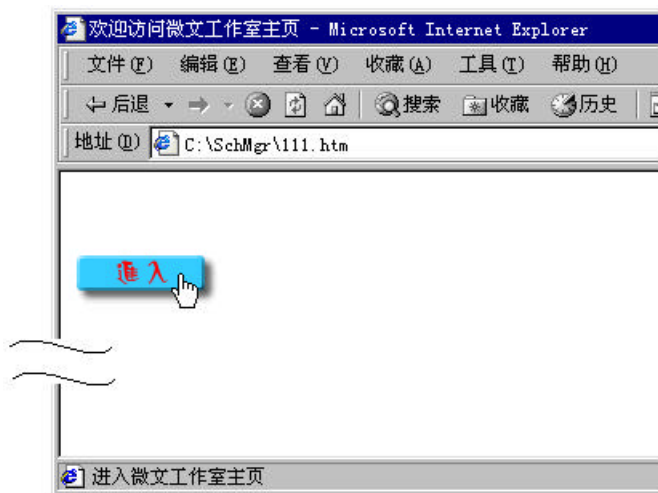


图21-26 改变状态行上的文本