



## 第11章 框 架

框架主要用于在一个浏览器窗口中显示多个 HTML 文档。通过构建这些文档之间的相互关系，从而实现文档导航、文档浏览以及文档操作等目的。框架是 Internet 网页中最常使用的页面设计方式，具不完全统计，大约有 80% 以上的主页都采用了框架技术。

本章将介绍如何在 Dreamweaver 中实现框架。

### 11.1 框架概述

在浏览网页时，我们肯定遇到过这样的情形：在浏览器窗口中被分隔成了几个不同的区域，每个区域中显示不同的文档内容。最常见的方式是将左方或上方的区域设置为目录区域，其中显示文档页面的目录索引或导航条，而将右方或下方的区域设置为主体区域，其中显示网页主体内容。通过单击不同的目录索引项或导航条按钮，就可以在主体区域实现网页之间的导航，在浏览网页的同时，目录索引或导航条始终显示于页面的目录区域中，这样便于用户继续浏览其他的网页。

在上述的情形中实际上就是应用了框架（Frames）技术。利用框架技术，可以将不同的文档显示在同一个浏览器窗口中。通过构建这些显示在同一窗口中的文档之间的相互链接关系，可以实现文档之间的相互控制。

实际上，我们不必连入 Internet，也可以看到框架效果，启动 Dreamweaver 的帮助系统，

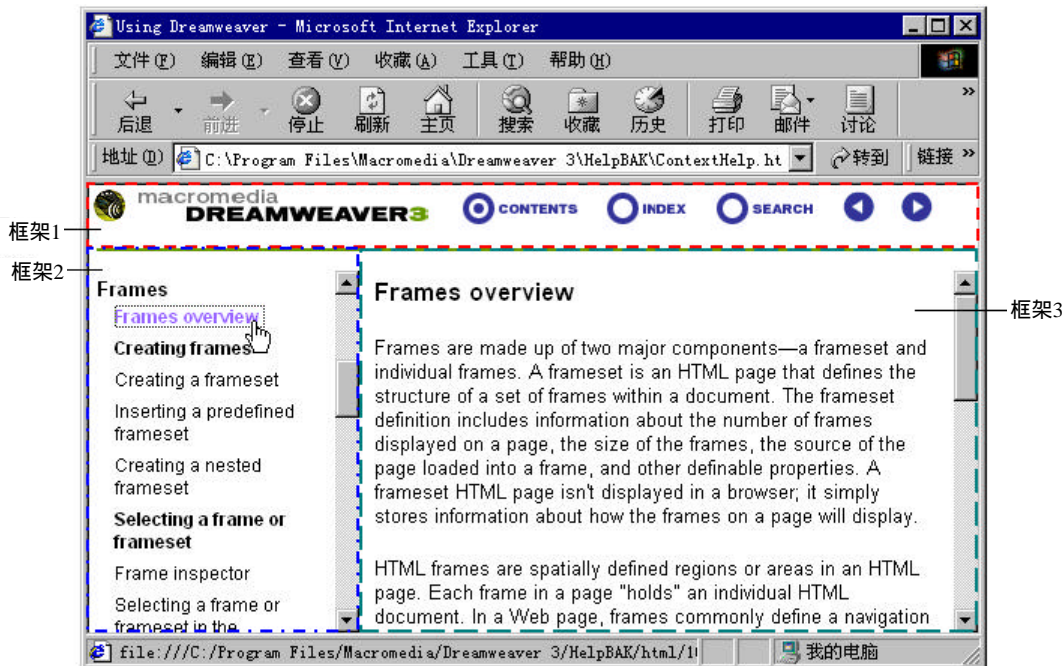


图11-1 使用了框架技术的Dreamweaver帮助系统

您就可以看到带有三个框架区域的浏览器窗口，如图 11-1 所示。

本节将介绍一些框架技术的基本概念，以便于读者更好地了解如何在 Dreamweaver 中设计框架。

### 11.1.1 框架和框架集

一般来说，框架（Frames）技术主要通过两种类型的元素来实现，一个是框架集（Frameset），另一个是框架（Frame）。请注意这每个术语对应的英文单词，其中第一个“框架”是复数形式的单词 Frames，而后一个“框架”对应的单词是单数形式的单词 Frame。在中文中我们很难区分它们，但前者表示技术，后者表示具体的元素，它们确实是概念不同的东西，所以有时候，我们又将前者的“框架”（Frames）称作“框架技术”，而将后者的“框架”（Frame）称作“框架文档”。

所谓框架集，顾名思义，就是框架的集合。框架集实际上是一个页面，用于定义在一个文档窗口中显示多个文档的框架结构。例如，它可以决定文档窗口中显示的文档数目、每个文档的大小，以及文档被载入框架集窗口中的方式等。一般来说，框架集文档中的内容不会显示在浏览器中。所以有时候，我们可以将框架集仅仅看成是一个可以容纳和组织多个文档的容器。

所谓框架，就是在框架集中被组织和显示的文档。在框架集中显示的每个框架事实上都是一个独立存在的 HTML 文档。

图 11-2 显示了框架和框架集之间的关系。在图中，1 个框架集中包含了 3 个框架文档。它很形象地显示了图 11-1 中的框架结构关系。在图 11-1 中，框架集是 ContextHelp.htm 文档，框架 1 是 navigation.htm 文档；框架 2 是 cont11.htm 文档；框架 3 是 10frames1.html 文档。您可以分别在相应的区域，单击鼠标右键，然后选择“查看源文件”命令来查看每个框架文档的源文件；打开浏览器的“编辑”菜单，选择“查看源文件”命令，可以查看框架集文档本身的源文件。

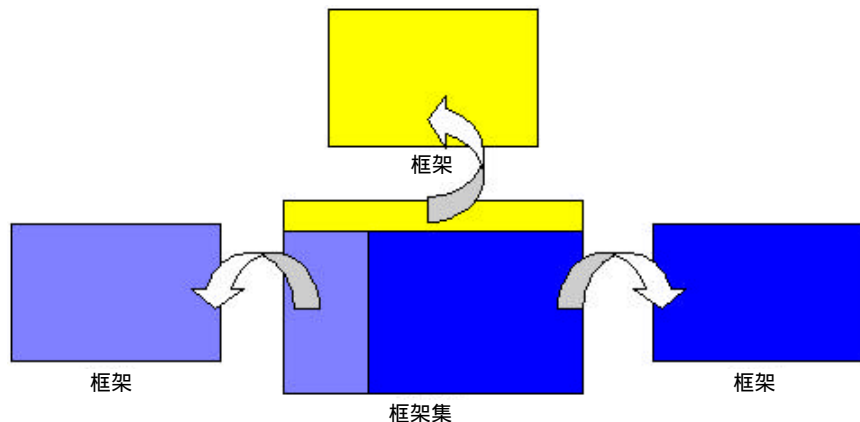


图11-2 框架和框架集

框架技术经常用于实现文档的导航。其作用有些像 Windows 的资源管理器，一个窗格显示站点中的文件结构，另一个窗格显示站点中文档的具体内容。利用框架技术将两个或多

个网页联系起来，使得在一个网页中的操作，会对另一个网页中发生影响。

### 11.1.2 用HTML实现框架技术

框架技术最早是由 Netscape 公司提出的，并在 Netscape Navigator 浏览器中最早得以实现，最初出现的时候，很多人对之提出了批评，认为它搞乱了文档之间的关系，使对文档的控制变得难于理解。

然而随着它在 Internet 上使用的普及，其巨大的优势也逐渐显露出来，终于被 W3C 组织接纳，最终被写入 HTML 规范。目前绝大多数浏览器，包括微软的 Internet Explorer，都提供了对它的完美支持。

在 HTML 代码中，实现框架技术非常简单，它主要通过三个标记：`<frameset>` 标记、`<frame>` 标记和 `<noframes>` 标记来实现。

#### 1. `<frameset>` 和 `</frameset>` 标记

采用框架技术的网页和普通的网页，在代码结构上有稍许差别。普通的网页将文档的主体内容放置于 `<body>` 和 `</body>` 标记间，而在采用框架技术的网页中，则用 `<frameset>` 和 `</frameset>` 标记替代了 `<body>` 和 `</body>` 标记。

`<frameset>` 和 `</frameset>` 标记决定了在文档中如何对框架文档进行布局，换句话说，它决定如何对浏览器窗口进行分区，以及在每个分区中显示什么文档的内容。

大多数情况下，在其中还包含 `<noframes>` 和 `</noframes>` 标记，以放置当浏览器不支持框架时显示的内容。

下面显示了一个使用框架技术的网页代码，它将浏览器窗口分割为左右两个等宽的窗格，左边窗格中显示名为 `left.html` 的文档，而右边窗格显示名为 `right.html` 的文档：

```
<html>
<head>
<title>框架技术示例 </title>
</head>

<frameset cols="50%, 50%">
  <frame src="left.html">
  <frame src="right.html">
</frameset>
</html>
```

`<frameset>` 标记带有很多属性，除了通用的属性之外，在控制框架集结构方面，`cols` 属性和 `rows` 属性最为重要，我们将之介绍如下：

`cols = MultiLengths` `cols` 属性用于定义在框架集中每个框架的宽度，其中 `MultiLengths` 用于描述多个框架的宽度列表。每个框架宽度之间以逗号分隔。

框架的宽度可以是像素值，可以是相对于整个框架集文档宽度的百分比，也可以使用相对宽度来定义。如果要用像素值定义框架宽度，可以直接用数值来表示，例如，“20”表示宽度是20像素；如果要用百分比定义框架宽度，则可以用数字后带有百分比符号的形式来表示，例如，“20%”表示该框架的宽度是整个框架集宽度的20%；如果要用相对宽度定义框架宽度，则可以用整数后加星号（\*）的形式来表示，例如，“`cols = "2*, 5*"`”表示左方框架宽度同右方框架宽度的比例为2:5。

在下面的代码中，定义了一个在水平方向上包含3个框架的框架集文档，其中，左方的框

架宽度为40像素，右方的框架宽度为整个框架集宽度的30%，而中间的框架宽度则是剩余的宽度：

```
<frameset cols = "40, *, 30%">
```

`rows = MultiLengths` `rows`属性用于定义在框架集中每个框架的高度，其中 `MultiLengths` 是用于描述多个框架的高度列表。每个框架高度之间以逗号分隔。

与 `cols` 属性类似，框架的高度可以是像素值，可以是相对于整个框架集文档高度的百分比，也可以使用相对高度来定义。

在下面的代码中，定义了一个在垂直方向上包含3个框架的框架集文档，其中，上方的框架高度为40像素，下方的框架高度为整个框架集宽度的30%，而中间的框架高度则是剩余的宽度：

```
<frameset rows = "40, *, 30%">
```

`frameborder = bool` `frameborder`属性用于确定在框架集中是否显示框架边框。实际上，它设置的就是是否显示框架和框架之间分隔条。

如果属性值为1或Yes，表明该框架集中所有的框架都显示边框。

如果属性值为0或No，表明该框架集中所有的框架都不显示边框。

`bordercolor = color` 该属性设置框架集中所有框架边框的颜色值，其中 `color` 是颜色的十六进制数值。

`border = length` 该属性用于设置框架集中各框架边框的宽度，其中 `length` 是宽度值，其单位是像素。

`framespacing = length` 该属性用于设置框架集中各框架之间的间隔，其中 `length` 是距离值，其单位是像素。

## 2. <frame>和</frame>标记

`<frameset>`和`</frameset>`标记决定了在文档中如何对框架文档进行布局，而 `<frame>`和`</frame>`标记则具体定义了框架集文档中的每个框架。每个 `<frame>`和`</frame>`标记对应于一个框架，它们必须出现在 `<frameset>`和`</frameset>`标记之间。

`<frame>`带有很多非常重要的属性，我们将之介绍如下：

`name = cdata` 在使用框架技术的网页中，`<frameset>`标记的 `name` 属性特别重要，它用于为每个框架文档设置名称。在利用链接在框架集文档窗口的窗格中打开框架文档时，通过将 `name` 属性和链接的 `target` 属性相结合，可以指定在哪个窗格中打开文档。

`src = url` `src`属性用于指定框架文档的来源，其中 `url` 是框架文档的URL地址。一般来说，框架文档是一个HTML文档，它以文件的形式，单独存储于磁盘上，从文件的角度上来看，框架文档和框架集文档是分属不同文件的。

`frameborder = bool` `frameborder`属性用于确定在框架集中是否显示框架边框。实际上，它设置的就是是否显示框架和框架之间分隔条。

如果属性值为1或Yes，表明该框架与其四周的框架相邻处显示分隔条。

如果属性值为0或No，表明该框架与其四周的框架相邻处不显示分隔条。

该属性设置会覆盖 `<frameset>` 标记中 `frameborder` 属性的设置。

`marginwidth = pixels` 该属性用于定义框架左右两边的空白间距，也即框架中内容同框架左右边框的距离。其中 `pixels` 是空白间距的像素值。

`marginheight = pixels` 该属性用于定义框架上下两边的空白间距，也即框架中内容同框架上下边框的距离。其中 `pixels` 是空白间距的像素值。

`noresize` 该属性用于设置是否允许调整当前框架的大小。如果在 `<frame>` 标记中出现该属性，则表明框架大小不可调节；如果在 `<frame>` 标记中没有出现该属性，则表明框架大小可以调节，这时可以利用鼠标拖动该框架同其他框架之间的分隔条，改变框架大小。

`scrolling = method` 该属性用于确定是否允许滚动框架中的内容。其中 `method` 是滚动属性，其可能的取值有 “auto”、“yes” 或 “no”。

如果属性值为 “auto”，则如果框架文档内容超出了框架的大小时，会出现框架滚动条，允许通过拖动滚动条，显示框架内容。

如果属性值为 “yes”，则无论框架文档中的内容是否超出框架的大小，都会显示框架滚动条。

如果属性值为 “no”，则即使框架文档中的内容超出了框架的大小，也不会出现框架滚动条。

`bordercolor = color` 该属性设置框架边框的颜色值，其中 `color` 是颜色的十六进制数值。

该属性设置会覆盖 `<frameset>` 标记中 `bordercolor` 属性的设置。

### 3. `<noframes>` 和 `</noframes>` 标记

不是所有的浏览器都一定支持框架技术，在 Internet 上存在着五花八门的浏览器，有些浏览器甚至是基于字符模式的。如果您的主页使用了框架技术，则使用不支持框架技术的浏览器进行浏览时，用户将看不到任何东西。

为了避免这种情况的发生，一般应该在框架集文档中显示一些提示信息，告诉用户当前网页使用了框架技术，需要使用支持框架的浏览器观看，如有必要，还可以制作没有使用框架技术的网页，以便所有的人都能访问网站。

利用 `<noframes>` 和 `</noframes>` 标记，可以解决在不支持框架的浏览器中显示提示信息的问题。如果浏览器支持框架，则会完全忽略 `<noframes>` 和 `</noframes>` 标记间的内容。如果浏览器不支持框架，则会显示 `<noframes>` 和 `</noframes>` 标记间的内容。您甚至可以在 `<noframes>` 和 `</noframes>` 标记之间添加 `<body>` 和 `</body>` 标记，然后像编辑正常网页一样进行编辑，构建不包含框架的网页内容。

在下面的例子里，我们创建了一条提示，告诉用户当前网页使用了框架技术，并提供一个链接，引导用户访问没有使用框架技术的网页：

```
<html>
<head>
<title>框架技术示例 </title>
</head>

<frameset cols="50%, 50%">
  <frame src="left.html">
  <frame src="right.html">
</frameset>
<noframes>
<body>
```

`<p>`本页面使用了框架技术，如果您看到这条提示信息，表明您的浏览器不支持框架，无法正确浏览本网页。您可以单击 `<a href="/noframes/index.html">这里</a>`，进入不包含框架的



```
主页。</p>
</body>
</noframes>
</html>
```

## 11.2 创建框架

了解了框架的概念和实现框架的代码之后，我们可以开始介绍如何在 Dreamweaver 中构建框架。

框架的构建总是从一个正常的文档开始的，利用 Dreamweaver，您可以很容易地将正常方式的文档分割为多个框架窗格，从而构建框架。在 Dreamweaver 中，可以从无到有自行设计各种框架，也可以使用 Dreamweaver 预设的几种框架格式。

### 11.2.1 创建框架集

在创建框架之前，最好首先将框架的边框显示特性激活，方法是打开“View”菜单，选中“Frame Border”（框架边框）命令，这时的文档窗口四周出现框架边框，如图 11-3 所示。

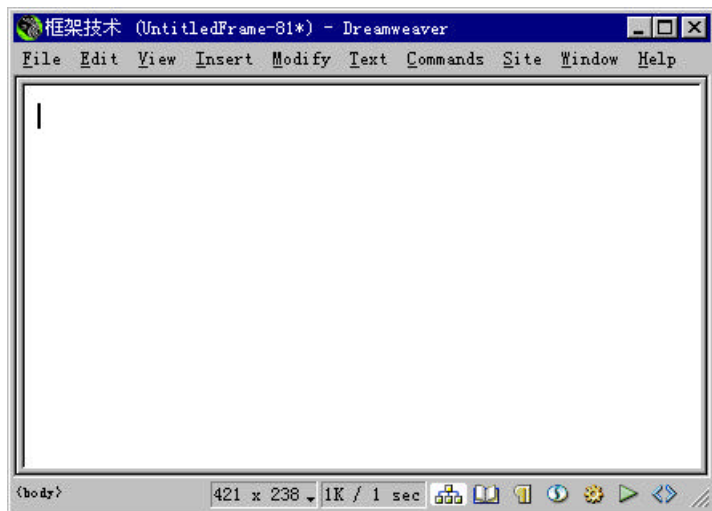


图11-3 显示了框架边框的文档窗口

#### 1. 创建框架

假设我们创建了一个新的空白文档，从该文档的基础上构建框架。您可以按照如下方法进行的操作：

方法一：按住 Alt 键，然后用鼠标拖动文档窗口四周显示的框架边框，将之拖动到希望的位置上，释放鼠标，即可构建框架。

将左边框往右拖动，可以向右分割框架。

将右边框往左拖动，可以向左分割框架。

将上边框往下拖动，可以向下分割框架。

将下边框往上拖动，可以向上分割框架。

请参看图 11-4 了解拖动边框所构建的框架和文档现有内容之间的关系。



图11-4 利用拖动框架边框所构建的框架同现有文档内容之间的关系

如果在按住 Alt 键的同时，拖动的是框架的边框角，则可以在左右或上下两个方向上同时分割框架，如图 11-5 所示。

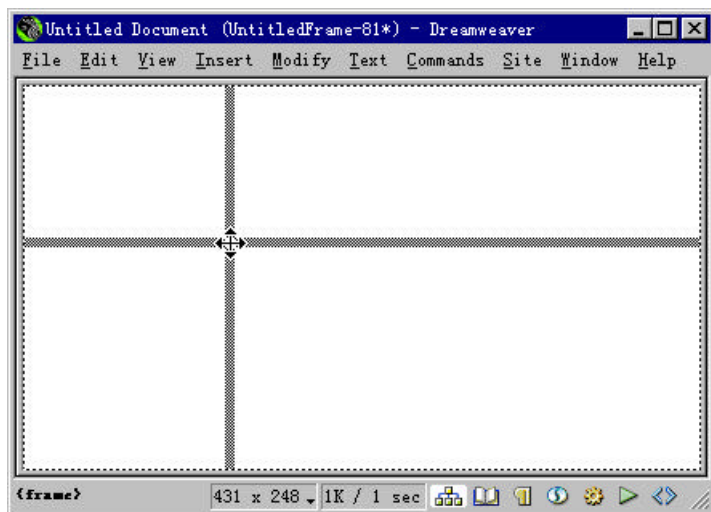


图11-5 拖动框架边框角创建框架

拖动框架边框左上角，可以往右下方分割框架。

拖动框架边框左下角，可以往右上方分割框架。

拖动框架边框右上角，可以往左下方分割框架。

拖动框架边框右下角，可以往左上方分割框架。

图 11-6 显示了拖动框架边框角创建框架的拖动方向及其含义。



图11-6 利用拖动框架边框边角所构建的框架同现有文档内容之间的关系

继续按住 Alt 键，拖动各框架边框（包括最初出现在文档窗口四周的边框，以及生成框架后各框架的边框），可以继续构建框架。

如果希望在某个框架窗格中分割框架，构建子框架，可以首先选中该框架，然后在按住 Alt 键的同时拖动边框。选中框架的方法可以参看本章 11.4.2 一节。图 11-7 显示了构建子框架同构建主框架的区别。

方法二：



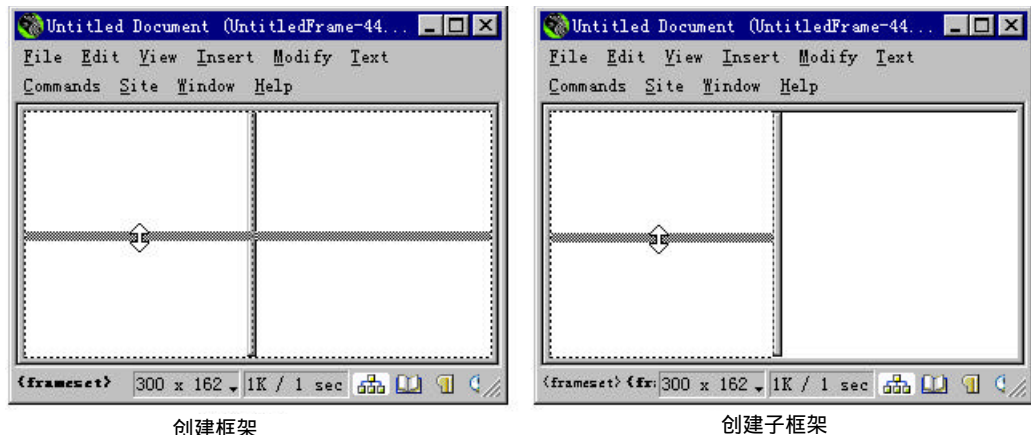


图11-7 构建框架和构建子框架

1) 单击要分割框架的窗口，将插入点放置入窗口中。如果已经存在框架，则需要单击某个框架窗格，将插入点放入该窗格内。

2) 打开“Modify”菜单，选择“Frameset”，打开相应的子菜单。

3) 选择相应的分割框架命令，其含义如下：

- “Split Frame Left”（向左分割框架） 选择该命令，则将当前文档窗口区域在左右方向上分割为两个窗格。当前文档窗口区域中的内容放置到左窗格中。
- “Split Frame Right”（向右分割框架） 选择该命令，则将当前文档窗口区域在左右方向上分割为两个窗格。当前文档窗口区域中的内容放置到右窗格中。
- “Split Frame Up”（向上分割框架） 选择该命令，则将当前文档窗口区域在上下方向上分割为两个窗格。当前文档窗口区域中的内容放置到上窗格中。
- “Split Frame Down”（向下分割框架） 选择该命令，则将当前文档窗口区域在上下方向上分割为两个窗格。当前文档窗口区域中的内容放置到下窗格中。

4) 继续单击某个框架窗格，将插入点放入其中，然后重复上面的操作，继续分割窗口，即可构建嵌套框架。

请参看图11-4了解这些菜单项对应的操作含义。

**注意** 在构建框架时，特别要注意的是现有文档同分割后的框架集之间的关系。在当前文档中分割框架，实际上是将当前文档中的内容保留在某个框架文档中，换句话说，在创建框架之后，原先的文档以一个框架文档的形式出现，很多人以为它变为框架集文档，这是错误的。这一点，您可以从文档窗口的标题栏上看出。例如，在一个标题为“框架技术”的文档中分割出框架，这时文档窗口的标题栏上会变化为一个未定义标题的形式，这表示当前文档窗口中的内容已经不是原先的文档，而是一个尚未保存的框架集了，如图11-8所示。

## 2. 删除框架

如果希望删除创建的框架，只需要用鼠标拖动框架边框，将之拖离页面或拖动到父框架边框上即可，如图11-9所示。

**注意** 您只能删除框架，却不能合并框架，这很显然，每个框架都是一个文档，删除框

架，实际上是从框架集文档中删除相应的 <frame> 标记，并重新设置 <frameset> 标记的 cols 或 rows 属性。框架文档并不变化，如果可以合并框架，则等同于将两个文档内容合并成一个文档内容，这是不可能的。

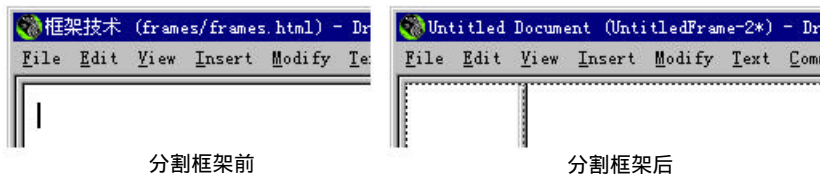


图11-8 分割框架前后文档窗口标题栏的变化

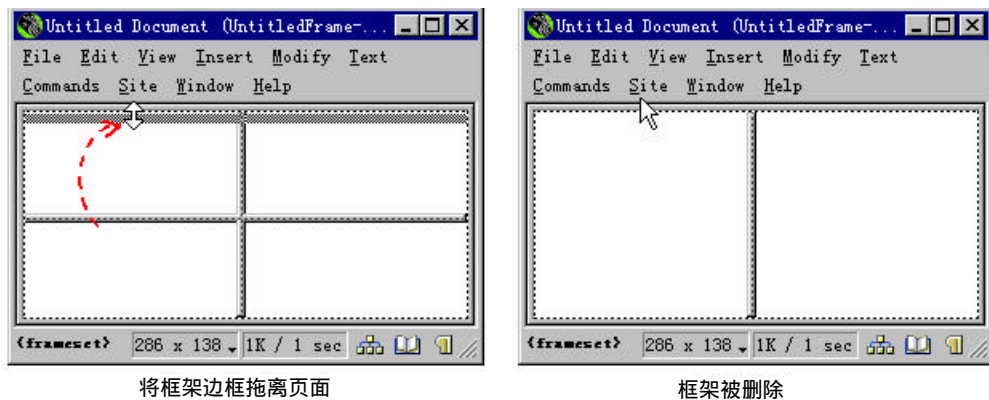


图11-9 删除框架

### 11.2.2 创建预定义框架集

在Dreamweaver中，预先定义了几种框架集形式，允许您快速在文档中构建框架，可以按照如下方法进行操作：

- 1) 单击要分割框架的窗口，将插入点放置入窗口中。如果已经存在框架，则需要单击某个框架窗格，将插入点放入该窗格内。
- 2) 在对象面板上，打开下拉列表，选择“Frames”，使对象面板上显示所有的框架按钮，如图11-10所示。
- 3) 单击相应的按钮，即可在当前文档窗口中构建出相同格式的框架。您也可以直接将按钮拖动到文档窗口中构建框架。
- 4) 继续重复上面的操作，可以创建嵌套框架。

注意 在对象面板中同样可以看出原先文档内容同分割框架所生成框架集之间的关系。其中蓝色的区域就是在创建框架后，原先文档内容所在的位置，也即原先文档作为框架文档存在的位置，其余的空白区域是新创建的框架文档所在位置。



图11-10 利用对象面板构建框架

### 11.2.3 创建NOFRAMES内容

如果浏览器不支持框架，则无法显示框架集和框架文档内容，所以应该在框架集文档中，

创建位于<noframes>和</noframes>标记间的提示信息，告诉用户如何访问网页。通常，将这种内容称作NOFRAMES内容，或非框架内容。

因为在文档窗口所显示的框架集中，各个窗格中的内容都是相应框架文档中的内容，要放置非框架内容，可能会让人有无从下手之感，所以 Dreamweaver 专门提供了相应的方法，允许您很方便地构建非框架内容，可以按照如下方法进行操作：

1) 打开“Modify”菜单，选择“Frameset”，再选择“Edit Noframes Content”（编辑非框架内容）命令。

2) 这时文档窗口变为一个普通的文档窗口，允许您在其中编辑非框架信息，如图 11-11 所示。您可以按照编辑普通文档的方法，编辑其中的内容，例如，可以在其中输入文字、插入图像以及创建链接等。

3) 再次打开“Modify”菜单，选择“Frameset”，再选择“Edit Noframes Content”命令，可以返回到原先的框架集文档窗口中。

该操作实际上是在文档集窗口中的 <noframes>和</noframes>标记间放置 <body>和</body>标记，然后在其中添加相应的文档内容。图 11-11 所示的情形对应如下的代码：

```
<noframes><body bgcolor="#FFFFFF">  
<p>本页面使用了框架技术，如果您看到这条提示信息，表明您的浏览器不支持框架，无法正确浏览本网页。  
</p>  
<p>您可以单击这里，进入不包含框架的主页。</p>  
</body></noframes>
```

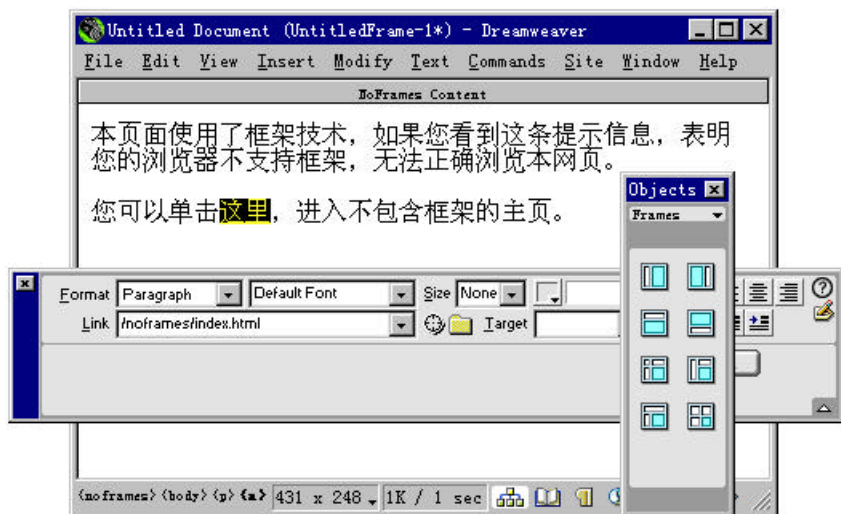


图11-11 编辑非框架内容

### 11.3 存储框架和框架集文件

我们知道，在框架集文档中组织各个框架文档，每个框架文档事实上都应该作为一个 HTML 文件存储在磁盘上，例如，如果一个框架集中包含左右两个窗格，则实际上应该存在 3 个文件，一个是框架集对应的文件，一个是左方框架对应的文件，另一个是右方框架对应的文件。

每次从文档窗口中构建框架时，Dreamweaver会为新生成的框架文档以及框架集文档临时命名，例如，可能会用诸如 UntitledFrame\_1 之类的名称命名框架集文档，以诸如 Untitled-1，Untitled-2 之类的名称命名各个框架文档。

在 Dreamweaver 中，您可以单独保存某个框架文档，或单独保存框架集文档本身，也可以将整个框架集（包括框架集文档本身以及其中的各个框架文档）保存。

如果希望在浏览器中预览框架效果，必须将各个框架文档以及框架集文档存储到磁盘上。

值得一提的是“文件”和“文档”这两个词，文件的英文是“File”，而文档的英文是“Document”，“文件”的概念较注重其磁盘存储特性，而“文档”的概念则较强调其中内容的可编辑性。但是在很多情况下，我们可以将“文件”和“文档”这两个词混用。

### 11.3.1 存储框架集文档

要存储框架集文档本身，可以按照如下方法进行操作：

- 1) 打开“File”菜单，选择“Save Frameset”（存储框架集）命令，即可保存框架集文档。
- 2) 如果框架集文档尚未存储，则会打开文件存储对话框，允许您选择存储路径和输入文件名称。选择路径并输入新框架集文档名称，再单击“Save”（保存）按钮，即可保存框架集文档本身。
- 3) 如果框架集文件已经被存储过，则该操作将在原先的基础上再次保存文档内容，不会再次出现文件存储对话框。

如果希望将框架集文档以另外的名称保存，可以按照如下方法操作：

- 1) 打开“File”菜单，选择“Save Frameset As”（另存储框架集为）命令。
- 2) 这时会打开文件存储对话框，允许您选择存储路径和输入文件名称。
- 3) 选择路径并输入新框架集文档名称，再单击“Save”按钮，即可保存框架集文档本身。

### 11.3.2 存储框架文档

要存储框架集文档窗口中某个框架，可以按照如下方法进行操作：

- 1) 在框架集文档窗口中，单击要保存的框架窗格区域，将插入点放置到该窗格中。
- 2) 打开“File”菜单，选择“Save”命令，即可保存该框架文档。
- 3) 如果框架文档尚未存储，则会打开文件存储对话框，允许您选择存储路径和输入文件名称，选择路径并输入新框架文档名称，再单击“Save”按钮，即可保存该框架文档。
- 4) 如果已经保存过该框架文档，则该操作只是再次在原先基础上保存框架文档，不会再出现文件存储对话框。

### 11.3.3 存储框架集中所有的文档

如果框架集中包含多个框架，分别保存每个框架文档则会显得非常不便。有时候用户甚至会遗漏某个文档。

Dreamweaver 提供了一次保存框架集中所有框架文档的方法，以方便用户的操作。您可以按照如下方法进行操作：

- 1) 打开“File”菜单，选择“Save All”命令。
- 2) 如果有文档（包括框架集文档或框架文档）尚未被保存，则会出现文件存储对话框，

提示您选择存储路径和输入文件名称。选择路径并输入新框架文档名称，再单击“ Save ”按钮，即可保存这些文档。如果有多个文档尚未保存，则会依次出现多个文件存储对话框。

3) 如果所有的文档都已经被保存过，则 Dreamweaver 会直接在原先的基础上保存文档，不会再出现文件存储对话框。

注意 “ Save All ” 命令不仅可以用于保存所有框架集中的文档，事实上，它还适用于保存当前操作的所有文档，包括当前文档窗口中的文档，也包括位于另外文档窗口中的文档。例如您另外打开了一个 Dreamweaver 文档窗口，正编辑其他类型的文档，该 “ Save All ” 命令同样也会将之保存。

## 11.4 设置框架和框架集的属性

框架的属性确定了框架的名称、框架源文件、框架的空白边距、框架的滚动特性、框架的重设大小特性以及框架的边框特性等。

框架集的属性可以控制框架集的整体属性，例如框架的尺寸、框架的颜色、框架之间边框的宽度等。

在 Dreamweaver 中，通过属性面板和框架面板，可以完成绝大多数的属性设置操作。

### 11.4.1 认识框架面板

在设置框架属性时，框架面板是最有用的工具之一，打开 “ Window ” 菜单，选择 “ Frames ” 命令，或是按下 Ctrl+F10 键，即可显示框架面板，如图 11-12 所示。

我们看到，在框架面板中，显示当前框架集文档窗口中已经出现的框架窗格结构，同时，在不同的框架区域中，还显示该框架的名称。

每当我们在框架集文档窗口中构建新框架、删除某个现有框架，或是修改框架的尺寸或名称时，框架面板上显示的框架结构都会相应发生变化。

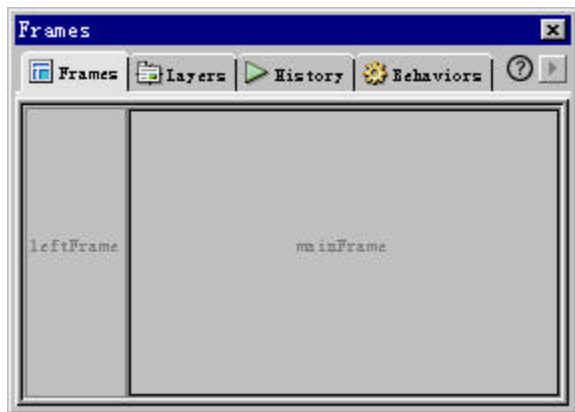


图11-12 框架面板

### 11.4.2 选中框架和框架集

您可以利用框架面板来选中框架或框架集，也可以直接从文档窗口中选中框架或框架集，

当然，利用框架面板是最方便的。

#### 1. 利用框架面板选中框架或框架集

要利用框架面板选中框架或框架集，可以采用如下方法：

要选中某个框架，可以直接在框架面板上单击对应的框架区域中任意位置。被选中的框架区域四周会出现黑色边框。

要选中框架集，可以单击框架面板四周的框架边框。这时整个框架面板区域四周显示粗黑边框。

图11-13显示了利用框架面板选中框架和框架集的情形。

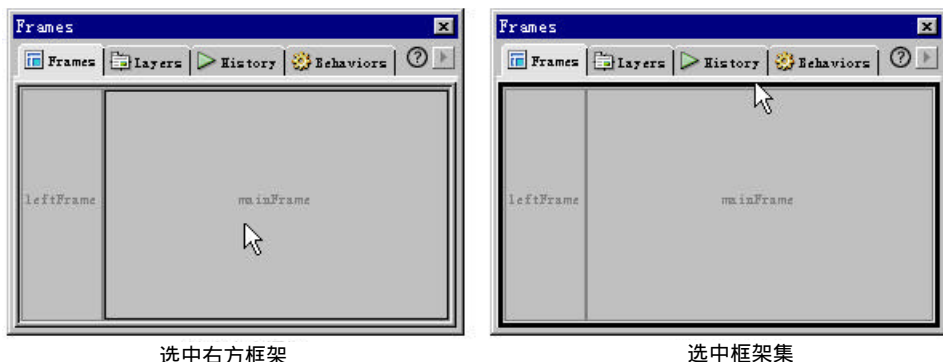


图11-13 利用框架面板选中框架和框架集

#### 2. 在文档窗口中选中框架或框架集

不利用框架面板，直接在文档窗口中，也可以选中框架或框架集。您可以按照如下方法进行操作：

在文档窗口中，按住Alt键，然后单击要选中的框架窗格中的任意位置，即可选中该框架。被选中的框架四周会出现虚线。

在文档窗口中直接单击框架边框，即可选中框架集。选中框架集时，所有的框架边框上都会显示虚线。

图11-14显示了在文档窗口中选中框架和框架集的情形。

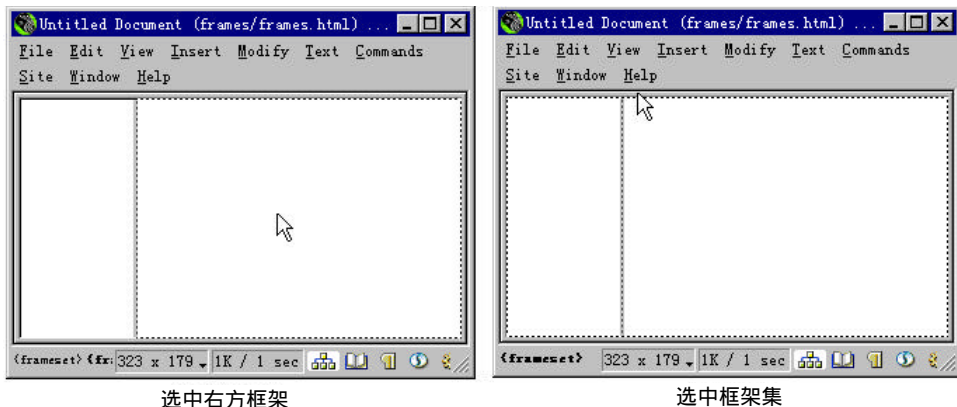


图11-14 在文档窗口中选中框架和框架集



### 3. 在选中一个框架的基础上选择其他框架

如果已经选中了某个框架，可以采用下面的方法，将选择从一个框架移动到另一个框架中：

要将选择从一个框架移动到另一个同一级别的框架中，可以按住 Alt 键，然后按下左右箭头键。图 11-15 显示了选中同级框架的情形，为了便于查看，我们用灰色表示选中状态。

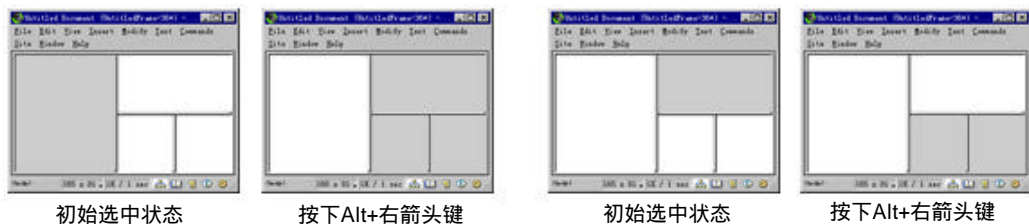


图11-15 选中同级框架

要将选择移动到当前框架的父级框架集中，可以按住 Alt 键，然后按下向上的箭头键。

图 11-16 显示了选中父级框架的情形，为了便于查看，我们用灰色表示选中状态。



图11-16 选中父级框架

要将选择移动到当前框架的子级框架中，可以按住 Alt 键，然后按下向下的箭头键。图 11-17 显示了选中子级框架的情形，为了便于查看，我们用灰色表示选中状态。



图11-17 选中子级框架

#### 11.4.3 设置框架属性

框架的属性确定了框架的名称、框架源文件、框架的空白边距、框架的滚动特性、框架的重设大小特性以及框架的边框特性等，利用框架的属性面板可以完成大多数的设置。

要设置框架属性，可以首先选中框架，然后从属性面板中设置框架属性。选中框架时的属性面板如图 11-18 所示。

Frame Name ( 框架名称 )    在该文本框中，可以输入框架的名称，它实际上设置的是

<frame>标记的name属性。该名称被JavaScript等程序引用，也可以同链接的target属性相结合，以指定链接的打开窗口和位置。



图11-18 框架的属性面板

框架的名称必须是一个单词，其中可以出现短下划线符号（\_），但是不能出现连字号（-）、句点（.）以及空格。框架的名称可以以字母开头，但是不能用数字开头。并且需要注意，不要使用JavaScript的保留关键字，例如top或navigator之类。

**Src（源）** 在该文本框中，可以设置该框架源文件的URL，您可以单击右方的文件夹窗口，然后从磁盘上选择框架文件。

该操作实际上设置的是<frame>标记的src属性。

另一种指定框架源的方式是首先在文档窗口中单击框架窗格，将插入点放入该窗格中，然后打开“File”菜单，选择“Open in Frame”（在框架中打开）命令。这时会打开文件操作对话框，提示您从磁盘上选择要放入该框架中的文件。

**Scroll（滚动条）** 在该下拉列表中，允许您设置该框架中出现滚动条的方式。该操作实际上设置的是<frame>标记的scrolling属性，可以有如下选择：

- Auto 选择该项，则将scrolling属性值设置为“Auto”，表明当框架文档内容超出了框架的大小时，会出现框架滚动条，允许通过拖动滚动条显示框架内容。
- Yes 选择该项，则将scrolling属性值设置为“YES”，表明无论框架文档中的内容是否超出框架的大小都会显示框架滚动条。
- No 选择该项，则将scrolling属性值设置为“NO”，表明即使框架文档中的内容超出了框架的大小，也不会出现框架滚动条。
- Default 选择该项，实际上是从<frame>标记中删除scrolling属性代码，大多数浏览器将之当作“Auto”方式。

**No Resize（不可重设大小）** 选中该复选框，则设置无法通过拖动框架的边框（也即框架和其他框架之间的分隔条）来改变框架大小，该操作实际上是为<frame>标记添加noresize属性。

如果清除该复选框，则可以通过拖动该框架同其他框架之间的分隔条来改变其大小。该操作实际上是从<frame>标记中删除noresize属性代码。

**注意** 这里设置的不可重设大小特性指的是在浏览器中浏览网页时的特性。在Dreamweaver的文档窗口中，您始终可以通过拖动框架边框来改变框架大小。

**Borders（边框）** 在该下拉列表中，可以控制当前框架的边框是否被显示。实际上该操作设置的是<frame>标记中的frameborder属性。可以设置如下的属性值：

- “Yes” 选择该项，则将frameborder属性值设置为“YES”，表明该框架与其四周的框架相邻处显示分隔条。
- “No” 选择该项，则将frameborder属性值设置为“NO”，表明该框架与其四周的框架

相邻处不显示分隔条。

- “Default” 选择该项，实际上是从<frame>标记中删除frameborder属性代码，大多数浏览器将之当作“ Yes ”方式。

注意 该选项会覆盖为框架集定义的边框设置。

只有在同当前框架相邻的所有框架的边框都设置为不显示时，框架间的边框才会真正不显示。

**Border Color (边框颜色)** 在该颜色框中，可以设置框架边框的颜色，您可以单击颜色按钮，打开颜色板，然后选择需要的颜色，也可以直接在文本框中输入十六进制颜色数值。

该操作实际上设置的是<frame>标记的bordercolor属性。

注意 该操作会覆盖对框架集中边框颜色属性的设置。

**Margin Width (边距宽度)** 该文本框中，可以设置当前框架左右方的空白边距，也即框架左右边框同框架内容之间的距离。该操作实际上设置的是 <frame>标记的marginwidth属性，其单位是像素。

**Margin Height (边距高度)** 该文本框中，可以设置当前框架上下方的空白边距，也即框架上下边框同框架内容之间的距离。该操作实际上设置的是 <frame>标记的marginheight属性，其单位是像素。

#### 11.4.4 设置框架集属性

框架集的属性可以控制框架集的整体属性，例如框架的尺寸、框架的颜色、框架之间边框的宽度等。

利用属性面板，可以很方便地设置大多数框架集属性。选中框架集时的属性面板如图 11-19所示。



图11-19 框架集的属性面板

**Borders** 在该下拉列表中，可以设置框架集中所有框架的边框是否被显示。该操作实际上是设置<frameset>标记的frameborder属性，可以有如下的值：

- “Yes” 选择该项，则将frameborder属性值设置为“ YES ”，表明该框架集中显示框架之间的分隔条。
- “No” 选择该项，则将frameborder属性值设置为“ NO ”，表明该框架集中不显示框架之间的分隔条。
- “Default” 选择该项，实际上是从<frameset>标记中删除frameborder属性代码，大多数浏览器将之当作“ Yes ”方式。

注意 对框架的属性设置会覆盖对框架集的属性设置，例如，如果您将某个框架集设置为无边框，但是又将其中的某个框架属性设置为有边框，则实际显示中该框架将显示边框。

**Border Color (边框颜色)** 在该颜色框中, 可以设置框架集中所有框架边框的颜色, 您可以单击颜色按钮, 打开颜色板, 然后选择需要的颜色, 也可以直接在文本框中输入十六进制颜色数值。

该操作实际上设置的是 `<frameset>` 标记的 `bordercolor` 属性。

**注意** 对框架的属性设置会覆盖对框架集的属性设置, 例如, 如果您将某个框架集边框颜色设置为红色, 但是又将其中的某个框架的边框颜色设置为蓝色, 则实际显示中该框架集中除了该框架之外所有的框架边框都显示为红色, 而该框架边框显示为蓝色。

**Border Width (边框宽度)** 在该文本框中, 可以设置框架集中所有框架的边框宽度。您可以在其中输入框架边框宽度数值, 单位是像素。输入 0 则可以不显示带有宽度的边框。

该操作实际上是为 `<frameset>` 标记设置 `border` 属性和 `framespacing` 属性。

利用预定义的框架集, 也即通过对象面板来创建框架集, 可以快速地构建带有在实际创作中经常使用的框架集属性的框架集, 所有的预定义框架集都带有如下属性: 没有边框、没有滚动条以及框架大小不可调节。

另外, 在框架集的属性面板中的 **Value (值)** 和 **Unit (单位)** 区域, 还可以设置框架集中不同框架的宽度或高度, 我们将在 11.4.5 一节介绍。

#### 11.4.5 设置框架大小

要设置框架集中框架的大小, 您可以采用多种方法。

方法一: 在文档窗口中, 利用鼠标拖动框架边框, 可以快速改变框架大小。

方法二: 利用属性面板, 也可以改变框架大小, 方法如下:

1) 选中框架集。

2) 单击属性面板右端的相应区域, 选中要设置大小的框架, 单击行头或列头可以选中相应的框架行或框架列, 单击框架区域中任意位置, 可以在选中该框架所在行或所在列两种状态之间切换, 如图 11-20 所示。

3) 在 “Value” 文本框中输入框架的大小数值, 在右方 “Unit” 下拉列表框中, 可以选择数值的单位, 可以有如下几种选择:

- **Pixels** 选中该项, 表明在 “Value” 文本框中输入的数值以像素作为单位。这种方式设置的框架大小是固定的, 无论如何改变浏览器窗口, 框架的大小都保持不变。它经常被应用于一些永远不希望改变框架大小的场合中, 例如导航条所在的框架文档。
- **Percent** 选中该项, 表明在 “Value” 文本框中输入的数值是当前框架同当前框架集大小的百分比数值。这种方式设置的框架大小是不固定的, 随着浏览器窗口大小的变化, 框架的大小也会发生变化。在源代码中, 该操作实际上在 `<frameset>` 标记的 `rows` 或 `cols` 属性值中的大小数值后添加百分比符号 (%)。
- **Relative (相对)** 选中该项, 表明在 “Value” 文本框中输入的数值是当前框架同其他框架之间的大小比例。在源代码中, 该操作实际上在 `<frameset>` 标记的 `rows` 或 `cols` 属性值中的大小数值后添加星号 (\*)。

**注意** 在设置框架大小时, 三种单位的优先级是不一样的, 其优先级顺序从高到低依次

为“Pixels”(像素)、“Percent”(百分比)、“Relative”(相对)。例如,如果将某个框架大小以像素作单位设置,然后又用其他的单位设置了其他框架的大小,则Dreamweaver会首先保留当前以像素作单位所设置的框架大小。



图11-20 利用属性面板选中框架集中的主框架

#### 11.4.6 设置框架和框架集的标题

文档的标题指的是在文档头部位于 `<title>` 和 `</title>` 标记之间的文字。在显示了框架集文档的窗口中,可以分别为每个框架文档设置标题,也可以为框架集文档本身设置标题。

要设置框架标题或框架集标题,可以采用如下方法:

- 1) 要设置框架文档的标题,可以将插入点放置到要设置标题的框架窗格中,或是选中该框架;要设置框架集标题,则需要选中整个框架集。
- 2) 打开“Modify”菜单,选择“Page Properties”命令,打开页面设置对话框。
- 3) 在“Title”(标题)文本框中,设置相应的标题名称。
- 4) 按下“OK”按钮,确定操作,即可设置相应框架文档或框架集文档的标题。

注意 在浏览框架集文档时,浏览器标题栏上显示的是框架集文档的标题,尽管每个框架文档都有其自己的标题,但它们并不显示在浏览器的标题栏上,设置框架文档的标题主要是为了便于管理。当然如果单纯浏览框架文档,则浏览器标题栏上仍会显示框架文档标题。

另外,不要将框架的标题同框架的名称相混淆,框架的名称是框架对应 `<frame>` 标记的 `name` 属性值,而框架的标题只是 `<title>` 和 `</title>` 标记间的文字。

从这里您也可以看出,在框架集文档窗口中,可以设置各框架文档的属性,也可以编辑个框架文档中的内容。只需要将插入点放置入该框架窗格内,然后按照常规的方法进行编辑或设置即可。

#### 11.5 在框架中使用链接

在使用了框架技术的网页中,通常存在大量的链接,大多数人之所以使用框架技术,就是为了利用框架来实现在站点文档间的导航。然而在框架中,对链接的控制较为不易。

例如,我们想象这样的情形:从一个在整个浏览器窗口中显示的首页页面中单击某个链接,进入带有框架的文档中,然后单击位于某个框架中的链接,返回首页,这时会出现什么情况呢?相信很多用户在初次使用框架时,都会遇到这样的情况:在返回首页时,不是以整



个浏览器窗口显示首页，而是在某个框架窗格中显示首页，如图 11-21 所示。

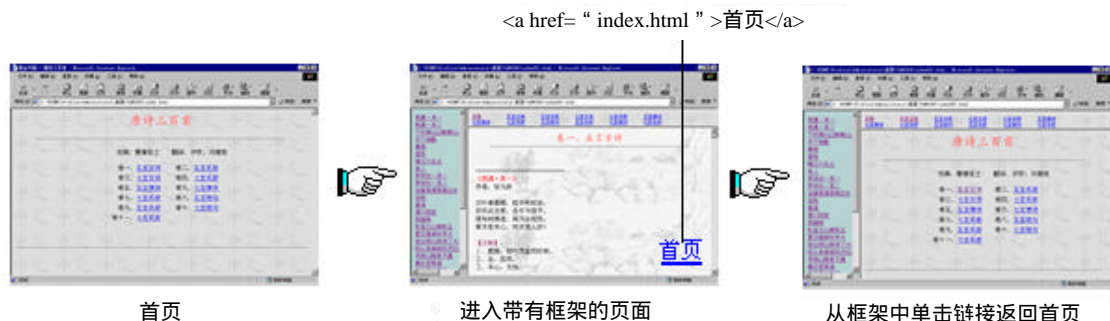


图11-21 框架中链接容易导致错误

对链接的控制变得不易，甚至容易导致混乱，这是框架技术最具争议的地方。然而，随着HTML的发展，目前已经提供了多种方法来控制框架中的链接行为，其中设置链接的 target 属性是最简单，也是最有效的方法。

我们曾经介绍过超链接标记，也即锚标记 <a>，该标记带有 target 属性。通过设置 target 标记，我们就可以控制单击链接后，目标锚端的文档在哪个框架窗格中被打开，如图 11-22 所示。

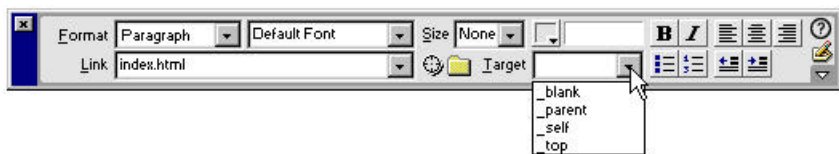


图11-22 从属性面板上设置链接的target属性

从如图 11-22 所示的属性面板上，我们可以看到，在 target 属性的属性值中，有如下 4 个预设的值：

- **\_blank** 如果将 target 属性值设置为 “\_blank”，则将开启一个新的浏览器窗口，并载入链接目标锚端的文档。
- **\_parent** 如果将 target 属性设置为 “\_parent”，则会在链接所在框架窗格的父级框架或窗口中，显示链接目标锚端的文档。
- **\_self** 如果将 target 属性值设置为 “\_self”，则会在链接所在的框架窗格中显示链接目标锚端的文档。
- **\_top** 如果将 target 属性设置为 “\_top”，则会以当前整个浏览器窗口显示链接目标锚端的文档内容。原先浏览器窗口中的框架和窗格都将消失。

现在，我们很容易解决图 11-21 所示的问题了，只要将链接的 target 属性设置为 “\_top”，就可以在返回首页时，以整个浏览器窗口显示，如图 11-23 所示。

不仅如此，通过设置链接的 target 属性值，可以任意控制链接的目标锚端文档在哪个框架窗格中被打开。您可以直接将链接的 target 属性设置为某个框架窗格的名称（也即该框架 <frame> 标记的 name 属性），这样单击链接时，链接目标锚端的文档将在具有指定名称的框架窗格中被打开。



```
<a href= " index.html " target= " _top " >首页</a>
```



首面



进入带有框架的页面



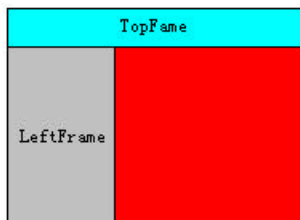
从框架中单击链接返回首页

图11-23 指定链接的target属性

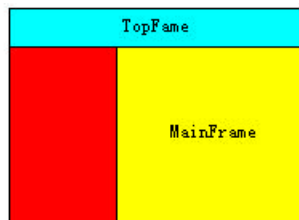
如图11-24所示，假设我们的文档中包含 3 个框架，其名称分别是 MainFrame、LeftFrame 和 TopFrame，请参看左方的图。无论链接位于哪个框架文档中，只要将 target 的属性值具体指定为某个框架的名称，就可在指定框架窗格中打开链接目标锚端的文档。在图 11-24 中，中间和右边的图说明了这种特性，其中红色区域表明设置相应 target 属性后目标锚端文档打开的位置。



原始文档



target= " MainFrame "



target= " LeftFrame "

图11-24 用框架名称指定target属性