

第 5 章 网站设计基础



本章从网站的基本语言 HTML 入手，介绍层叠样式表 CSS 和可扩展的标识语言 XML，包括它们的应用和应用技巧等；简单介绍 Visual Studio 2005 的使用；最后介绍了脚本语言 VBScript 和 JavaScript 的应用，以及使用它们的一些技术和技巧，并结合这些知识的应用，给出较典型的实例代表。



- 超文本标识语言 HTML
- 层叠样式表 CSS
- 可扩展标识语言 XML
- Visual Studio 2005 的基本使用
- 脚本语言 VBScript 应用
- 脚本语言 JavaScript 应用

5.1 HTML 标识语言

网站的开发不是仅仅设计、制作网页，如果只是制作网页的话，我们完全可以使用所见即所得的网页制作软件，如 FrontPage、Dreamweaver 等，而不需要掌握超文本标识语言（Hyper Text Markup Language, HTML）、脚本语言、动态服务器页（Active Server Pages, ASP）等 Web 程序设计技术。虽说 FrontPage、Dreamweaver 等均有网站开发功能，但与语言相比其灵活性就差多了，另外它们生成的程序源代码要复杂、冗长得多。所谓的 Web 程序设计包括超文本标识语言、脚本语言 VBScript 和 JavaScript、ASP、Perl/CGI 等。超文本标识语言、脚本语言是网页的基础，是网站生长的“土壤”，加上层叠样式表 CSS 以及可扩展标识语言 XML 等，可以使开发出的网站形式更丰富、内容更精彩。ASP 只能运行在服务器端，而超文本标识语言、脚本语言既可以运行在服务器端，又可以运行在客户端。

5.1.1 认识 HTML 标识语言

超文本标识语言 HTML 结构简单，语法较松散，易学易用。它是一种文字处理语言，而不是“程序”语言。HTML 的标识嵌于文本格式的文档中，用来控制文字、图片等在浏览器中的表现，以及如何建立文件之间的链接。程序与之最大的不同，就是可用来控制操作系统或应用程序执行并完成某项作业。使用超文本标识语言“处理”过的文档应该满足，不管在

任何操作系统任何浏览器上读起来或看上去都是一样的。

超文本标识语言最初用于发布信息，并没有在网络上使用。但由于它极容易使用的显著特点，随着网络带宽的逐渐增加，人们将其作为网络上发布信息的首选语言。超文本标识语言的领头先锋是美国麻省理工学院（Massachusetts Institute of Technology, MIT）计算机科学实验室的 WWW 标准化组织（World Wide Web Consortium, 又称 W3C），其互联网地址为 <http://www.w3c.org>。有关 HTML 语言的各种参考资料或 Web 的相关技术资料，以及 W3C 行将发布的各种新版特征、最新消息等，均可以在该网站上查到。

用超文本标识语言书写的源程序简单清晰，如例 5.1 是一个典型的 HTML 源程序。

例 5.1 典型的 HTML 源程序（L5-1.htm）。

```
<HTML>
<HEAD><TITLE>HTML 源程序</TITLE></HEAD>
<BODY>
  <H2>杭州碧波亭对联欣赏</H2>
  <OL><LI>三千里外一条水</LI><BR><BR>
    <LI>十二时中两度潮</LI>
  </OL>
  <MARQUEE bgcolor="#FFCEA2" scrollamount="1" width="80%">三千里外一条水</MARQUEE>
  <MARQUEE bgcolor="#2FF0FD" behavior="alternate">十二时中两度潮</MARQUEE>
</BODY>
</HTML>
```

说明：打开“记事本”写入例 5.1 的源程序，保存时扩展名必须是.htm，打开浏览器运行本例，或直接双击该文件，运行结果见图 5.1。<H2></H2>是标题文字标识；是有序列表标识；是列表项标识；
是换行标识；<MARQUEE>是移动文字的标识。位于<MARQUEE></MARQUEE>标识之间的文字将会移动，其他标识或属性的含义请参阅相关书籍。

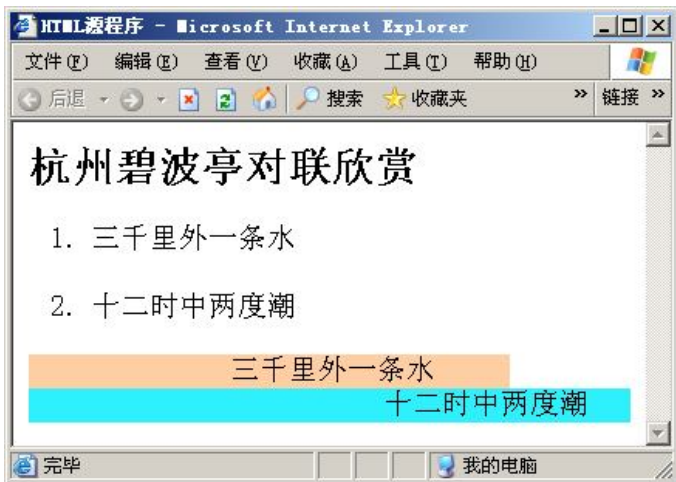


图 5.1 浏览器中的例 5.1

标准的 HTML 文件一般分为三个部分：序（Prologue）、头部（Head）和文件主体（Body）。HTML 文件的主要部分是头部和文件主体。除了头标识和主体标识外，就是 HTML 的

附加标识——序（HTML 的版本声明），这种标识放在 HTML 文档的第一行，一般的 HTML 文档中通常可以不写“序”。序是用来告知浏览器所遵循的 HTML 版本。当前的浏览器一般不要求含有序，而且也不会处理序，因此该标识可以忽略。若想保持编写严格遵循标准通用标识语言（Standard Generalized Markup Language, SGML）的 HTML 文档，应该保留序的位置。

HTML 文件的基本结构如下：

	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">	<! 序>
<HTML>		<! HTML 文档起始标识>
<HEAD>		<! HTML 的头信息部分>
<TITLE>		<! HTML 的标题部分>
	HTML 文件的基本结构	
</TITLE>		<! 标题结束标识>
</HEAD>		<! 头信息结束标识>
<BODY>		<! 这里是 HTML 的正文部分>
	HTML 文件的基本结构样板，所有正文在这里书写。	
</BODY>		<! 正文部分结束标识>
</HTML>		<! HTML 文档结束标识>

说明：

- (1) HEAD 标识内的信息一般不在浏览器的主窗口中显示。
- (2) 需要表示的内容和描述内容的标识格式等，一般放在 BODY 标识内。
- (3) 注释有两种方法。
 - <!-- 注释内容 -->，“注释内容”中不可出现右尖括号，一般用于标识里的内容。
 - <!-- 注释内容 -->，“注释内容”中可以出现包括尖括号在内的任何括号，一般用于注释大段的内容。注意，本注释仅用于注释超文本标识语言。
- (4) 可以使用所有编辑纯文本文件的软件，或带有 HTML 编辑器的软件来编辑 HTML，扩展名必须写为.html 或.htm。如记事本、写字板、Word、FrontPage、Hotdog、Dreamweaver 等。
- (5) “HTML 文件的基本结构”在浏览器中的显示效果见图 5.2。

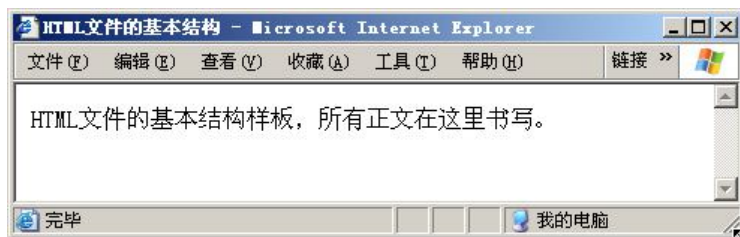


图 5.2 用浏览器打开的“HTML 文件的基本结构”文件

超文本标识语言 HTML 是标识的集合，是标准通用标识语言 SGML 的一个子集。被一对尖括号“<”和“>”括起来，标识一般成对出现。如“基本结构”中的<HEAD>和</HEAD>，<HEAD>为起始标识，加斜线的<HEAD>即</HEAD>为结束标识，称之为双标识；
没有结束标识，称为单标识。各种标识与不同网页内容的有机合成就构成了多姿多彩的互联网世界。标识（Tags）和属性（Attributes）构建了 HTML 的语法。浏览器只要读到 HTML 的标识



图 5.3 提交表单页

例 5.4 HTML 综合应用 (L5-4.htm), 如图 5-4 所示。

```
<HTML>
<HEAD><TITLE>HTML 综合应用</TITLE></HEAD>
<BODY bgcolor="#ffffff" text="#000000" link="#0000ff" alink="#ff00ff" vlink="#ff0000"
background="背景图.jpg">
  <A href="http://www.lhx.com/" target="resource window">绿海星</A>
  <A href="http://www.hemc.edu.cn">学院</A>
  <A href="http://cn.yahoo.com">雅虎中国</A>
  <CENTER>
    <H2>黄河颂</H2><HR>          <!-- <HR>显示一条水平直线 -->
    <EMBED src="黄河颂.mp3" autostar="TRUE" hidden="TRUE" loop="TRUE">
    <BGSOUND src="黄河颂.mp3" loop="INFINITE">
  </CENTER>
  <IMG src="车展.jpg" alt="欢迎来到车博会" align="right" border="0" hspace="420" vspace="22"
height="118" width="200">
</BODY>
</HTML>
```



图 5.4 HTML 综合应用

说明: BODY 标识是网页制作中主要的组成部分之一,对于网页的背景或背景图片的设置以及超链接时字体的变化均应放在 BODY 中。bgcolor 为背景颜色设置属性,“#FFFFFF”是十六进制数,格式为“#RRGGBB”——红红绿绿蓝蓝。取值范围在 0~F 之间,该值也可直接用英文的 red、green、blue、yellow、white 等词设置(请参阅附录 A)。一般来说设置背景图片,也同时设置背景色。这样在图片尚未下载完毕之前使屏幕背景不至于太单调。link——设定超链接的颜色,alink——设定单击超链接时的颜色,vlink——设定单击超链接以后(已访问过)的颜色。background 取背景图片,图片的格式一般为.GIF 和.JPG。超链接标识<A>的 target 属性设为“resource window”,表示单击链接时,浏览器打开一个新视窗显示网页的内容。同样,如果 target 的属性值设置成“window_name”或“_blank”也有相同的效果,请读者自己试一试。一般主页中常放上<EMBED>和<BGSOUND>两个标识,以防不同的浏览器在解释网页时,无法读出背景音乐。<BGSOUND>标识播放背景音乐时,窗口最小化会使背景音乐停止;而<EMBED>标识则不会出现这种情况。这就产生了一个问题,当最小化窗口后,又打开时,会产生二重音乐。<EMBED>中的属性 autostar、hidden、loop 分别表示自动启动播放、隐藏播放器、循环播放,它们均有两个值 TRUE 和 FALSE,<BGSOUND>中的 loop 属性可取值 INFINITE、TRUE 和 FALSE,分别表示无止尽播放、循环播放和播放一次。通常背景音乐的格式是.WAV 和.MID,也可以播放.MP3 等,如本例。标识中的 align 属性用 top、middle、bottom、left、right 五种位置放置图片。hspace 和 vspace 分别设置图片距屏幕左上角的水平和垂直距离(以像素为单位)。height 和 width 分别设置图片的高和宽,以满足主页对图片尺寸的要求。也可使用 lowsrc 属性替代 src 属性,要求浏览器在画面载入以前先装载低分辨率的图像,以加快访问网页的速度。语句中同时存在 align 属性和 hspace、vspace 属性时,则按 hspace、vspace 属性的值显示图片。

例 5.5 汽贸网站中的 HTML 应用 (L5-5.htm)。

```
<HTML>
<HEAD><TITLE>汽贸网站中的 HTML 应用</TITLE></HEAD>
<BODY>
<TABLE bgcolor="#FFCCDD" border="0" width="800" id="table1" height="445" cellspacing="0"
cellpadding="0">
  <TR>
    <TD height="20" valign="bottom">
      <TABLE border="0" bgcolor="#9900FF" id="table1" width="803" cellspacing="0"
        cellpadding="0" height="23">
      <TR>
        <TD width="382"></TD>
        <TD width="48"><FONT color="#FFFFFF"><A href="http://www.lhxqm.com/index.asp" onclick=
"this.style.behavior=url(#default#homepage); this.sethomepage('http://www.lhxqm.com/');" class="yt">设为
首页</A></FONT></TD>
        <TD width="48"><FONT color="#FFFFFF"><A href="javascript:window.external.Addfavorite
('http://www.lhxqm.com/', '绿海星汽贸公司网站')" target="_self" class="yt">收藏本页</A></FONT></TD>
        <TD width="48"><A href="javascript:window.print()" class="yt">打印本页</A></TD>
        <TD width="48"><A href="mailto:lxh@163.com" class="yt">联系我们</A></TD>
        <TD width="48"><A href="mailto:lxh@163.com" class="yt">在线订购</A></TD>
        <TD width="48"><A href="http://www.lhxqm.com/default.asp" class="yt">网站管理</A></TD>
```

```

</TR>
</TABLE>
</BODY>
</HTML>

```

说明：本例是截取“汽贸”网站 index.asp 文档中的一段代码。可以看出，此处缺乏多个配置文件，所以本例显示时并不到位，不美观，也不实用，在此仅仅举例而已。

5.2 层叠样式表 CSS

通过超文本标识语言 HTML，可以将网页的内容布置得有声有色，但有时我们仍感不足。层叠样式表（Cascading Style Sheets, CSS）正好弥补了 HTML 标识在组织页面内容上的“缺陷”，它主要用来定义网页字符数据的处理，包括字符的格式化、显示、编排、打印和特殊效果等。CSS 由 WWW 标准化组织，又称互联网联盟（World Wide Web Consortium, W3C）制定。一些有关 CSS 的资料以及与 Internet 相关的资料、标准、规则、建议、推荐等，读者可以通过 W3C 的网站 <http://www.w3c.org> 查询。

5.2.1 CSS 样例与结构

层叠样式表的文件格式分为两部分：选择器（Selector）和样式规则（Rule），由相应的格式代码文本文件组成，与 HTML 不一样，它比 HTML 代码复杂，表现网页的形式多样化，可以更加精确地描述页面内容。从形式上看，CSS 的语法格式非常类似 C 语言中自定义的一个函数。如：

```

BODY{
    font-family: "隶书";           /* 定义字体 */
    font-size="32px";             /* 定义字体尺寸，32 点 */
    color: "#00FF00";             /* 定义字体颜色，绿色 */
    letter-spacing:"8pt";          /* 定义字间距，8 磅 */
}

```

在上面这段样式表文件中，BODY 是选择器，大括号内为样式规则。样式规则用于指定样式要套用的对象，一般来说指的是 HTML 标识。样式规则用于设定样式内容，由属性（如 font-family）和属性值（如隶书）组合而成。属性指定标识所具有的特征，如字体、大小、颜色等；属性值则指定对应属性的取值，如隶书、16pt、#FF00FF 等。分号是多重属性之间的间隔符，上例中的样式规则采用分行书写（也可以写在同一行），这样更具可读性。注释符与 C 语言相同，也是“/* */”。要注意的是 IE 浏览器和 Netscape 浏览器对 CSS 的支持可能会有所不同，使用时应注意它们的差别。样式表中的单位有：pc 十二磅活字、pt 磅、px 像素（或称点，默认值也是像素）、em 以当前字体的小写字母 m 的高度为单位高度、ex 以当前字体的大写字母 X 的高度为单位高度、in 英寸、cm 厘米、mm 毫米。

5.2.2 CSS 应用

CSS 的使用比较灵活，在 HTML 中主要有三种使用方法（有关层叠样式表的其他内容，请读者参阅 CSS 的相关书籍，也可以上 <http://www.w3c.org> 查询）。

- 直接在 HTML 语句中使用。
- 嵌入 HTML 文档中使用。
- 使用独立的文件。

例 5.6 CSS 直接和嵌入 HTML 中的应用 (L5-6.htm)。

```
<HTML>
<HEAD>
  <TITLE>CSS 直接和嵌入 HTML 中的应用</TITLE>
  <STYLE>
    .clSpan1 {background-color:red;color:white;}      /* 以下是 CSS 嵌入应用 */
    .clSpan2 {background-color:yellow;color:black;}
    .clSpan3 {background-color:blue;color:white;}
    .clSpan4 {background-color:lime;color:black;}
    .clDiv1 {background-color:yellow;color:black;}
    .clDiv2 {background-color:magenta;color:white;}
    .clDiv3 {background-color:cyan;color:black;}
    .clDiv4 {background-color:black;color:white;}    /* 至此句结束 */
  </STYLE>
</HEAD>
<BODY>      <!-- 比较下面两个超链接, 第 1 个是直接使用 CSS, 第 2 个是普通超链接 -->
  <A href="L5-21.htm" style=text-decoration:none;font-size:40>您知道它的下联吗? </A>
    <FONT size=2><A href="L5-21.htm ">(这是一幅拆字联, 答案在本书内找)</A></FONT><P>
    <SPAN class="clSpan1">冰凉酒</SPAN>      <!-- 使用 CSS 规则, 下同 -->
    <SPAN class="clSpan2">一点</SPAN>
    <SPAN class="clSpan3">二点</SPAN>
    <SPAN class="clSpan4">三点</SPAN>
  <HR>
    <DIV class="clDiv1">冰凉酒</DIV>      <!-- 使用 CSS 规则, 下同 -->
    <DIV class="clDiv2">一点</DIV>
    <DIV class="clDiv3">二点</DIV>
    <DIV class="clDiv4">三点</DIV>
</BODY>
</HTML>
```

说明: 为“行内元素 (Inline Element)”, <DIV></DIV>称作“块级元素 (Block Level Element)”。我们在代码中输入了 4 组词, 词与词之间既无空格, 也无换行。然后, 将各词分别配上或<DIV></DIV>标识, 并配置样式规则。从图 5.5 中可以看出, 加标识的前 4 组词, 仍然在一行显示, 没有因为的插入而破坏在一行显示的结构。所以, 用户可使用任意改变同一行范围内的显示样式, 例如颜色、背景、字体等。后 4 组词配上<DIV>标识, 在源文件中未加分行标识, 但是从图 5.5 中可以看出, 浏览器给它们各自分配了一个新的独立区域, 形成各自的“块”。代码中的第 1 个超链接是直接使用 CSS 样式规则, 使之取消了超链接的下划线, 并加大了字体。

若在源文件中仅插入一个或<DIV>, 则什么都不会发生, 直到用户为它们定义样式, 才能起应有的作用。因此, 和<DIV>又被称作“空元素”。和<DIV>的引入, 更加增强了页面设计的灵活性。其实, HTML 标识基本上都可以分成这两类。如<SUP>、

、和<INPUT>等是典型的“行内元素”，而<H1>、<P>、<HR>和<FORM>等就属于“块级元素”。

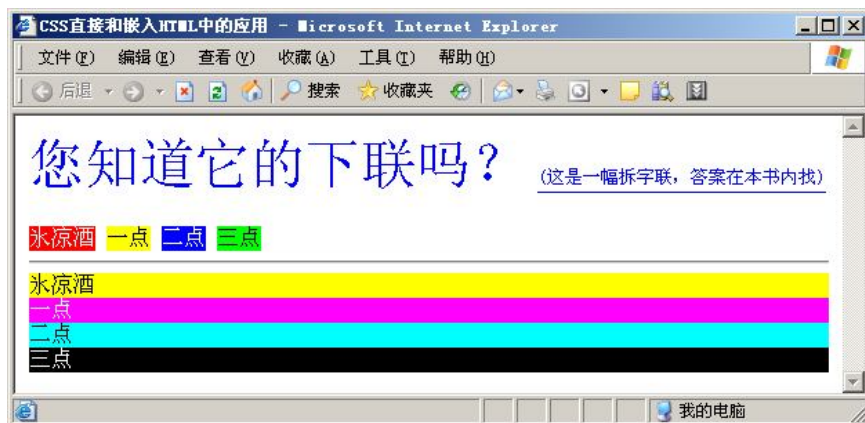


图 5.5 直接和嵌入使用 CSS

例 5.7 建立一个扩展名为.css 的独立文本文件，用以定义 CSS 格式。采用链接或导入两种方法调用此独立文件，使用 CSS 规则（L5-7.htm）。

```
<HTML>
<HEAD>
<TITLE>链接和导入样式表</TITLE>
  <LINK rel="StyleSheet" href="DeCss.css" type="text/css">    <!--链接样式表-->
</HEAD>
<BODY>
  江流天地外，山色有无中。<BR>
  此处采用链接样式表方式，所定义的字体为隶书、大小 32 点、间距 8 磅、颜色为绿色。
</BODY>
<STYLE>
  <!--                      /* 为不能识别 CSS 的浏览器而设 */
    @import url("DeCss.css");          /* 导入样式表 */
  -->
</STYLE>
<BODY>
  <P>郡邑浮前浦，波澜动远空。<BR>
  此处采用导入样式表方式，调用了同一个文件，所以字体、字号、间距、颜色相同。
</BODY>
</HTML>
```

说明：本例所调用的样式规则就是 5.2.1 节下的那段代码，将其命名为 DeCss.css，放在与本例相同的目录下即可。

例 5.8 使用 CSS 设置滚动条的颜色、阴影、立体样式等（L5-8.htm）。

```
<STYLE>
<!--
  BODY{                      /* 设置滚动条 */
    scrollbar-3dlight-color:'ff0000';          /*滚动条上边和左边的边沿颜色*/
```




图 5.6 标识与鼠标样式

表 5.1 CSS 鼠标样式指定及图例

指定方法	演示效果图例	说明
cursor:auto;	(无)	这是默认的属性值，即鼠标根据页面不同的元素对象呈现不同的样式。例如超链接为手形，文字和输入框为“I”形等
cursor:default;		鼠标在任何地方都呈现普通的箭头形状
cursor:hand;		鼠标将呈现手的形状，同超链接的鼠标形状
cursor:crosshair;		在绘图时遇到过的精确定位的十字形鼠标
cursor:text;		“I”字形的鼠标，平时在文本上和输入框上可以见到
cursor:move;		鼠标由水平和垂直双箭头组成，移动 Windows 窗口时常见
cursor:wait;		鼠标呈现两个倒三角沙漏的形状，平时见于 Windows 程序忙的时候
cursor:help;		鼠标样式为一个普通箭头加一个问号，在 Windows 窗口右上角单击“？”按钮后，也可以看到这种鼠标样式
cursor:n-resize;		鼠标箭头朝上方
cursor:s-resize;		鼠标箭头朝下方
cursor:e-resize;		鼠标箭头朝右方
cursor:w-resize;		鼠标箭头朝左方
cursor:ne-resize;		鼠标箭头朝右上方
cursor:nw-resize;		鼠标箭头朝左上方
cursor:se-resize;		鼠标箭头朝右下方
cursor:sw-resize;		鼠标箭头朝左下方

5.3 可扩展标识语言 XML

5.3.1 关于 XML

XML (eXtensible Markup Language) 的可扩展性是 XML 区别其他标识语言的最基本特

征。XML 的核心在于以一种标准化的方式建立数据表示的结构,而将具体标识的定义留给用户。XML 的可扩展性使 XML 可以满足各种不同领域数据描述的需要(见例 5.10 的源代码),并可以对计算机之间交换的任何数据进行编码。尽管 XML 的灵活性使得用户可以快速、容易地描述任意的内容,但由于 XML 本身并不能解释它标识的含义,而浏览器只能解释 HTML,所以大多数应用要求 XML 的标识集合能够满足某种标准或者双方的约定,这样才可能由“第三者”将其“翻译”成 HTML,最终由浏览器呈现。

万维网联盟 W3C 对 XML 做了很好的说明,XML 简化了定义文件类型的过程,简化了编程和处理 SGML 文件的过程,简化了在 Web 上的传送和共享;XML 可以广泛地运用于 Web 的任何地方,可以满足网络应用的需求;使用 XML 将使编程更加简单,更清晰和更有利于阅读理解的代码,将有助于代码的学习和创建。

XML 有四大主要特点:良好的数据存储格式、可扩展性、高度结构化和便于网络传输。凭借这些优良的特性,XML 已经被业界倡导作为电子商务的通用描述语言。XML 提供了丰富的语法来存储、携带和交换数据,用户能方便地在独立的平台之间交换数据和针对特定应用定义用户自己的标识语言。

XML 不是 HTML 的替代产品,也不是对 HTML 的升级,XML 只是对 HTML 进行了补充,扩展了 HTML 的更多功能,在以后较长的一段时间里我们仍然有可能继续使用 HTML。与 HTML 不同,HTML 是一种显示输出标识语言,只显示信息,不描述信息;XML 是一种描述信息的标识语言,它不涉及任何显示信息的处理,而是使用其他语言来完成这一任务。由于 XML 结构清晰、开放性强,使得它荣获了新一代“世界语”的美称,其扩展名是.xml。

担当 XML 显示输出任务语言之一的是 XSLT (Extensible StyleSheet Language Transformations,可扩展样式表语言转换),它可以将 XML 转化成 HTML,可以过滤、选择和格式化 XML 的数据,该文件的扩展名是.xml。

HTML 和 XML 的最大区别是,HTML 用规定的标识来描述、显示网页内容,是一个定型的标识语言。如<HR>表示一条水平分隔线,有固定的外观、尺寸。XML 则没有规定的标识,用户可以自由定义,XML 不能描述网页具体的外观、内容等,只是描述内容的数据形式和结构。所以不能像使用 HTML 一样,直接用 XML 编写网页程序。无论该程序中是否包含了 XML 数据,它始终要被转换成 HTML 格式才能在浏览器上显示。

HTML 和 XML 的本质区别是,HTML 的显示内容和显示混在一起,共存于同一个文件;XML 则将显示内容和显示分开(可以在同一文件,也可以不在同一文件),代码结构清晰。如例 5.11 给出的是数据(显示内容),而例 5.12 给出的是显示方式。我们仅需改变显示方式,就可以改变页面的排版样式,而数据无需做任何改动。

可以说 XML 是一种标识语言,但更确切地说它是一种创建标识语言的元语言。从例 5.10 中可以看到,这里的所有标识都是用户自己定义的。一个 XML 元素由起始标识和结束标识,以及标识之间的数据构成。起始和结束标识用来描述标识之间的数据。标识之间的数据被认为是元素的值。例如:

```
<商标>北京</商标>
```

XML 元素“<商标>”的值是“北京”。元素名(“商标”)允许把“北京”这个值标出来,这样可以把类似的数据区分开。例如,有可能另一个元素“<产地>”的值也叫“北京”。

```
<产地>北京</产地>
```

由于每个元素都有不同的标识名，所以可以很容易地区分上面两个元素的值。一个基本的 XML 文档就是一个 XML 元素，它可以嵌套 XML 元素。

例 5.10 一个普通的 XML 文档结构。

```
<汽车类>
  <轿车>
    <品牌>北京</品牌>
    <系列>老式吉普</系列>
    <型号>北京吉普 213</型号>
    <产地>中国北京</产地>
    <价格>2.1</价格>
    <出厂日期>1995.12</出厂日期>
  </轿车>
</汽车类>
```

说明：本例中 XML 元素“汽车类”就是一个有效的 XML 文档。以前在 HTML 中只能使用标准的标识，如<HEAD>、<TITLE>、<BODY>等，现在可以使用一个任意的标识名，如本例的<轿车>、<品牌>等。要在浏览器中直接打开本例是不可以的，必须在本例的首行前加上<?xml version="1.0" encoding="GB2312" ?>。此时，虽说可以运行本例（见图 5.7），但它仍是一个 XML 文档结构，并不是我们要显示的真实内容。因为 XML 只是描述性语言，要想在浏览器中显示它的“真实”内容，还要学习后面的知识。



图 5.7 浏览器中的 XML 文档

XML 是用来传递数据的，这些数据是规范的树型结构，XML 提供了对这种结构的操作方法。构建一个基本的 XML 文档有三点必须要注意：

- 所有元素必须有一个与之匹配的结束标识。
- 所有元素的嵌套必须正确，不允许交错。
- 所有属性值必须加引号。

还应当注意，XML 对字母的大、小写敏感，一些在 HTML 中可以“混”过去的语句，在 XML 中将受到严格的限定。XML 需要使用 IE 5.0 以上版本的浏览器打开。

例 5.11 标准的 XML 文档结构, 运行结果见图 5.10 (L5-11.xml)。

```
01:<?xml version="1.0" encoding="GB2312" ?>           <!-- 声明语句 -->
02:<?xml-stylesheet type="text/xsl" href="L5-12.xsl"?> <!--声明并指定用样式表转换 XML 文档-->
03:<二手车>
04:  <轿车>
05:    <品牌>北京现代</品牌>
06:    <系列>领翔</系列>
07:    <型号>2.0GL MT</型号>
08:    <产地>北京</产地>
09:    <总价-万元>12.5</总价-万元>
10:    <出厂日期>2008.10</出厂日期>
11:  </轿车>
12:  <轿车>
13:    <品牌>一汽大众</品牌>
14:    <系列>新宝来</系列>
15:    <型号>1.6L 自动豪华</型号>
16:    <产地>长春</产地>
17:    <总价-万元>10.7</总价-万元>
18:    <出厂日期>2008.9</出厂日期>
19:  </轿车>
20:</二手车>
```

说明: 本例是一个标准的 XML 文档, 但必须配合例 5.12 才能显示描述数据。行号仅为方便说明而设, 不是程序的组成部分。第 1 行告诉处理程序这里是 XML 文档, 其中 version 表示遵守的 XML 规范版本, encoding 表示文档所用的语言编码, 本例使用的是简体中文 GB2312。第 2 行告诉浏览器按 L5-12.xsl 指定的样式表将 XML 文档转换为 HTML 文档。

注意: 只要使用汉字作标识, 如 “<二手车>”, 则必须使用属性 encoding="GB2312"。

5.3.2 关于 XSLT

XSLT 语言由 W3C 定义, 该语言的 1.0 版本在 1999 年 11 月 16 日作为“推荐书”发布。XSLT 的最初目的是将信息内容与 Web 显示分离。如其最初的定义那样, HTML 通过按抽象概念 (如段落、重点和编号列表) 定义显示来实现设备独立性。

W3C 定义了两个系列的样式表标准。其一是在 HTML 中广泛使用的 CSS 语言, 它也可以在 XML 中使用; 其二就是功能强大的样式表语言 XSL。虽然 CSS 应用广泛, 但是它不能执行计算、重新整理或排序数据、组合多个源码中的数据或根据用户或会话的特征个性化显示的内容。针对 CSS 语言的不足, XSL 采纳了 SGML (Standard Generalized Markup Language) 社区中开发的 DSSSL (Document Style Semantics and Specification Language, 文档样式、语义和规范语言) 中一些好的构思。

从 XSL 系列的“家族”关系 (参见图 5.8) 上可以看出, XSL 在转换 XML 文档时分为明显的两个过程, 一是转换文档结构; 二是将文档格式化输出。这两个过程可以分开并单独处理, XSL 在发展过程中逐渐分裂为 XSLT (结构转换) 和 XSL-FO (formatting objects, 格式化输出) 两种分支语言, 其中 XSL-FO 的作用类似于 CSS 在 HTML 中的作用。这里介绍的只是第一个过程 XSLT。实际上, 现在人们常常使用 XSL 将 XML 文档转换成 HTML, 并使

用 HTML 浏览器作为格式化引擎。因为 HTML 实际上只是 XML 词汇表的一个示例，而 XSLT 可以使用任何 XML 词汇表作为其目标。

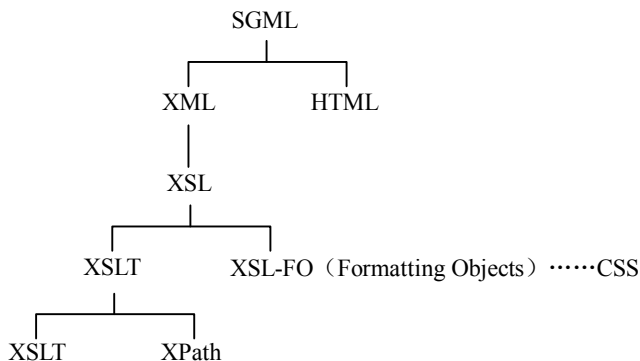


图 5.8 XSL 的“家谱”

XSLT 实际上也是一个 XML 文档。通过使用 XML 的尖括号标识语法来表示文档的结构。XSLT 处理器使用树状结构作为其输入，并形成另一个树状结构作为输出（见图 5.9）。通过对 XML 文档进行语法分析来生成输入树状结构，而输出树状结构通常被串行化到另一个 XML 文档中。但 XSLT 处理器本身操作的是树状结构，而不是 XML 字符流。这个概念最初给许多用户的感觉是不切实际的，结果却对理解如何执行更复杂的转换起了关键作用。

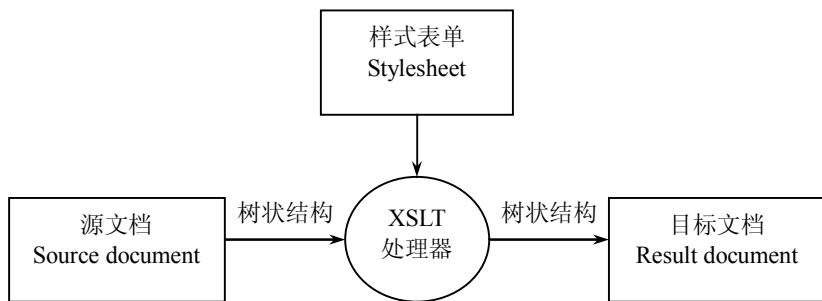


图 5.9 XSLT 处理器的操作

XSLT 使用叫作 XPath 的子语言来引用输入树中的节点。XPath 本质上是与具有层次结构的 XML 数据模型相匹配的查询语言。它可以通过按任何方向浏览树来选择节点，并根据节点的值和位置应用谓词。它还包括用于基本字符串处理、数字计算和布尔代数的工具。XPath 在单独的 W3C 推荐书中定义，它允许使用在其他上下文中再使用的查询语言，特别是用于定义扩展超链接的 XPointer。

XSLT 以传统语言（如 Lisp、Haskell 和 Scheme）中的功能性编程的概念为基础，样式表由模板组成。XSLT 语言允许定义变量，但不允许现有变量更改它的值，即没有赋值语句。XSLT 是一种用于操作 XML 文档的完整高级语言，就如同 SQL 是操作关系表的高级语言一样。应该注意到 XSLT 不仅是一种样式设计语言，它比 CSS，甚至 CSS2 的功能更强大。

例 5.12 定义并显示 XML 文档内容的 XSLT 文档 (L5-12.xsl)，如图 5.10 所示。

```

01:<?xml version="1.0" encoding="GB2312" ?>                                <!-- 声明语句 -->
02:<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"> <!--声明按 XSLT 规范转换文档-->
  
```

```

03:<xsl:template match="/">
04:<HTML><TITLE>XML 数据输出</TITLE>
05:  <BODY>
06:    <H3>二手车购价</H3>
07:    <TABLE border="2" bgcolor="#AAFF00">
08:      <TR>
09:        <TH>品牌</TH>
10:        <TH>系列</TH>
11:        <TH>型号</TH>
12:        <TH>产地</TH>
13:        <TH>总价-万元</TH>
14:        <TH>出厂日期</TH>
15:      </TR>
16:      <xsl:for-each select="二手车/轿车">          <!-- 循环输出 -->
17:        <TR>
18:          <TD><xsl:value-of select="品牌"/></TD>          <!-- 输出指定的值 -->
19:          <TD><xsl:value-of select="系列"/></TD>
20:          <TD><xsl:value-of select="型号"/></TD>
21:          <TD><xsl:value-of select="产地"/></TD>
22:          <TD><xsl:value-of select="总价-万元"/></TD>
23:          <TD><xsl:value-of select="出厂日期"/></TD>
24:        </TR>
25:      </xsl:for-each>
26:    </TABLE>
27:  </BODY>
28:</HTML>
29:</xsl:template>
30:</xsl:stylesheet>

```



图 5.10 XSLT 处理 XML 文档的结果

说明：XSLT 本身也是 XML 文档，第 1 行自然也就是标准的声明语句。第 2 行是标准的 XSLT 文档的首行代码，xsl:stylesheet 表示将文档作为样式表处理；xmlns:xsl 属性用来防止元素名称重复和混乱，并遵守 W3C 的 XSLT 规范。第 3 行表示源文件在当前目录下。第 16、18 行等语句是遵守 XSLT 语法的语句，请参阅其他书籍。第 29、30 行是结束标识。

5.3.3 XML 应用

通过以上介绍,我们了解到 XML 只能描述数据,所以制作 XML 网页的关键也就是解决其显示问题。显示 XML 数据的方法有多种,例 5.11、例 5.12 是诸多方法中的一种,在下面的例题中还将介绍。

例 5.13 从数据岛(Data Islands)读取数据(数据岛是被 HTML 页面引用或包含的 XML 数据)。XML 数据既可以包含在 HTML 文件内,也可以放在某外部文件内(L5-13.htm)。

```

01:<HTML>
02:<HEAD>
03:  <TITLE>XML 数据在文件内</TITLE>
04:</HEAD>
05:<BODY>
06:  <XML ID="xml 二手车">
07:    <?xml version="1.0" ?>
08:    <二手车>
09:      <轿车>
10:        <品牌>北京现代</品牌>
11:        <系列>领翔</系列>
12:        <型号>2.0GL MT</型号>
13:        <产地>北京</产地>
14:        <总价-万元>12.5</总价-万元>
15:        <出厂日期>2008.10</出厂日期>
16:      </轿车>
17:      <轿车>
18:        <品牌>一汽大众</品牌>
19:        <系列>新宝来</系列>
20:        <型号>1.6L 自动豪华</型号>
21:        <产地>长春</产地>
22:        <总价-万元>10.7</总价-万元>
23:        <出厂日期>2008.9</出厂日期>
24:      </轿车>
25:    </二手车>
26:  </XML>
27:  <TABLE datasrc="#xml 二手车" border="1">
28:    <THEAD>
29:      <TH>品牌</TH>
30:      <TH>系列</TH>
31:      <TH>型号</TH>
32:      <TH>总价-万元</TH>
33:      <TH>产地</TH>
34:      <TH>出厂日期</TH>
35:    </THEAD>
36:    <TR>
37:      <TD><DIV datafld="品牌"></DIV></TD>
38:      <TD><DIV datafld="系列"></DIV></TD>

```

```

39:      <TD><DIV datafld="型号"></DIV></TD>
40:      <TD><DIV align="center" datafld="总价-万元"></DIV></TD>
41:      <TD><DIV datafld="产地"></DIV></TD>
42:      <TD><DIV datafld="出厂日期"></DIV></TD>
43:  </TR>
44: </TABLE>
45:</BODY>
46:</HTML>

```

说明：本例的数据与显示处于同一个文档中，没有使用属性 `encoding="GB2312"`。第 8 行至第 25 行是数据岛（XML 数据）。本例利用 `<XML>` 元素的 ID 属性来完成数据的读入（请注意“xml 二手车”是 XML 数据的根元素，可以仅用汉字或英文），通过能接收数据的 HTML 标识（如 `<TABLE>` 标识，把数据绑定到该 HTML），并格式化显示数据岛内的数据（显示输出的数据顺序不一定与数据岛内的数据顺序一致，如本例的“产地”与“总价”）。本例显示的是元素 `<轿车>` 内各 XML 子元素的内容。第 27 行的 `<TABLE>` 标识利用 `datasrc` 属性引用行内 XML，`datasrc` 属性实际上是通过在要处理的 XML 元素的 ID 属性的前面加上 # 来实现的。TD 元素不能绑定数据，这里使用 DIV 元素的 `datafld` 属性绑定要显示的 XML 元素，也可以使用 SPAN 元素，它们的介绍可参阅例 5.6。运行结果见图 5.11。

品牌	系列	型号	总价-万元	产地	出厂日期
北京现代	领翔	2.0GL MT	12.5	北京	2008.10
一汽大众	新宝来	1.6L 自动豪华	10.7	长春	2008.9

图 5.11 数据与显示代码同处一个文档

例 5.14 从文件的外部读取数据岛数据（L5-14.htm）。

```

01:<HTML>
02:<HEAD>
03:  <TITLE>XML 数据在文件外部</TITLE>
04:</HEAD>
05:<BODY>
06:<xml src="L5-15.xml" id="xmlhouse" async="false"></xml>
07:<TABLE datasrc="#xmlhouse" width="100%" border="1">
08:  <THEAD>
09:    <TH>品牌</TH>
10:    <TH>系列</TH>
11:    <TH>型号</TH>
12:    <TH>产地</TH>
13:    <TH>总价-万元</TH>
14:    <TH>出厂日期</TH>
15:  </THEAD>
16:  <TR>

```

```

17:    <TD><SPAN datafld="品牌"></SPAN></TD>
18:    <TD><SPAN datafld="系列"></SPAN></TD>
19:    <TD><SPAN datafld="型号"></SPAN></TD>
20:    <TD><SPAN datafld="产地"></SPAN></TD>
21:    <TD align="center"><SPAN datafld="总价-万元"></SPAN></TD>
22:    <TD><SPAN datafld="出厂日期"></SPAN></TD>
23: </TR>
24:</TABLE>
25:</BODY>
26:</HTML>

```

说明：第 6 行是将 XML 文件 L5-15.xml 调至 xmlhouse 数据岛内，async="false"属性的作用是确保任何超文本标识语言在解析之前调入 XML 数据。运行结果见图 5.12。



品牌	系列	型号	产地	总价-万元	出厂日期
北京现代	领翔	2.0GL MT	北京	14	2008.10
一汽大众	宝来	1.6L 自动豪华	长春	10.7	2008.9

图 5.12 数据与显示代码不在同一个文档

例 5.15 供例 5.14 读取的 XML 数据 (L5-15.xml)。

```

01:<?xml version="1.0" encoding="gb2312" ?>
02: <二手车>
03:   <轿车>
04:     <品牌>北京现代</品牌>
05:     <系列>领翔</系列>
06:     <型号>2.0GL MT</型号>
07:     <产地>北京</产地>
08:     <总价-万元>14</总价-万元>
09:     <出厂日期>2008.10</出厂日期>
10:   </轿车>
11:   <轿车>
12:     <品牌>一汽大众</品牌>
13:     <系列>宝来</系列>
14:     <型号>1.6L 自动豪华</型号>
15:     <产地>长春</产地>
16:     <总价-万元>10.7</总价-万元>
17:     <出厂日期>2008.9</出厂日期>
18:   </轿车>
19: </二手车>

```

例 5.16 应用 CSS 显示 XML 文档。XML 文档中含有需显示的数据，CSS 文件是显示格式的设置文件。运行结果见图 5.13。

XML 文件 (L5-16.xml):

```
01:<?xml version="1.0" encoding="GB2312" ?>
02:<?xml:stylesheet type="text/css" href="L5-16.css" ?>
03:<secondhand>
04:  <cprod>
05:    <cname>北京现代</cname>
06:    <cseri>领翔</cseri>
07:    <ctype>2.0GL MT</ctype>
08:    <carea>北京</carea>
09:    <price>12.5 万元</price>
09:    <cdate>2008.10</cdate>
10:  </cprod>
11:  <cprod>
12:    <cname>一汽大众</cname>
13:    <cseri>新宝来</cseri>
14:    <ctype>1.6L 自动豪华</ctype>
15:    <carea>长春</carea>
09:    <price>10.7 万元</price>
16:    <cdate>2008.9</cdate>
17:  </cprod>
18:</secondhand>
```

CSS 文件 (L5-16.css):

```
cprod{display: block;}
cname{display: block; font-size:20pt; font-weight:bold}
cseri{display:block; text-indent:2em}
ctype{display:block; text-indent:2em}
carea{display:block; text-indent:2em}
price{display:block; text-indent:2em}
cdate{display:block; text-indent:2em}
```

说明：XML 文件的第 2 行指定使用 CSS 样式文件显示 XML 数据。与 XSL 相比，CSS 略显不足，CSS 比较“笨拙”，不适合用于元素顺序、位置经常要变动的文件等情况，而 XSL 常会使用户随心所欲。



图 5.13 CSS 显示 XML 文档

5.4 Visual Studio 2005 简介

Visual Studio 2005 (以下简称 VS 2005) 是一套完整的开发工具集，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++、Visual C# (读作 C-sharp) 和 Visual J# 全都使用相同的集成开发环境 (Integrated Development Environment, IDE)，利用此 IDE 可以共享工具且有助于创建混合语言解决方案。另外，这些

语言利用了 .NET Framework 的功能, 通过此框架可使用简化 ASP Web 应用程序和 XML Web Services 开发的关键技术。

VS 2005 可用于快速构建和部署功能强大而安全的 .NET 应用程序, 解决最具挑战性的软件开发需要, 而构建的 VS 2005 进一步完善并增强了 Visual Studio .NET 2003 的功能, 新的 IDE 环境更高效、更人性化, 并与旧版本产品高度兼容。

使用 VS 2005 开发人员可以:

- 构建功能强大而且响应能力极好的基于 Windows 的应用程序。
- 构建功能强大而且响应能力极好的 Pocket PC 应用程序。
- 构建完善而安全的 Web 应用程序。
- 构建对设备有智能感知能力的完善而安全的移动 Web 应用程序。
- 在以上任何一种应用程序中使用 XML Web Services。
- 避免“DLL 灾难”。
- 消除代价高昂的应用程序部署和维护问题。

Visual Studio .NET 是唯一的从头到尾都基于通过 XML Web Service 进行集成这一思想而构建的环境。它允许应用程序通过 Internet 共享数据, XML Web Services 使开发人员能够利用新的和现有的代码构建应用程序, 而不用考虑平台、编程语言或对象模型。Visual Studio .NET 的基础是建立在 .NET 框架 (见图 5.14) 之上, 它提供了在设计、开发、调试和部署 Web 应用程序、XML Web services 和传统的客户端应用程序时所需的各种工具, 大大提高了开发者的效率, 简化了服务器端的开发, 提供了高效地创建和使用网络服务的方法等; 用户可以使用其中的任何一种语言开发应用程序。

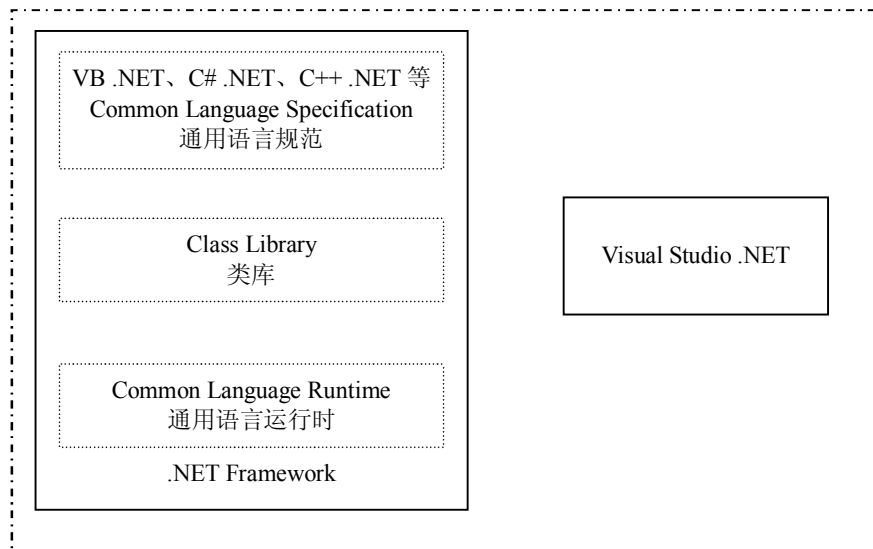


图 5.14 .NET 框架

5.4.1 初识 Visual Studio 2005

VS 2005 有两个主要版本: Visual Studio 2005 Professional 和 Visual Studio 2005 Team

System (准确地说, 针对不同的人员, 有三种不同的 Team System, 但它们的核心 Visual Studio 功能是一样的), 而其中有些实用特性只在 Team System 中才有。安装完 Visual Studio 2005 就是个人版, 再安装 Team 插件, 才能是 Team 版。服务器和 N 个客户端必须都装上 Team 才是一个完整 Team System。

VS 2005 的功能可以细分为 9 类: 项目与方案设计、文档和研究、安全与建模、编码、自动化、其他效率提升特性、构建和部署、调试和测试、VS 2005 的扩展。

VS 2005 的代码编辑器智能体现在以下几个方面:

(1) 代码分色显示。代码编辑器能够识别用户所编写语言的语法, 对其中的关键字高亮显示。

(2) 自动布局代码。在输入代码时, 编辑器将自动设置代码的缩进, 匹配代码块的首尾括号等。

(3) 自动语法检查。在输入代码时, 编辑器还将自动进行一些语法检查, 把可能产生错误的代码加下划线; 当鼠标放在有下划线文本上时, 会显示出错误信息。

(4) 代码折叠。在代码编辑器的左边有一系列的小减号图标 (见图 5.15 中的鼠标所指处), 每个减号标记出一个新代码块的起点。单击这些减号图标, 可以关闭对应的代码块视图, 同时减号图标变成加号图标 (见图 5.16, 这是源代码 custom.asp 的三大代码块)。单击加号图标, 则对应的代码块恢复显示。这一操作就像关闭或打开树型控件中的节点一样。

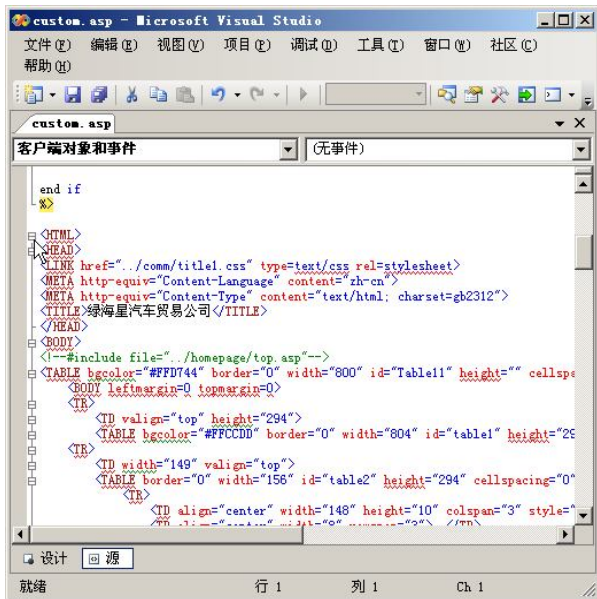


图 5.15 代码展开的状态

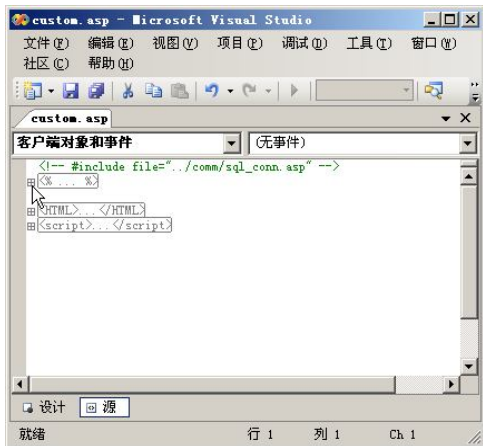


图 5.16 源代码被折叠

(5) 智能感知 (IntelliSense)。智能是指编辑器会自动列出对象的属性、方法以及方法的参数等, 该功能不仅减少了代码的输入工作量, 还能确保其正确无误。

VS 2005 的智能感知包括以下内容:

- 列出成员: 输入对象名称, 然后输入一个点, 代码编辑器就会显示一个列表框 (如图 5.17 中的 request 对象)。用户可在其中选择需插入到代码中的成员。

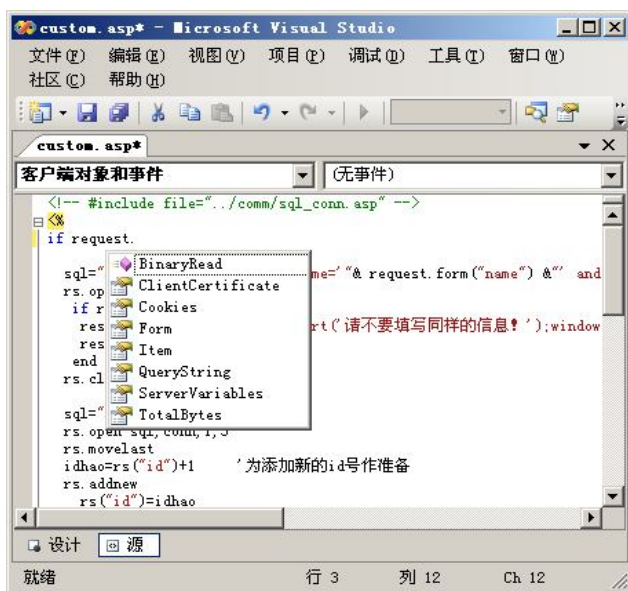


图 5.17 显示对象的成员列表框

- 参数信息: 在输入方法名称后, 输入左括号, 编辑器就会显示其参数列表, 如图 5.18 中的 response.write。

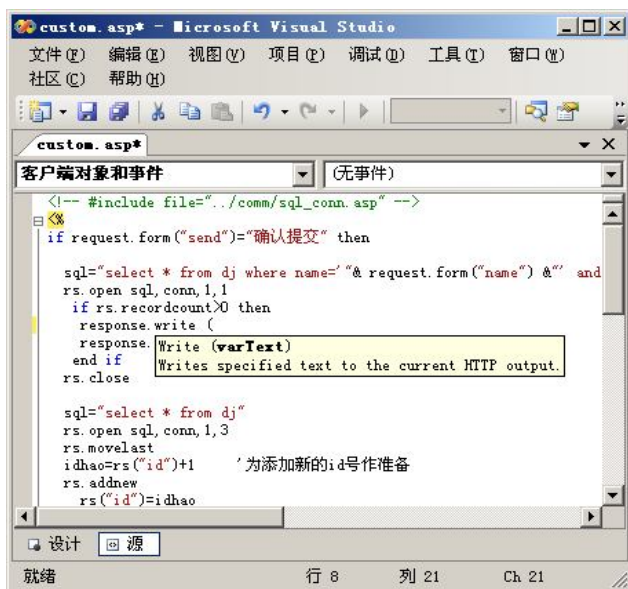


图 5.18 显示方法的参数列表

- 完成单词: 输入类名或成员名的一部分后, 按快捷键 Alt+→, 编辑器将显示一个列表框, 其中列出用户可能要输入的名称, 如图 5.19 所示, 从中选择一个即可完成输入。

还可以通过菜单或快捷键访问智能感知功能。智能感知命令位于“编辑”菜单的最下方, 如图 5.20 所示。

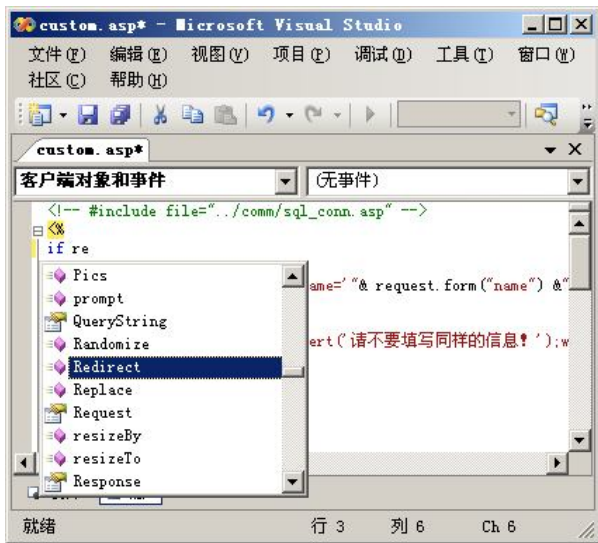


图 5.19 完成单词

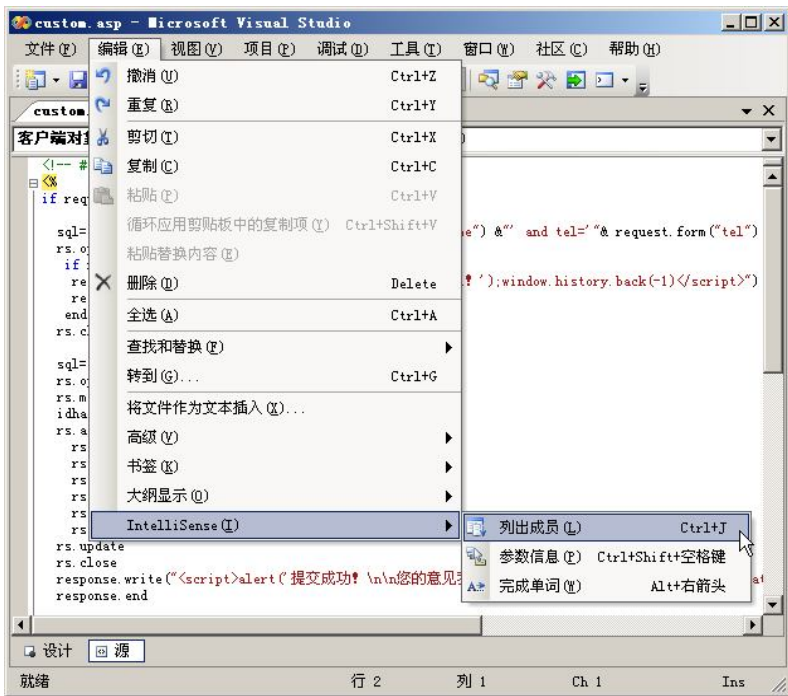


图 5.20 通过菜单访问智能感知

5.4.2 使用 Visual Studio 2005

编写.NET 程序最简单的环境其实有一个记事本和一个编译器就足够了，之所以需要工具集或者 IDE 是因为现在企业需要快速、高效地开发出稳定、实用的计算机应用程序，而工具集的作用也在于帮助程序员快速、高效地开发。

单击“开始/所有程序/Microsoft Visual Studio 2005”，启动 Visual Studio 2005，进入“起

始页”，如图 5.21 所示，选择“文件/新建/文件”命令（如图 5.22）所示，弹出“新建文件”对话框（如图 5.23 所示）。在“新建文件”对话框的类别中可以选择“常规”或“脚本”，在“常规”类别下，可以建立文本文件、HTML 页、样式表、XML 文件、XML 架构、XSLT 文件、位图文件、图标文件、光标文件、本机资源模板等。也可以选择“脚本”类别，建立 JScript 文件、VBScript 文件和 Windows 脚本宿主等（如图 5.24 所示）。其他详细操作请参阅 VS 2005 的相关书籍。

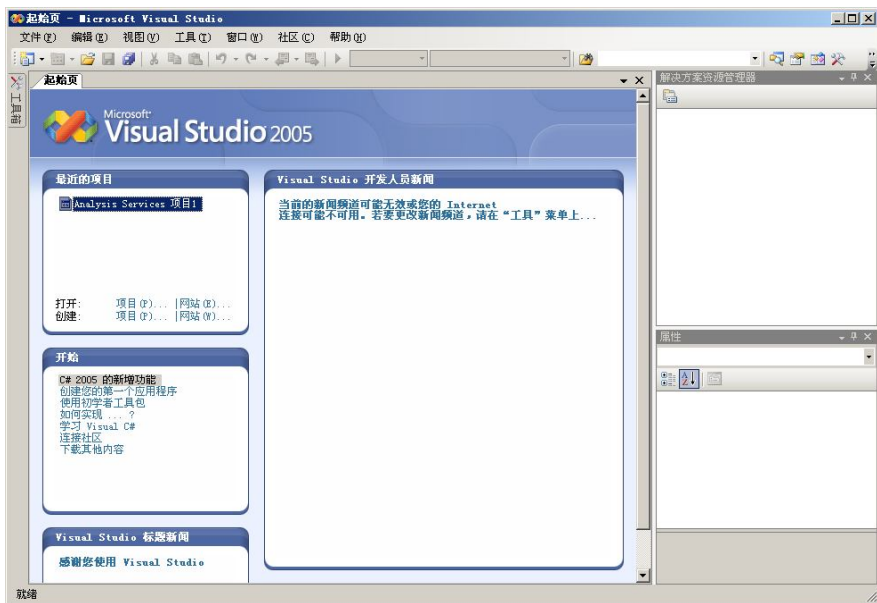


图 5.21 VS 2005 的起始页

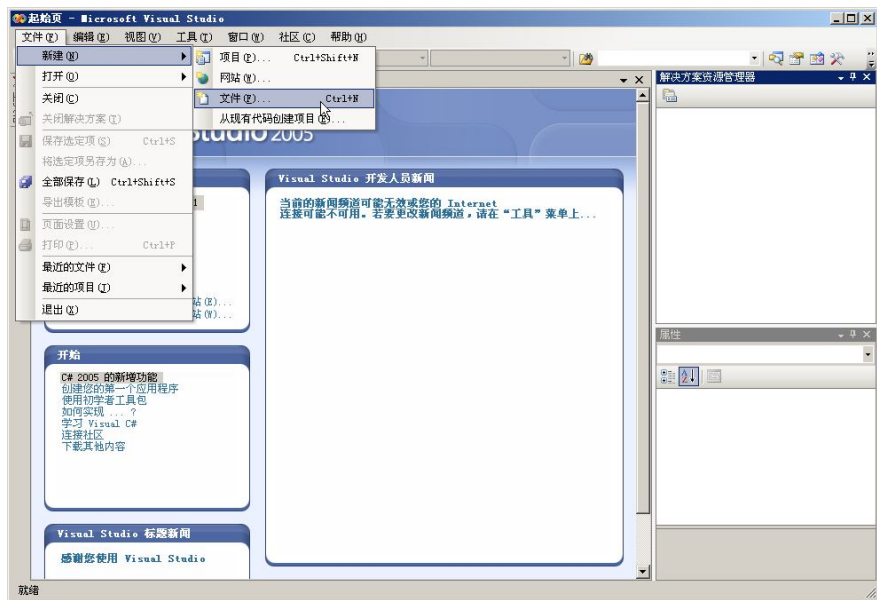


图 5.22 选择“文件/新建/文件”



图 5.23 “新建文件”对话框



图 5.24 建立脚本文件

5.5 使用 FrontPage 2003

FrontPage 2003 是微软 Microsoft Office 2003 的组件之一，整个窗口界面、菜单类型均与 Word、Excel 等类似，以便 Office 用户可以非常轻松地学习 FrontPage 2003；同时，FrontPage 2003 与 Office 的无缝集成使用户能够有效地利用已有的许多数据。

FrontPage 2003 提供了一个万维网开发环境，它简单易学、易用，且功能强大。FrontPage 2003 不但可以用来创建和制作网页，同时提供了网站设计、开发、制作与管理等多项功能。可以使用 FrontPage 2003 “所见即所得”的强大功能，在短时间内组合文字、表格、图片、声音和其他对象，构制成一幅美丽的网页。FrontPage 2003 也提供了方便实用的 HTML 源代码编辑器，通过它用户可以直接写或修改源代码。在 FrontPage 2003 提供的“拆分”视图下，初学者可以一边设计网页，一边学习 HTML 源代码。使用 FrontPage 2003 可以在一种轻松的环境下学习和强化对 HTML 的认知和应用。

5.5.1 FrontPage 2003 的主界面

启动 FrontPage 2003（选择开始/程序/Microsoft Office/Microsoft Office FrontPage 2003），打开 FrontPage 2003，如图 5.25 所示，进入网页设计视图。这是一张为用户准备好的空白网页，类似于 Word、Excel 的初始界面。FrontPage 2003 主窗口左下方的视图切换栏内提供 4 种

网页视图：设计、拆分、代码和预览视图（若选择制作框架网页，还会出现无框架视图，参见图 5.55）。

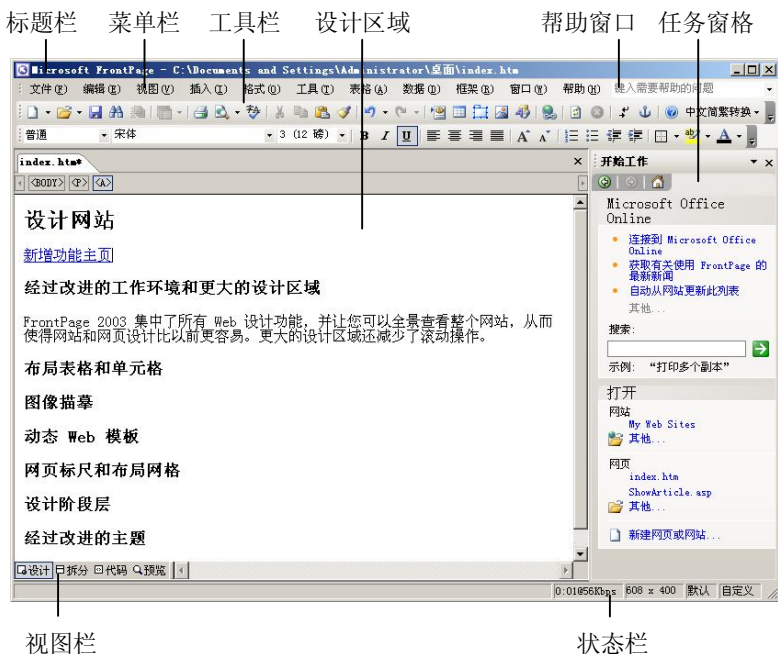


图 5.25 FrontPage 2003 的初始主界面

5.5.2 FrontPage 2003 的视图

FrontPage 2003 提供了比以往版本更丰富的视图模式，以适应用户的各种需要。这里共有 7 种视图模式，分别是网页、文件夹、远程网站、报表、导航、超链接和任务视图。

1. “网页”视图

打开 FrontPage 2003，立即进入默认的显示界面“网页”视图模式。“网页”视图模式是设计网页的基本视图模式，在这种模式下用户可以所见即所得（WYSIWYG）地非常容易地制作自己心仪的网页。“网页”视图模式又分 4 种：“设计”视图、“拆分”视图、“代码”视图和“预览”视图（见图 5.26 的视图栏），其界面分别如图 5.26、图 5.27、图 5.28 和图 5.29 所示。当打开或者新建一个框架网页时（选择“开始/程序/FrontPage 2003/文件/新建/新建网页/其他网页模板/框架网页”），FrontPage 2003 主界面的视图栏中会增加一个“无框架”视图（参见图 5.55）。

（1）“设计”视图。“设计”视图显示 Web 页面，用户在此使用“所见即所得”的方式设计、编辑、修改网页，查看一般网页的编辑效果和一般浏览器上不出现的页面要素（如注释、无边框表格线、共享边框线等）。

（2）“拆分”视图。“拆分”视图是 FrontPage 2003 的新增功能。使用“拆分”视图，用户可以同时看到设计区和代码区。无论在哪个区进行设计、编辑、修改等操作，另一区的内容都会随之改变。使用这种视图，初学者可以很快地掌握 HTML 的基本语法和超文本标识语言的各类标识。

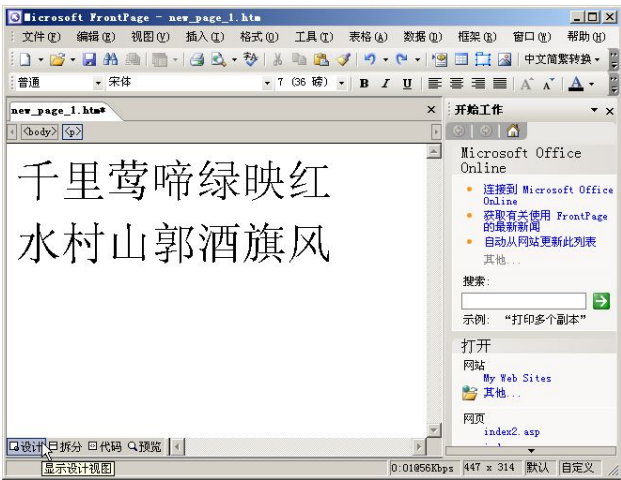


图 5.26 “设计”视图

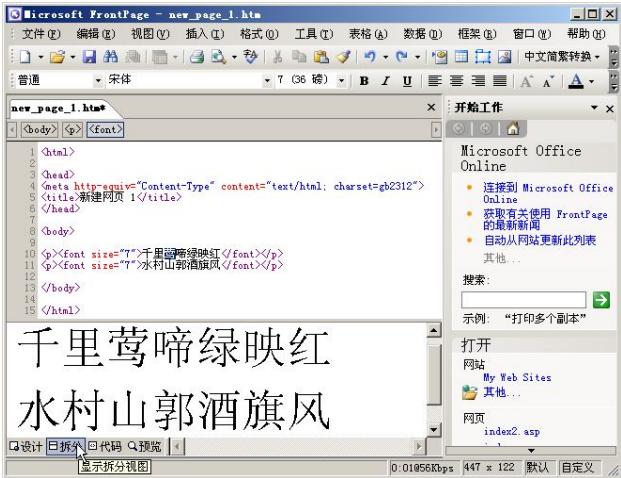


图 5.27 “拆分”视图

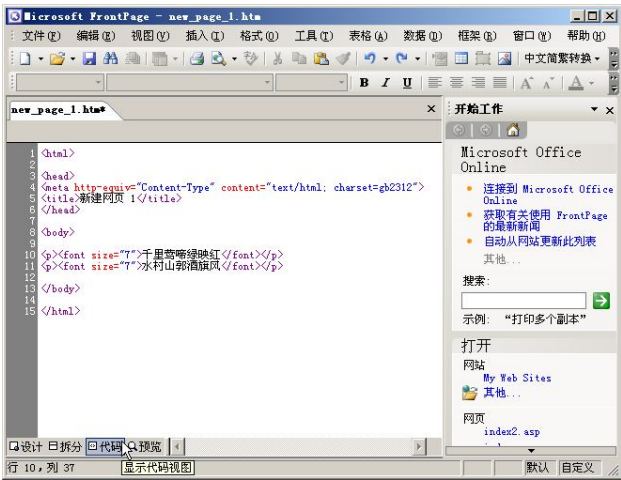


图 5.28 “代码”视图

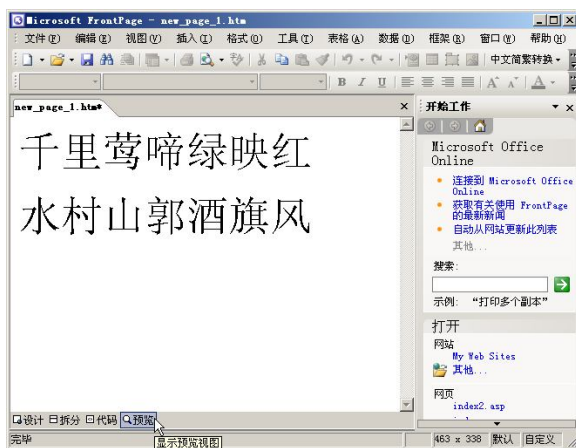


图 5.29 “预览”视图

(3) “代码”视图。在“代码”视图中显示的是页面文件的源代码,用户可以看到 FrontPage 2003 自动生成的页面 HTML 源代码。这对高级用户来说无疑是很重要的,因为网页的内容千变万化,并不是“所见即所得”的编辑方式都能满足要求,有时在“代码”视图下修改网页参数要简单得多。

(4) “预览”视图。“预览”视图提供一个接近浏览器的显示方式(之所以说“接近”,是因为显示某些网页对象时与浏览器有出入,仅供参考;读者做好的页面必须用浏览器查看效果)。用户不但可以预览编辑好的页面,还可以预览页面中的动态图案(DHTML)和 ActiveX 控件。虽说“预览”视图提供了方便的预览环境,但读者应当切记,浏览器的判断才是网页显示效果的最后裁决,“预览”视图仅提供粗略显示网页效果的环境。

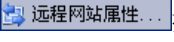
(5) “无框架”视图。“无框架”视图是为不支持框架显示的浏览器设置的。选择“无框架”视图,可以创建一个不带框架的页面,以满足不支持框架显示的浏览器在显示框架时的需求,而不因在显示此框架网页时产生错误。“无框架”视图请见图 5.55 的视图栏。

2. “文件夹”视图

仅仅制作单一页面时,“文件夹”视图和后面的其他视图在菜单中呈灰色,不可以选择,只有在设计制作网站时才有可能选择这些视图,而且此时视图栏中会显示这些视图。

“文件夹”视图用于有效地组织和管理网站。单击视图栏的“文件夹”按钮,FrontPage 2003 将以列表的形式给出当前 Web 站点的文件夹和网页文件的组织结构,对于站点内的文件包括名称、标题、大小、类型、修改日期、修改者和注释等,见图 5.30,或者单击菜单栏“视图/文件夹”命令。

3. “远程网站”视图

“远程网站”视图用于站点和站点内部文件的发布。单击视图栏中的“远程网站”,打开“远程网站”视图窗口(见图 5.31)。单击  按钮设置远程网站属性,然后,用户可在弹出的窗口中直接将当前站点中的文件和文件夹拖放至远程站点中发布。

4. “报表”视图

“报表”视图(见图 5.32)用于对当前站点的量化统计管理,包括网站摘要、文件、共享内容、问题、工作流、使用率等六大块。单击菜单栏的“视图/报表”命令,或者单击视图栏中的“报表”按钮,打开“报表”视图窗口。

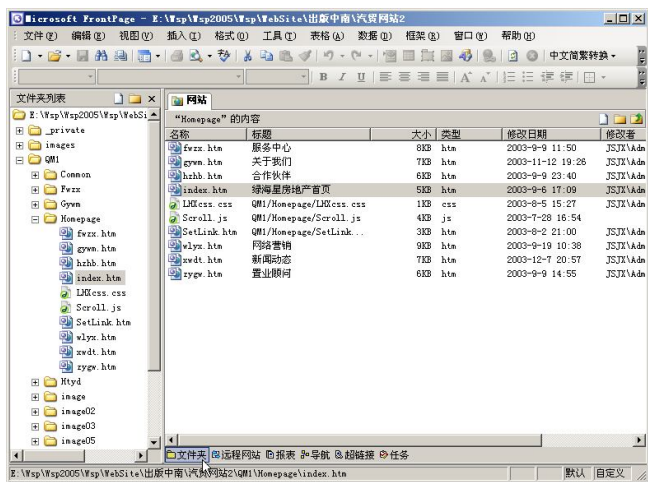


图 5.30 “文件夹”视图窗口

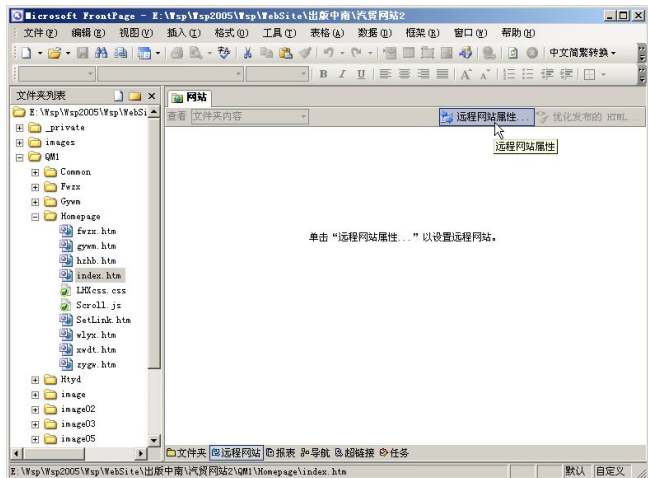


图 5.31 “远程网站”视图

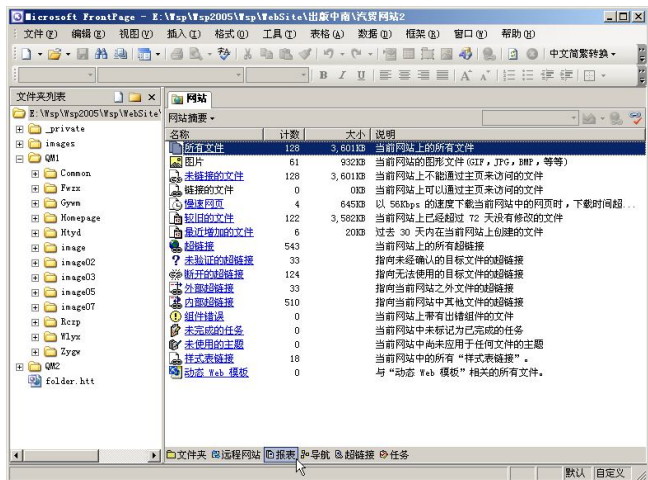


图 5.32 “报表”视图窗口

5. “导航”视图

“导航”视图用于查看站点中所有的网页以及这些页面之间的组织逻辑关系，见图5.33。单击视图栏中的“导航”按钮，打开“导航”视图窗口。导航图框中的“+”号，表示图框处于折叠状态；“-”号表示图框处于展开状态。可以单击这两种符号切换折叠与展开状态。

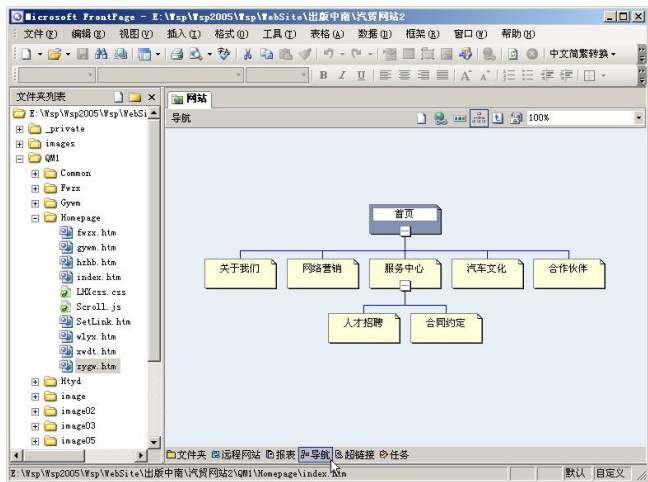


图 5.33 “导航”视图窗口

6. “超链接”视图

“超链接”视图用于显示网页和页面文件间的所有超链接关系，见图 5.34。单击其中的超链接，将自动打开预置对应的网页。

7. “任务”视图

“任务”视图用于组织或管理各用户小组以掌握工作进度，跟踪正在完成以及尚未完成的工作，见图 5.35。所有的工作都将出现在任务列表中，“任务”视图将以常规方法显示它们。

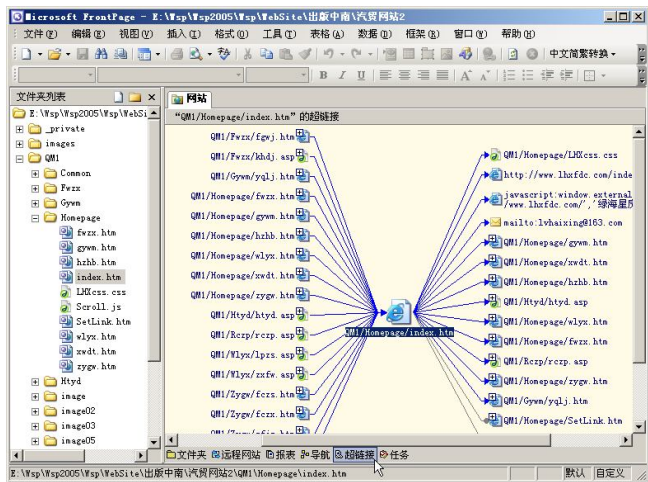


图 5.34 “超链接”视图窗口

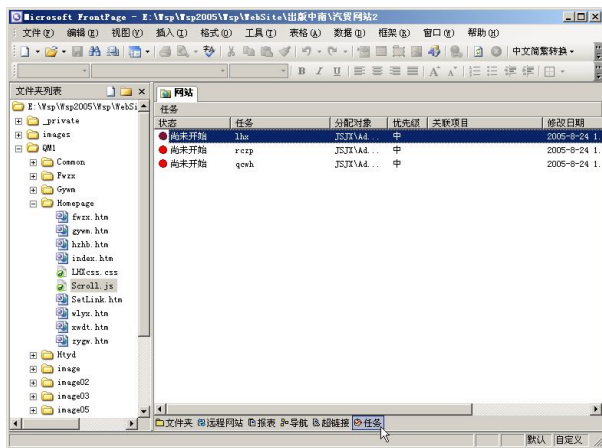


图 5.35 “任务”视图窗口

5.5.3 创建网页页面

1. 直接创建新网页

选择“开始/程序/Microsoft Office/Microsoft Office FrontPage 2003”启动 FrontPage 2003 (如图 5.25 所示), 在系统默认的“设计”视图下使用“所见即所得”的方式, 可以直接输入文字、插入图片、建立超链接等, 以简洁、方便、快速地创建网页; 还可以通过“代码”视图直接修改源代码或在“拆分”视图下一边设计一边查看源代码; 最后可以通过“预览”视图查看创建的网页效果。

2. 使用模板向导创建新网页

在菜单栏选择“文件/新建”命令, 打开“新建”任务窗格, 在“新建”任务窗格的“新建网页”栏中单击“其他网页模板”, 打开“网页模板”对话框 (如图 5.36 所示)。在“网页模板”对话框中有常规、框架网页和样式表三个选项卡。“常规”选项卡提供常用的网页结构组成模板; “框架网页”选项卡提供常用框架网页的结构模板; “样式表”选项卡提供常用的层叠样式表架构。在“常规”选项卡下可以选择合适的模板, FrontPage 2003 针对不同的用户设计了各种不同用途的模板和向导, 利用这些模板和向导用户可以轻松地创建自己的网页。使用模板向导的最大好处是节省了用户的宝贵时间, 避免了许多冗余操作。

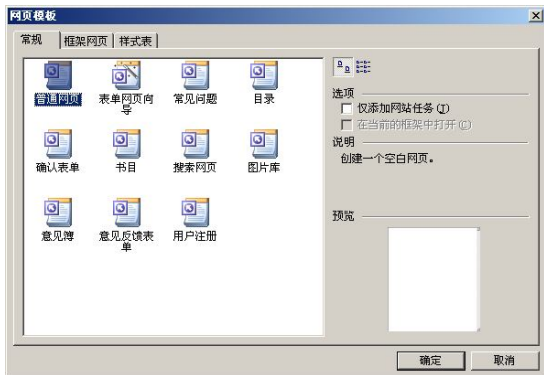


图 5.36 “网页模板”对话框

3. 设置网页的内容

在定制的网页内容中,有几项较常用的设置:网页的标题、背景音乐、背景颜色或图片、网页的版式。

在 FrontPage 2003 的菜单栏中,单击“文件/属性”命令或在网页设计区右击,选择“网页属性”命令,均会弹出“网页属性”对话框,它含有五个选项卡:常规、格式、高级、自定义、语言和工作组,如图 5.37 所示。

(1) 常规。在“常规”选项卡中(见图 5.37),可以设置:

- 网页的基本位置、标题、网页说明、关键字和默认的目标框架等。
- 背景音乐的位置、循环次数等。

(2) 格式。在图 5.38 的“格式”选项卡中,可以选择:

- 背景图片和它的来源或添加水印。
- 背景和前景(文本)的颜色。
- 超链接(未链接、链接时和链接后)的颜色。

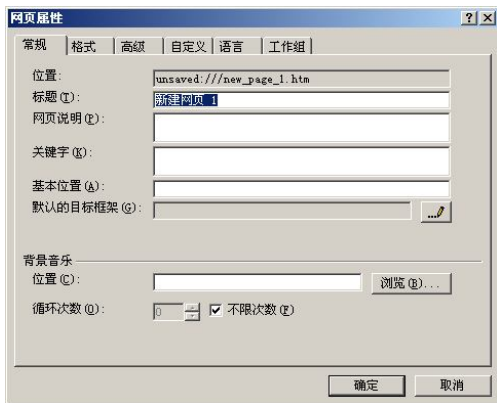


图 5.37 “常规”选项卡



图 5.38 “格式”选项卡

(3) 高级。在图 5.39 的“高级”选项卡中,可以选择:

- 以像素为单位的上、下、左、右边距和边距的宽度、高度。
- 正文样式、翻转样式等。
- 设计阶段控件脚本(客户端或服务器端平台、服务器端脚本、客户端脚本)。

(4) 自定义。“自定义”选项卡一般较少使用,是为满足用户添加一些系统变量和用户变量而设,如图 5.40 所示。在“自定义”选项卡中,可以选择:

- 系统变量(HTTP-EQUIV)。
- 用户变量。

(5) 语言。进入“语言”选项卡,用户可以为不同的语言平台编写各自的网页,如图 5.41 所示。在“语言”选项卡中,可以选择:

- 网页语言。
- HTML 编码。

(6) 工作组。在图 5.42 的“工作组”选项卡中,可以选择:

- 可用类别。

- 分配对象的名称、审阅状态等。
- 发布网站时是否包含此文件。



图 5.39 “高级”选项卡



图 5.40 “自定义”选项卡

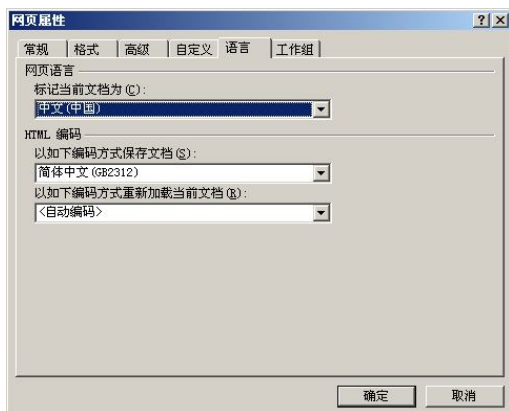


图 5.41 “语言”选项卡



图 5.42 “工作组”选项卡

4. FrontPage 2003 中网站的主题

(1) 设定网站主题。FrontPage 2003 提供了一个网站主题的设置，给网站的所有网页（或单一页面）建立一个使用标准。这样做是为了使网站中的各网页保持统一的外观，而且在变更外观时，只需改变这个设定就可以改变所有网页的设定，使设计变得十分简单和便利。

单击菜单栏的“格式/主题”命令，打开“主题”任务窗格，进入主题设定视窗，见图 5.43。选中任一主题，单击展开三角形，可以选择“应用为默认主题”、“应用于所选网页”或者“自定义”。选择“应用为默认主题”设置网站主题。

(2) 设置特定网页的主题。在一个网站的设计过程中，很可能有一些较特殊的页面。要建立有别于网站主题的次主题或者设置个别网页的主题，只需要先选中要更改的网页，单击“格式/主题”命令，在“主题”视窗（见图 5.43）的“主题”任务窗格中，选择“应用于所选网页”即可。

(3) 自定义。选择“主题”任务窗格中的“自定义”，可以对同一主题进行颜色、图形、文本以及背景图片等调整，以此获得不同的风格。请参阅图 5.43。

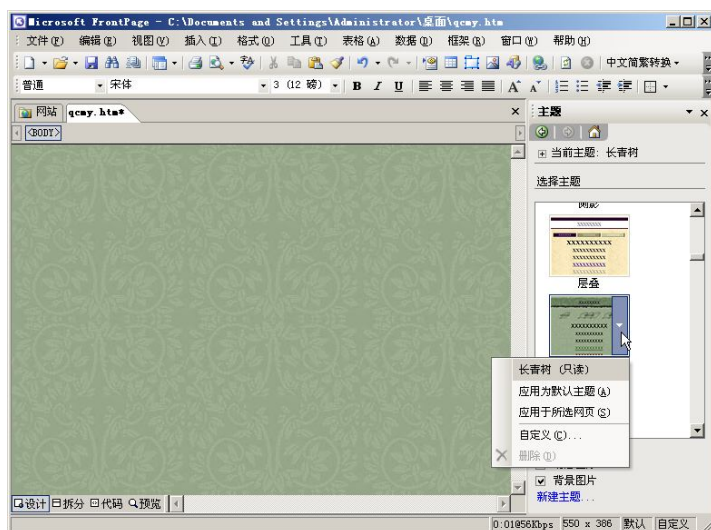


图 5.43 设定主题

5.5.4 文字与超链接

1. 在页面中添加文本

通常网页中不能缺少的内容就是文本，它是网页中最基本的组成部分。

(1) 输入文本。进入 FrontPage 2003 (见图 5.25)，打开“设计”视图，可以直接在设计区输入用户需要在网页中显示的文本素材。FrontPage 将自动为用户生成相应的 HTML 源代码。

用户还可以在“拆分”或“代码”视图下，查阅或修改对应页面的 HTML 源代码。使用这两种视图对初学 HTML 的用户无疑是一个良好的环境，同时对 HTML 编程老手来说，也是一个修改源代码的良好界面。

(2) 文本的基本操作。

1) 改变文字的大小、字体、颜色。和 Office 2003 的其他组件以及旧版本的 Office 一样，在 FrontPage 2003 中编辑、修改或改变文字的大小、字体、颜色等，只需在工具栏中选取相应工具，操作几乎与 Word 2003 完全一样。

2) 分段。在“设计”视图下编辑文本时，FrontPage 2003 将键入的回车，认为既是换行，又是分段。它会把下一行文字作为新的段落起点，并与上一行文字隔有一定的距离。若用户不需要分段，只是换行，可以使用组合键 Shift+回车。

2. 创建超链接

互联网最精彩、最迷人之处莫过于超链接 (Hyperlink)。只要轻轻点击鼠标，全球的信息海洋任君遨游，这就是超链接的结果。超链接是将事先准备好的文本、图片、按钮等对象与其他对象建立一种链接，即在源端点和目标端点之间建立一种链接。源端点是超链接的起始端点，又称源锚；目标端点是链接的对象，又称目标锚。


超链接按源端点的链接对象划分，可以分为超文本链接和非超文本链接。超文本链接的源锚文本下方一般都有下划线；非超文本链接是用除文本以外的其他对象构成的链接，这里的源锚既可以是图片、表格、表单，也可以是多媒体对象等。

超链接按目标端点的链接对象划分,可以分为外部链接、内部链接、电子邮件链接等。

在网页中建立超链接以后,剩下的只是点击鼠标了。简单地在所设定的超链接(文本、图片等)上单击,就能完成超链接的任务,到达被链接的“彼岸”。

(1) 建立文本或图片的超链接。在 FrontPage 2003 下建立文本或图片超链接的步骤是:

1) 选中要建立超链接的文本或图片。

2) 在菜单栏中单击“插入/超链接”命令,弹出图 5.44 的“插入超链接”对话框,或单击按钮,或右击选择“超链接”命令。

3) 在弹出的“插入超链接”对话框中,可以选择链接到:

- 原有文件或网页。
- 本文档中的位置。
- 新建文档。
- 电子邮件地址。

4) 选择“原有文件或网页”,在“地址”栏内直接输入被链接的地址(如绿海星网站地址 <http://www.csasp.net>),若不知道要建立的“地址”,可以依据“插入超链接”对话框中的一些按钮来获得帮助,参见图 5.44。

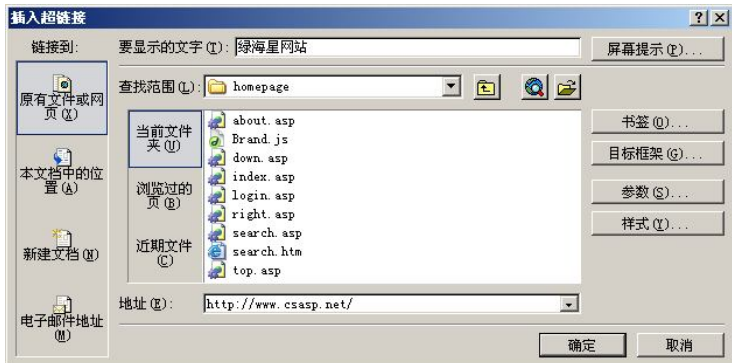


图 5.44 “插入超链接”对话框

(2) 建立网站内部的超链接。网站内部的超链接有两种情况,一种是网站内部文件之间的关联,另一种是网站内部文件与外部的超链接。

在“插入超链接”对话框(见图 5.44)中,选择“原有文件或网页”,FrontPage 2003 给出“当前文件夹”、“浏览过的页”和“近期文件”三个按钮,帮助用户较快地确定链接对象。若链接对象不在以上选择之列,可以通过“查找范围”和“地址”分别确定链接网站内部文件或网站外部的超链接。

(3) 建立网页内部的超链接。前面谈到过<A>标识,它有一个 name 属性,用于建立文档内部的超链接。文档的内部超链接在 FrontPage 2003 中称为“书签”,适用于文本较长的网页。通常利用这种功能在网页的前部建立一个带有超链接的目录,使浏览者进入网页时就可以对网页的内容有一个概貌性的了解,并可以快速地选定要阅读的内容。

1) 设定书签。

- 选取要设定书签的文字。
- 在菜单栏单击“插入/书签”命令,弹出“书签”对话框,见图 5.45。

- “书签名称”文本框内显示已选取的设定书签文字(可以改选其它的名称),单击“确定”按钮即可,被设定为书签的文字下方会出现虚线。
- 删除“书签”,只需在图 5.45 中选中要删除的书签名称,单击“清除”按钮即可。

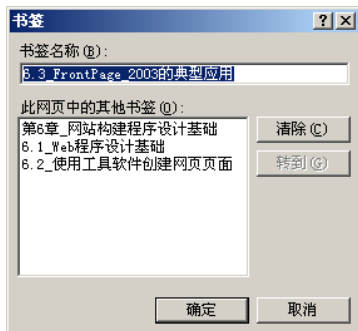


图 5.45 “书签”对话框

2) 建立内部超链接。

- 在所编辑的文本中,选定想要设置内部超链接的文字。
- 从菜单栏中选取“插入/超链接”命令,或右击,单击“超链接”命令。
- 在“插入超链接”对话框右边(见图 5.44),单击“书签”按钮,在“书签”列表框中选取需链接的书签名称。

例 5.17 建立网页内部的超链接(L5-17.htm)。


```
<HTML>
<HEAD>
  <TITLE>网页内部的超链接</TITLE>
  <STYLE>
    <!--
      P.title1 {font-family:"黑体"; font-size:"30"; color:"Maroon"}
      P.title2 {font-family:隶书; font-size:20; color:black}
      P.title3 {font-family:"宋体"; font-size:"20"; color:"Olive"}
      .note1 {color:Green}
      .note2 {color:Blue}
    -->
  </STYLE>
</HEAD>
</HEAD>
<BODY>
  <P class="title1">《<A name=念奴娇>念奴娇</A>·赤壁怀古》
  <FONT size=2><A href="#注释">至注释</A>→</FONT></P>
  <P class=title2>苏轼</P>
  <BLOCKQUOTE>
  <P class="title3">
    大江东去,<BR>浪淘尽,<BR>千古风流人物。<BR>
    故垒西边,<BR>人道是,<BR>三国周郎赤壁。<BR>
    乱石崩云,<BR>惊涛裂岸,<BR>卷起千堆雪;<BR>
    江山如画,<BR>一时多少豪杰。<BR><BR>
```

```
    遥想公瑾当年, <BR>小乔初嫁了, <BR>
    雄姿英发, <BR>羽扇纶巾, <BR>谈笑间, <BR>强虏灰飞烟灭。 <BR>
    故国神游, <BR>多情应笑我, <BR>早生华发。 <BR>人生如梦, <BR>一尊还酹江月。
</P>
</BLOCKQUOTE>
<A name=注释>注释</A>: <BR>
<DIV class="note1">念奴娇（词牌名）<BR>
    苏轼（1037-1101），字子瞻，号东坡居士，眉山（今四川眉山县）人。 <BR></DIV>
<SPAN class="note2">他是北宋时诗坛成就卓越的大家。</SPAN>
    宋仁宗景佑三年生，嘉佑二年进士，累官至端明殿学士兼翰林侍读学士，礼部尚书。 <BR>
    因讥讽朝政被贬任杭州通判；历徙湖州、黄州、常州。
    哲宗嗣位，召至京师，任中枢舍人。后又因哲宗亲政启用新党， <BR>
    苏轼又被一贬再贬，之贬到海南，宋徽宗登基大赦天下，苏轼北返时在常州逝世。 <BR>
    时为徽宗建中靖国元年，享年六十六岁。高宗朝，赐太师，谥文忠。 <BR>
    <FONT size=2><A href="#念奴娇">返回页首</A></FONT>
</BODY>
</HTML>
```

说明：本例已经设置好内部链接，如带有下划线的部分。练习此题时，读者可以先输入诗句，再做书签，即暂不输入带下划线的部分。然后按下列步骤操作：

- 1) 打开 FrontPage 2003 在“代码”视图输入例 5.17，注意暂不输入下划线部分。
- 2) 选择“插入/书签/确定”，将词牌名“念奴娇”和“注释”设为书签。
- 3) 分别选中“至注释”和“返回页首”，右击，在快捷菜单中选取“超链接”命令。
- 4) 在弹出的“插入超链接”对话框中，单击“书签”按钮，分别选取“注释”书签和“念奴娇”书签，单击“确定/确定”按钮。
- 5) 在“预览”视图中，单击“至注释”，光标将移动到词尾；单击“返回页首”，光标将移动到诗词的起始处。


(4) 利用浏览器寻找、建立外部的超链接。若用户暂时不能确定外部链接的确切地址，可按下列步骤操作：

- 1) 单击“插入超链接”对话框（见图 5.44）中“查找范围”右边的按钮。
- 2) 在新开启的预设浏览器中，输入（或找到）要链接的 URL 地址或文件，例如，欲链接绿海星网站的“教学资料”栏目。

3) 在新打开的浏览器中输入 www.csasp.net，找到“教学资料”栏目（如图 5.46 所示），按 Alt+Tab 键返回 FrontPage 2003，此时“地址”栏内已加上该链接的 URL 地址或文件。

(5) 建立电子邮件超链接。可以用两种方法建立电子邮件超链接：

- 直接在“插入超链接”对话框（见图 5.44）的“地址”栏内输入邮件地址。
- 单击“插入超链接”对话框中的“电子邮件地址”按钮，按提示在文本栏内输入邮件地址。

(6) 取消超链接。用 FrontPage 2003 编辑文本时，只要输入含有 <http://>或@（读作：at）等的字符串时，FrontPage 2003 都会自动为其建立地址或邮件的超链接。FrontPage 2003 的这种“高智商”有时并不被用户接受。此时要取消地址或邮件等的超链接，可以直接单击撤销按钮。对已有的超链接地址或邮件等，只需选中后右击，选择“超链接属性”命令，在弹

出的“插入超链接”对话框中（见图 5.44），将“地址”栏中的内容清除，或者单击“删除链接”按钮。



图 5.46 为创建超链接新开启的浏览器

5.5.5 对图像的处理

文字是网页不可缺少的组成部分，是最基本的结构单元，而网页上的图形图像主要用于装饰页面和烘托主题。虽说文本是网页结构的基础，但网站并未因文字的数量和华丽的词藻而誉满全球。能够体现网站灵魂，打动浏览者的网页，是文字与图形图像的完美结合。精美的图像、和谐的背景，能对页面起到极大的渲染效果，将给网站营造一种特殊的气氛。但是过大过多的图像会降低页面的下载速度，导致网站的其他信息遭到削弱；同时，也会给访问者留下不佳的印象，影响网站的形象。一般情况下应当尽量缩减图像的尺寸，限制颜色数目的使用，以低分辨率保存图像，在一个站点中尽可能地使用相同的图像。网页中常用的两种图像文件的格式是 GIF 和 JPEG，它们的共同特点是：压缩比高、失真小，GIF 图像格式可以存储动画，可以制作成透明图像嵌于网页中。

1. 网页的背景和背景图像

根据需要在网页中加上背景（色）或背景图像，可以增强网页的观赏性。采用背景时，页面中的文字内容应取与其对应的对比色，使得文字更加鲜亮，易于阅读；若采用背景图像，也应突显文字，可以通过图像处理软件 Photoshop 加工背景图像，或模糊、或增亮、或降低对比度等，或反之。背景图像只是网页的配角，不可喧宾夺主。FrontPage 2003 中添加背景和背景图像的方法是，在“设计”视图的设计区右击，选择“网页属性”命令，在打开的“网页属性”对话框中选择“格式”选项卡，见图 5.38，根据需要配置背景色或背景图像。或者单击菜单栏“文件/属性”命令，在弹出的“网页属性”对话框中选择配置。

2. 添加图

在 FrontPage 2003 中，可以非常简单地将一幅图像添加到网页中：

（1）将现有的图形图像文件直接插入到网页中，单击菜单栏“插入/图片”命令（类似 Word 等）；将一幅图形图像从另一处插入到本网页中。


(2) 添加的图像可能与网页的要求不太恰当, 用户也可能不太满意, 这时常常要对图像进行编辑。选中图像右击, 其图像编辑方法与 Word 几乎完全相同。


3. 在图片上书写文字

(1) 一种方法是直接打开“图片”工具栏(视图/工具栏/图片), 如图 5.47 所示; 另一种方法是, 选中图片右击, 选择“显示图片工具栏”, 打开“图片”工具栏。




图 5.47 “图片”工具栏

(2) 在“图片”工具栏上单击文本按钮, 如图 5.47 所示; 在图片中间会出现一个文本框(在此之前, 有时可能会弹出“确认”和“取消”对话框, 用户可根据需要选取), 可在其中输入文字, 或单击文本框四周的小黑点, 缩放文本框, 或用鼠标拖动该文本框, 置于所需之处。

(3) 在“图片”工具栏中, 除按钮以外, 还提供二十多种具有其他功能按钮, 诸如: 插入文件中的图片、自动缩略图、绝对定位、上(下)移一层、向左(右)旋转、水平(垂直)翻转、增加(减低)对比度、增加(减低)亮度、剪裁、线型、设置图片格式、设置透明色、颜色、凹凸效果、重新取样、选择、图像热点(可以设置长方形热点、圆形热点、多边形热点和突出显示热点)、还原等, 这些功能为用户处理页面中的图像提供了极大的帮助。

4. 创建自动缩略图

FrontPage 2003 提供自动缩略图, 可以帮助用户选定图片以最佳比例缩小显示在屏幕上, 它可以降低图片的分辨率和文件的大小, 以加快下载速率。FrontPage 2003 自动对自动缩略图创建指向实际图片的超链接。

选中图片, 单击“图片”工具栏(见图 5.47)的第三个按钮, 产生选定图片的自动缩略图。初学者最好将自动缩略图和所生成的超文本文件放在同一个文件夹内。

5. 图像的替代文本和替代图

当鼠标移动到图像上, 图像上出现的黄色文本标签称为图像的替代文本。替代文本主要是对所替代图像作文字说明或当某些原因致使图像不能正确显示时的文字描述。替代图类似于替代文本, 其作用是在正式的图形图像尚未完全送达用户的浏览器之前, 预先显示一幅分辨率较低的图片。

选中图片并右击, 在“图片属性”对话框中选择“常规”, 设置图形图像的替代图和替代文本。在“低分辨率”框内(见图 5.48)输入替代图片的路径、名称等, 或单击“浏览”按钮选择低分辨率替代图片所处的位置; 在“文本”框中输入替代文本, 并选中该复选框。

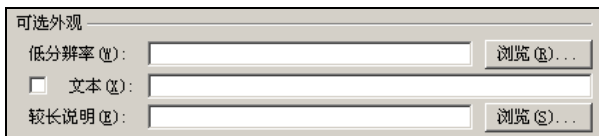


图 5.48 “常规”选项卡中的“可选外观”栏

6. 透明图像的建立

通常将图像的背景设为透明，其最大优点是可使图像与网页背景或其他图像很好地融合为一体。选择 FrontPage 2003 中“图片”工具栏（见图 5.49）上的“设置透明色”按钮，可以根据要求将图像的某种色彩设置为透明色。



图 5.49 “图片”工具栏中的“设置透明的颜色”按钮

透明色的设置过程：

- (1) 单击待处理的图片，在“图片”工具栏上单击“设置透明色”按钮。
- (2) 此时光标变为笔状，将鼠标光标移至待处理的图片上。
- (3) 在图片中单击任何一种颜色，这种颜色（只要图像上有）都会从图片中消失。

注意：①仅有 GIF 图像可以生成透明图像。②用同样的方法可以将其他任何一种色彩设置为透明色。③在一幅 GIF 图像中，只能针对一种色彩做透明色处理。

7. 图像地图

图像地图（Image Map）又称热点（Hot Spot），是图形图像上已设置的具有超链接属性的区域。创建图像地图的一般步骤是：

- (1) 选中待处理图像，右击，选择“显示图片工具栏”命令。
- (2) “图片”工具栏中提供矩形、圆形和多边形热点按钮，用户根据页面的需求选取。
- (3) 再次将鼠标移动到图像上，光标会变成铅笔状。
- (4) 在图像上按下鼠标左键并拖动，使之形成热点框；松开鼠标的同时 FrontPage 2003 会自动弹出“插入超链接”对话框，依据对话框的提示键入所需超链接。
- (5) 重复以上操作，可以在图像的其他位置继续创建热点。
- (6) 可以用鼠标将热点移至图像内的任一位置，也可选中热点周围的小黑点缩放热点。
- (7) 双击热点会弹出“插入超链接”对话框，可以修改图像上的热点超链接。
- (8) 若要删除图像热点，可以在选中该热点后按 Delete 键。

注意：①若选取的是多边形热点，在该多边形热点绘制封闭以后，才会弹出“插入超链接”对话框。②单击“文本”按钮，可以为图像生成含有文本注释的热点。③要想查看图像地图上的所有热点，可单击“突出显示热点”按钮。

5.5.6 表格与表单

表格是信息的组织形式，无论有形表格或无形表格（隐去表格线的表格）均呈网格状。在网页设计中利用它可以生成层次分明、内容复杂的页面。通常所见的网站，大多数采用无形表格来分割、排列、布局页面内容。

在 FrontPage 2003 中，选择菜单命令“表格/插入/表格”，弹出“插入表格”对话框（见图 5.50），根据需求建立表格。也可以选择“表格/绘制表格”菜单命令，手工绘制自选表格，或在弹出的工具栏内选择 FrontPage 2003 的新增功能“绘制布局表格”或“绘制布局单元格”

制表。还可以选择“表格/布局表格和单元格”菜单命令，在“布局表格和单元格”任务窗格中选择所需操作。

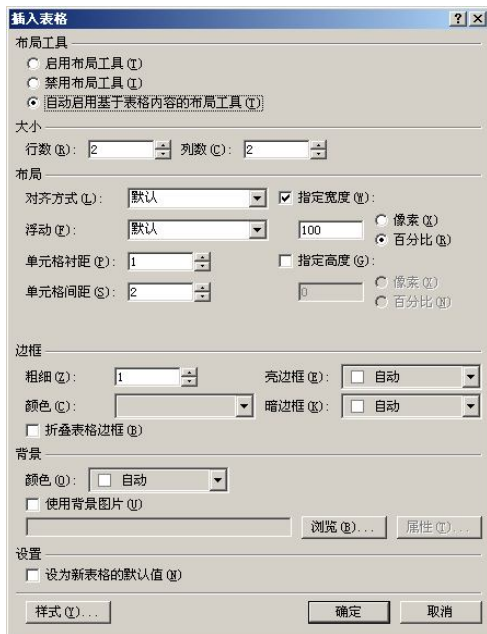


图 5.50 “插入表格”对话框

除了可以在图 5.50 的“插入表格”对话框中对表格进行设置以外，还可以在已绘制的表格单元内右击，在弹出的快捷菜单（见图 5.51）中选择“表格属性”或“单元格属性”命令，按提示和要求操作。FrontPage 2003 调整表格的方法也很简单，只需表格边框上移动鼠标，当鼠标的箭头变为双箭头时，就可以对当前表格进行扩展或收缩。FrontPage 2003 的新增功能“绘制布局表格”、“绘制布局单元格”可以非常简单地在已有表格中添加表格或表格单元。

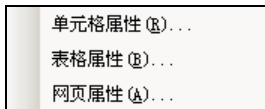


图 5.51 表格快捷菜单

与表格只有一字之差的表单是各类输入栏的集合，是客户端与服务器端之间进行数据交互的常用页面。图 5.52 是一个非常简单的客户端提交表单。当用户在客户端填写完表单上的所有数据，单击“确定”按钮，表单中的数据将通过提交程序送到服务器端，并由服务器端返回必要的应答数据，完成客户端与服务器端之间的交互操作。虽说表单与表格只相差一个字，但在概念和功能上完全不同，在使用时请读者注意其区别和应用环境。

1. 表单的构成

表单在形式上可能多种多样，但任何一款 HTML 表单一般都由三个基本部分组成：

- 一个或多个输入域。
- 一种提交结果的方式。
- 一个用于收集、处理并记录提交数据的程序。



图 5.52 提交表单

第三部分“提交程序”已超出本书的范围，读者可以参阅其他的参考书，如本书参考文献[21]，在此不多介绍。

2. 表单的创建

在 FrontPage 2003 中，可以使用多种方法轻松、快捷地创建表单。

(1) 用户自由选择表单元件（如文本框、文本区、文件上载、复选框、单选按钮、分组框、下拉框、按钮、高级按钮等）

1) 选择菜单命令“插入/表单”。

2) 在下拉菜单中用户可自由选取需插入的表单元件（如文本框、文本区、文件上载、复选框、单选按钮、分组框、下拉框、按钮、高级按钮等）。

3) 在第一次插入一个表单元件后，屏幕上将出现一个包含此表单元件的虚线框，以后的表单元件应该放在此虚线框内，因为它们同属于一个提交内容。如果在一个网页页面中需同时存在两个或两个以上的不同提交表单，屏幕上将出现两个或两个以上的虚线框，读者在使用时应分别对待。

(2) 使用模板。

1) 选择菜单命令“文件/新建”，打开“新建”任务窗格。

2) 单击“其他网页模板”，打开“网页模板”对话框（如图 5.36 所示）。

3) 单击“确认表单”。

3. 处理表单数据

表单中收集的数据最终是要提交给服务器端存储或处理的，FrontPage 2003 为此提供了十分方便的表单处理功能，用户不用编制提交程序，就能将表单中的数据提交给服务器。这个操作过程非常简单，只要将鼠标移动到表单上（即表单控件的虚线框内）并右击，在快捷菜单中单击“表单属性”命令，将弹出“表单属性”对话框，如图 5.53 所示。在图 5.53 中，“发送到数据库”有可能为灰色（不可选），其主要原因有三点：①本网页不属于站点下的网页，也就是说它是一个独立的页面；②本站点下无可用的数据库；③尚未建立数据库。在“表单属性”对话框中，根据不同需求选用的处理方式有：

(1) 将结果发送到某个文件。

(2) 将结果以电子邮件的形式发送。

(3) 发送至某数据库保存。

(4) 发送至其他对象（用户定制脚本来处理数据）。



图 5.53 “表单属性”对话框

5.5.7 框架

框架是组织页面内容、排版的一种方式，用户可以通过使用框架更多地控制浏览器窗口中的页面组合。采用框架设计的网页称为框架集，它们实际上是 HTML 页面的集合。每个框架页（即框架集）由控制框架结构的文件和至少两个同时显示但内容不同的 HTML 文件组成。框架网页的好处之一是各显示页面可分别浏览，它是网页设计中较难、较复杂的内容，可以说理解和掌握了框架网页的设计，就完全掌握了网页制作的全过程。

1. 创建框架

- (1) 选择菜单命令“文件/新建”，打开“新建”任务窗格。
- (2) 单击“其他网页模板”，打开“网页模板”对话框（如图 5.36 所示）。
- (3) 选择“框架网页”选项卡，见图 5.54。



图 5.54 “框架网页”对话框

(4) 在“框架网页”选项卡下，选中所需框架（右下角有框架页样式预览，“预览”上方有针对所选框架模板的布局说明），单击“确定”按钮，如图 5.55 所示选择的是“横幅和目录”框架页面，它是网站中使用最多的一种框架页面形式。

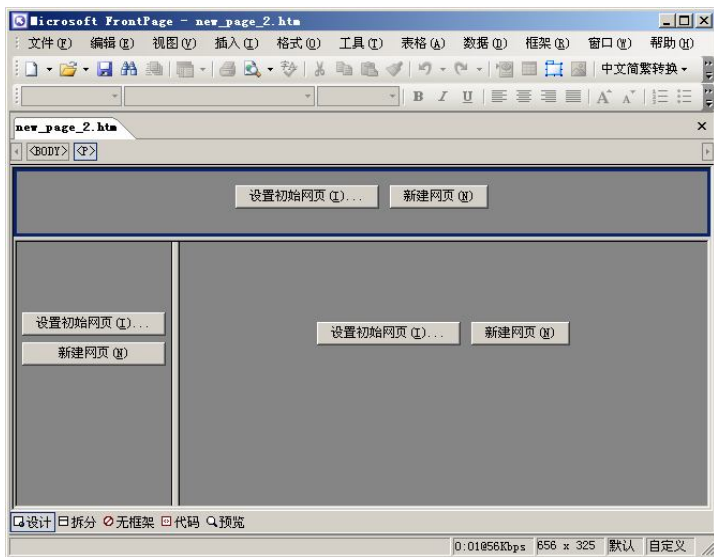


图 5.55 编辑窗口的“框架网页”

(5) 在新建的各框架页部分,“设置初始网页”按钮表示可以使用已有的网页页面,单击弹出“插入超链接”对话框,可以选择链接已存在的页面;使用“新建网页”按钮表示要创建一个新的页面。

选择框架网页时,FrontPage 2003 的编辑窗口下部会多出一个“无框架”视图。“无框架”视图用来创建不带框架的网页,为不支持框架页的浏览器专设;若想编辑、修改各框架页的源程序,可以单击“代码”视图或“拆分”视图(图 5.55 选择的是横幅部分,当前 FrontPage 2003 的操作对象是横幅)。

2. 分割和删除框架

用户可以非常轻松地分割和删除框架,按下述步骤即可完成:

- (1) 选定需要分割或删除的框架。
- (2) 在菜单栏选择“框架/拆分框架”或“删除框架”命令。

3. 存储框架

框架网页是由控制框架结构文件和两个或两个以上其他 HTML 文件组成的。因此,存储框架网页就有存储整个框架结构的框架文档和存储框架内各部分的框架文档两种形式。

- (1) 存储整个框架结构文档。

- 1) 在菜单命令中选取“文件/保存”或“另存为”。

- 2) 输入文件名,单击“保存”按钮。

- (2) 存储部分框架结构文档。


- 1) 在菜单命令中选取“框架/另存为新网页”。

- 2) 在弹出的“另存为”对话框(见图 5.56)中,输入文件名,单击“保存”按钮。

- 3) 还可以从图 5.56 左边的悬浮按钮或选择其他方式保存当前文档的位置。

4. 建立框架链接

在框架网页的任意一个框架内建立超链接,其方法如下:

- 1) 单击按钮;或右击,选择“超链接属性”命令,创建超链接。

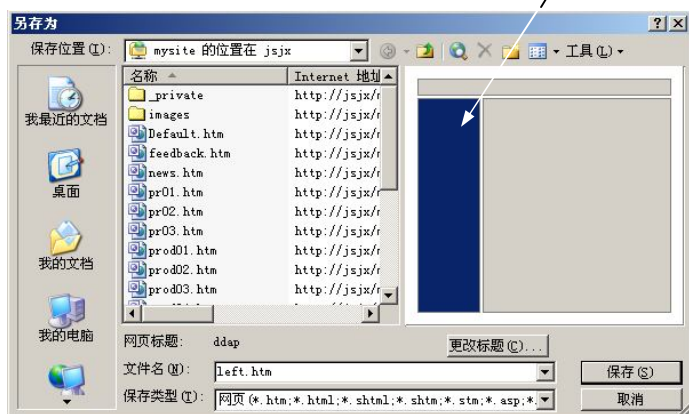


图 5.56 “另存为”对话框

2) 在“插入超链接”对话框内（见图 5.44），单击“目标框架”按钮，在弹出的“目标框架”对话框（见图 5.57）中，单击选择、设定不同框架页的超链接，单击“确定”按钮。

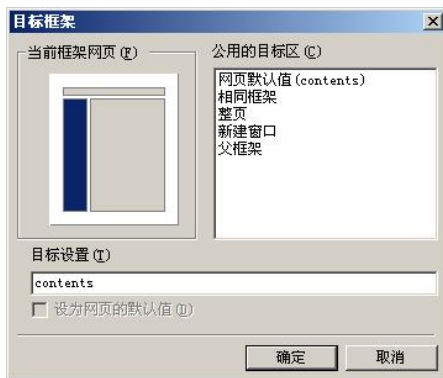


图 5.57 “目标框架”对话框

(3) 在弹出的“插入超链接”对话框内（见图 5.44），建立本站的文件超链接，或在“地址”栏中输入超链接地址，单击“确定”按钮完成。

5. 对框架属性的修改

单击菜单命令“框架/框架属性”，或在任意框架内右击，弹出“框架属性”对话框。该对话框内有名称、初始网页、框架大小、边距等，用户可根据需要选择修改。

6. 为不支持框架的浏览器创建网页

目前的浏览器功能齐全，但为了“顾全大局”FrontPage 2003 给不支持框架的浏览器定制了一种功能，单击“无框架”视图，FrontPage 2003 将显示“此网页使用了框架，但您的浏览器不支持框架。”的空白网页。此功能提供给用户为不支持框架的浏览器设置网页，以保证网页平台与浏览器的无关性。

5.5.8 页面布局与控制

使用工具软件的好处之一是可以非常方便、快捷地排版和布局一帧页面，并对其中的内容进行有效的定位控制。在 FrontPage 2003 中主要可以使用三种方法：

1. 主题

设定网站主题，使网站的所有网页（也可以是单一页面或多个页面）使用一个统一的标准，保持相同的外观，而且在需要改变外观时，只要改变对主题的重新设定就可以完成对所有网页的设定。并且在创建新网页时，自动应用默认主题。主题影响外观的主要方面是颜色、图形、样式。

2. 共享边框

在 FrontPage 2003 中设置共享边框，使网站上一个或多个网页具有共同的页面区域。该区域仅占整个页面的一部分，可以在页面的顶部、底部，或左边、右边。共享边框非常适合做网站页面的横幅（页眉）和版权信息（页脚）。FrontPage 2003 中的共享边框在默认状态下是关闭的，其打开步骤是：

- (1) 选择“工具/网页选项”，打开“网页选项”对话框。
- (2) 单击打开“创作”选项卡，见图 5.58。



图 5.58 选择“共享边框”复选框

(3) 在“FrontPage 和 Sharepoint 技术”选项组中选中“共享边框”多选框，单击“确定”按钮。

(4) 选择打开“共享边框”以后，可以在菜单栏“格式”命令下找到“共享边框”选项。

3. 表格

使用表格布局网页灵活多样，可以将图片、文字等内容分割排列布局，非常适合处理一些细小、定位控制精确的内容，可参见图 8.4。

4. 框架

框架是一种特殊的 HTML 网页，它是将浏览器窗口分成不同的区域，在各个区域中可以显示不同的页面，这是有别于主题和共享边框的最大不同。FrontPage 2003 提供多种框架网页模板，可以有选择地改变或创建框架网页。框架网页通常用于目录、文章或信息列表，创建内置导航栏目，显示一致的浏览界面。

5.5.9 其他功能应用

1. 行为事件

FrontPage 2003 新增“行为”功能，它重新组织、增强了原有的插入 JavaScript 脚本、文本对象模型 (DOM) 与层叠样式表 (CSS) 等功能，增加、增强了系统、浏览器、鼠标事件、设置文本等功能。读者不一定详细了解其技术环节，只需通过选择菜单完成就可以了。

(1) 插入“弹出消息”(浏览器事件，在打开浏览器时激发)。

- 1) 单击菜单栏“格式/行为”命令，打开“行为”任务窗格。
- 2) 单击“插入”下拉菜单，选择“弹出消息”。
- 3) 在“弹出消息”对话框中，输入需要在网页打开时弹出的消息框的文本内容。

(2) 播放声音。

- 1) 在“行为”任务窗格中，单击“插入”下拉菜单，选择“播放声音”。
- 2) 在“播放声音”对话框内填入声音文件，单击“确定”按钮。
- 3) 当触发事件时播放声音。

(3) 设置状态栏的文本。

- 1) 在“行为”任务窗格中，单击“插入”下拉菜单，选择“设置文本”。
- 2) 单击“设置状态栏的文本”。
- 3) 在弹出对话框的消息框中输入需在状态栏显示的文本内容。

(4) 交换图像。

- 1) 在“行为”任务窗格中，单击“插入”下拉菜单，选择“交换图像”。
- 2) 在“交换图像”对话框内填入欲交换的图像文件，单击“确定”按钮。
- 3) 还可以选择“交换图像恢复”，交替显示图像，见例 5.18。

例 5.18 在网页中创建交替显示的动态图像。要求当鼠标移动到该图像上时，图像会变为另外一幅图像，移开鼠标图像复原。

- 1) 准备两幅大小相同的图像(也可以不一样)，并放置于同一文件夹下。
- 2) 将其中任意一幅图像插入网页页面，并选中该图像。
- 3) 在菜单栏中选择“格式/行为”命令，单击“插入”下拉菜单，选择“交换图像”。
- 4) 在弹出的“交换图像”对话框(见图 5.59)中，填入交换图像的地址，单击“确定”按钮。
- 5) 返回到“插入”下拉菜单，选择“交换图像恢复”。
- 6) 在弹出的“交换图像恢复”对话框(见图 5.60)中，单击“确定”按钮，完成。
- 7) 保存文件，并用浏览器打开(或“预览”)，当鼠标移至设定的图像上时，图像立即改变；如果撤离鼠标，图像又会恢复原样。
- 8) 还可以改变行为事件，移动鼠标至“行为”任务窗格中“事件”栏内的任一“事件”上，单击下拉箭头，可以选择不同的行为事件(见图 5.61)。
- 9) 若要删除“交换图像”动态效果，可以在“行为”任务窗格的“事件”栏中，选中动态“事件”效果，单击“删除”按钮。
- 10) 在“行为”任务窗格的“事件”栏中，选中任一“事件”，按 Del 键将删除所有事件。

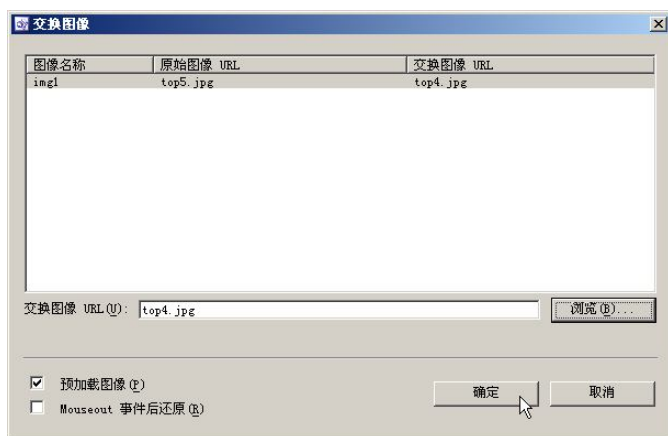


图 5.59 “交换图像”对话框

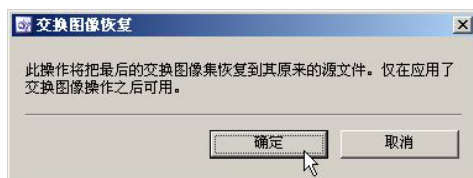


图 5.60 “交换图像恢复”对话框

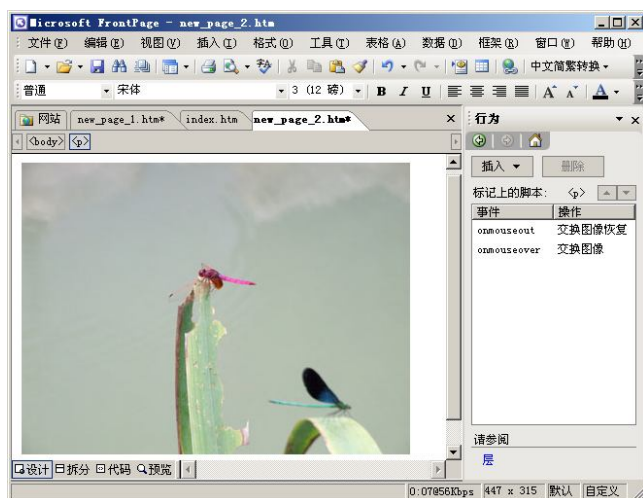


图 5.61 “交换图像恢复”界面

2. Web 组件

可以直接在 FrontPage 2003 的页面中插入 DHTML (Dynamic HTML) 动态效果、音乐、Flash 动画等, FrontPage 2003 的“Web 组件”内容丰富, 本书只介绍以下几点。

(1) 动态文字。

- 1) 单击菜单栏“插入/Web 组件”, 打开“插入 Web 组件”对话框 (见图 5.62)。
- 2) 在“插入 Web 组件”对话框的“组件类型”中选择“动态效果”。
- 3) 在“选择一种效果”栏内选择“字幕”, 单击“完成”按钮。

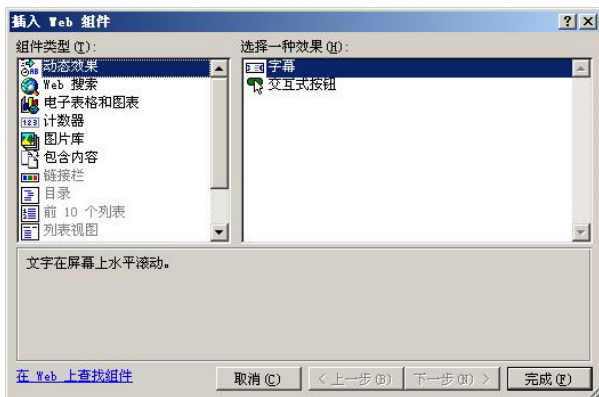


图 5.62 “插入 Web 组件”对话框

(2) 嵌入音乐。

1) 打开“插入 Web 组件”对话框（见图 5.62）。

2) 在“插入 Web 组件”对话框的“组件类型”中选择“高级控件”。

3) 双击“插件”，在弹出的“控件属性”对话框的“数据源”栏内输入音乐文件。

(3) 插入背景音乐。

1) 在菜单栏中选择“文件/属性”（或在页面上右击，选中“网页属性”）。

2) 在弹出的“网页属性”对话框中选中“常规”选项卡。

3) 在“背景音乐”栏下单击“浏览”按钮，选取所需的音乐文件。

4) 单击“确定”按钮完成。

5) “嵌入音乐”和“背景音乐”可以使用多种音乐文件类型，如 wav、midi、ra/ram、aiff、au 以及 mp3 等，由于网络带宽的原因，推荐在网页中尽量使用较小的音乐文件；以上两种方法使用的标识不一样，前者使用<EMBED>，后者使用<BGSOUND>，可以通过“代码”视图了解。

(4) 嵌入 Flash 影片。

1) 打开“插入 Web 组件”对话框（见图 5.62）。

2) 在“插入 Web 组件”对话框的“组件类型”中选择“高级控件”。

3) 双击“Flash 影片”，在弹出的“选择文件”对话框中输入 Flash 文件。

5.6 网站中的脚本语言

脚本语言的使用，使我们有了客户端的“可执行”程序和事件驱动的软件开发环境；可以使数据发送到服务器之前先进行处理和校验，动态地创建新的 Web 内容。之所以称为脚本语言，是由于它不是一个完整的语言系统，只能“寄宿”在 HTML 之上，离开 HTML，脚本语言就无法“生存”了。脚本语言不但可以运行在客户端，同时可以运行在服务器端。

在 Web 页面中，可以同时使用 VBScript 和 JavaScript 两种或其他多种脚本语言，IE 浏览器提供对上述两种脚本语言的支持。VBScript 和 JavaScript 有许多相似之处，但也存在一些差异。

VBScript 既不能用来建立对象，也不支持用户定义类，它只能支持可重用的子程序过程和函数过程。JavaScript 利用对象的能力比 VBScript 要强一些，可以使用 JavaScript 定义类，

可以说 JavaScript 是一种基于对象（Object-Based）的脚本语言。

5.6.1 VBScript 脚本语言

脚本语言 VBScript 的最大优点是易于使用，直接包含在 HTML 文档中。在 HTML 内插入<SCRIPT></SCRIPT>标识，在其之间写上脚本语言 Script 规定的代码，用纯 ASCII 文本建立文件（其扩展名仍然是.htm）。打开此文件，浏览器便能根据脚本语言的要求进行解释。VBScript 的编写和修改十分便利，对于有 Visual Basic 语言使用经验的人，可以非常容易地掌握和理解 VBScript 脚本语言。见例 5.19。

例 5.19 从输入/输出看 VBScript（L5-19.htm）。

```
01:<HTML>
02:<HEAD><TITLE>VBScript 的输入/输出</TITLE></HEAD>
03:<BODY>
04:<SCRIPT language="VBScript">
05:  <!-- 对旧版本的浏览器将注释掉第 6、7 行的脚本语句
06:    MyCar=InputBox("请您选择车型: ", " VBScript 的输入/输出")
07:    ShMyCar=MsgBox("您选的车型是: " & VbCrLf & MyCar)
08:  -->
09:</SCRIPT>
10:</BODY>
11:</HTML>
```

说明：编辑本例，并在浏览器中运行。在弹出的对话框（即输入框，见图 5.63）中，输入数据“新宝来 1.6L 自动豪华”，单击“确定”按钮，弹出输出结果信息窗（即输出框，见图 5.64）。InputBox 是输入函数，MsgBox 是输出函数。



图 5.63 输入对话框

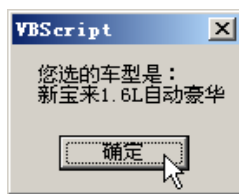


图 5.64 输出结果

其他相关说明如下：

- (1) 行号不是程序的组成部分，只为解释程序而设。
- (2) 第 4~9 行是脚本语言，它不是一种独立的语言，必须依附 HTML。
- (3) 作为 VBScript 脚本语言的源程序存盘时，其扩展名仍然是.htm 或.html。
- (4) 通常将脚本放在头部<HEAD></HEAD>或正文部分<BODY></BODY>中，用<SCRIPT>和</SCRIPT>做标记，浏览器能根据脚本语言的要求进行解释，但也允许将脚本放在 HTML 文档<HTML></HTML>的外面；
- (5) <SCRIPT>标识的含义是在 HTML 文档中插入浏览器端脚本。其中的 language 是<SCRIPT>的属性，用来指定使用何种脚本语言，这里默认的脚本语言是 JavaScript。
- (6) 可以用任何一种具有文本编辑功能的编辑器编写 VBScript 源程序。
- (7) 作为过程或函数定义时，VBScript 一般都放在<HEAD></HEAD>中。

(8) 所有的脚本语言最好能集中放置，这样可以确保在 BODY 部分调用代码之前，脚本语言的代码全被读取和解释。

(9) 第 6 行的 `VbCrLf` 是 VBScript 的符号常量。输出信息需要换行时，应在行末增加回车换行代码 `VbCrLf`，并使用连接符 “+” 或 “&”。`MsgBox` 还可以作为语句使用。

5.6.2 VBScript 应用

这里主要从过程、事件、对象三个方面介绍 VBScript 的应用。VBScript 的过程分为两类：子程序过程 (Sub 过程) 和函数过程 (Function 过程)。通常我们将 Sub 过程与鼠标事件 `OnClick` 结合在一起使用，这种由事件激活的过程称为事件过程。事件过程是当激活某个事件（如 `OnClick`）时，该事件做出的响应，即执行所指定的程序模块。这种事件过程构成了 VBScript 应用程序的主体。所谓事件 (Events)，是由 VBScript 预先设计好的、能够被对象识别的动作。对于对象而言，事件就是发生在该对象上的事情。在响应事件时，事件驱动应用程序执行指定的代码。IE 浏览器的每个对象都有一个预定义的事件集，当其中的某个事件发生，并且在相关联的事件过程中存在代码时，IE 浏览器将执行这些代码。

在 HTML 中，可以将 VBScript 语句定义为事件处理过程 (Event Handler)，然后通过事件来驱动。事件就是用户在浏览器上所执行的一些操作，例如单击、双击鼠标等。浏览器可以识别一些特定的事件，并在事件处理过程中自行定义适当的 VBScript 程序，处理必要的事件，表 5.2 列出了常用的 VBScript 事件。

表 5.2 VBScript 的事件

事件		注释
键 盘 事件	OnKeyDown	当一个键被按下时激发
	OnKeyPress	当用户的键盘输入被转换为一个字符时激发
	OnKeyUp	当用户释放一个键时激发
鼠 标 事件	OnClick	当用户单击一个元素或按 ENTER 键时激发
	OnDbclick	当用户双击一个元素时激发
	OnMouseOut	当鼠标在元素之间移动时，首先激发 OnMouseOut 事件，表示鼠标已离开原来元素。然后激发 OnMouseMove 事件，表示鼠标已经移动。最后，激发 OnMouseOver 事件，表示鼠标已经进入到新的元素。OnMouseDown 和 OnMouseUp 分别表示鼠标按下和离开时激发
	OnMouseMove	
	OnMouseOver	
	OnMouseDown	
其 他 事件	OnMouseUp	
	OnChange	当用户按 Tab 键从一个元素离开或在一个元素上按 ENTER 键，离开该键时 OnChange 事件激发
	OnError	当加载一个图像或其他元素过程中发生一个错误时激发，或当一个脚本错误发生时激发
	OnLoad	在文档被加载和页面上所有元素被完全下载后激发
	OnScroll	无论什么时候页面的滚动框或其中的任何元素被重新定位时激发
	OnUnload	在文档将被卸载之前或当漫游到另外的文档时立即激发

例 5.20 利用 Sub 子程序过程和 Function 函数过程调用, 求给定半径的球体体积; 并求该球内接立方体的边长和体积以及两者的体积差 (L5-20.htm)。

```

01:<HTML>
02:<HEAD><TITLE>Sub 过程、Function 过程、带参调用</TITLE>
03:<CENTER>Sub 过程、Function 过程、函数的带参调用</CENTER>
04:<SCRIPT language="VBScript">
05:    Private Sub BallV()                输入球的半径, 并调用计算的过程
06:        r=InputBox("请输入球的半径: ")
07:        If r=False Then
08:            MsgBox "您取消了本次操作!",16
09:            Exit Sub
10:        ElseIf r=True Then
11:            MsgBox "您未输入数据!",48
12:            Exit Sub
13:        End If
14:        Ball=4/3*3.14*r*r*r
15:        DiV=Ball-Bianc(r)*Bianc(r)*Bianc(r) '调用函数过程 Bianc()求差值, r 为实参
16:        MsgBox"球的半径是: " & r & ", 球的体积是: " & Ball & VbCrLf & VbCrLf & _
17:            "立方体的边长是: " & Left(Bianc(r),5) & ", 立方体的体积是: " & _
18:            Left(Bianc(r)*Bianc(r)*Bianc(r),8) & VbCrLf & VbCrLf & _
19:            "两者体积差是: " & DiV
20:    End Sub
21:
22:    Function Bianc(radius)                '计算立方体边长的过程, radius 为形参
23:        Bianc=sqr(2)*radius              'sqr()是求平方根的函数
24:    End Function
25:    BallV                                '调用过程 BallV
26:</SCRIPT>
27:</HEAD><BODY></BODY>
28:</HTML>

```

说明: 本例中既用到了 Sub 过程, 又用到了 Function 过程, 并且使用了函数的参数调用。使用 Function 过程求出立方体的边长, 用 Sub 过程计算其他值。Bianc 函数过程求出正方形的边长。Sub 子程序过程 BallV 则调用此函数, 调用时, BallV 的实际参数 r 将被传递给 Bianc 函数, 并作为它的形式参数 radius 参与运算。

例 5.21 在 VBScript 中使用事件处理过程的 4 种方法 (L5-21.htm)。

```

<HTML>
<HEAD><TITLE>4 种事件处理</TITLE>
<SCRIPT language="VBScript">
    Sub Event1_OnClick
        MsgBox "①使用事件处理过程名",48
    End Sub
</SCRIPT>
<SCRIPT language="VBScript">
    Sub Event2
        MsgBox"②使用 HTML 控件中的属性",48
    End Sub

```

```

End Sub
</SCRIPT>
<SCRIPT language="VBScript" for="Event4" event="OnClick">
    MsgBox "④使用 FOR/EVENT 属性控制",48
</SCRIPT>
</HEAD>
<BODY>
<INPUT name="Event1" type="button" value="事件处理过程①, 请单击"><P>
<INPUT type="button" value="事件处理过程②, 请单击" OnClick='Event2()'><P>
<INPUT type="button" value="事件处理过程③, 请单击"
    OnClick=MsgBox "③使用内联事件控制",48' Language="VBScript"><P>
<INPUT name="Event4" type="button" value="事件处理过程④, 请单击">
</BODY>
</HTML>

```

说明：执行本例，弹出 4 个按钮，如图 5.65 所示，单击按钮，处理的分别是例题中说明的 4 种事件。Event1_OnClick 是事件处理过程名。通用过程名 Event2 作为属性 OnClick 的属性值，供控件的属性调用。注意，过程名应放在单引号内，此例的 Event2() 有无圆括号均可。事件处理过程③是在<INPUT>标识中完成的，它将事件名 OnClick 作为标识的属性来对待。该属性的值放在一对单引号内，当激活 OnClick 事件时，执行单引号内的语句等。需注意在 HTML 文档中，属性值通常用双引号括起来，此处的属性值只能用单引号。使用这种事件过程，事件的属性值还可以包含多个语句，但是各个语句之间必须用冒号分隔。

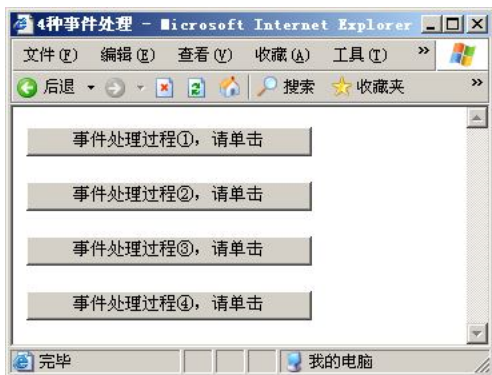


图 5.65 事件处理过程的 4 种方法

事件处理过程④直接在<SCRIPT>标识内用 for 和 event 指定与脚本程序段相联系的对象和事件。for="Event4"和 event="OnClick"在程序中的含义是：当名为 Event4 的按钮驱动 OnClick 事件后，程序将执行<SCRIPT></SCRIPT>标识中的脚本。

例 5.22 Window、Document 对象的属性、方法应用 (L5-22.htm)。

```

01:<HTML>
02:<HEAD><TITLE>Window、Document 对象的属性、方法</TITLE></HEAD>
03:  <A name="n1" href=http://www.waterpub.com.cn>水利水电出版社</A><P>
04:  <A name="n2" href=http://www.lhx.com>绿海星汽车贸易公司</A><P>
05:  <A name="n3" href=http://www.baidu.com>百度搜索</A><P>
06:  <A name="n4" href=http://www.google.com>谷歌搜索</A><P>

```

```

07:<SCRIPT language="VBScript">
08:  Window.Name="windo"           '将窗口名 windo 赋给 Window 对象的 Name 属性
09:  With Document                  '使用 With 语句
10: '使用 Document 对象的属性
11:  .bgColor="#A0A0A0"             '设置背景色
12:  .LinkColor="#FF0000"           '设置超链接的颜色
13:  .AlinkColor="#00FF00"          '设置单击超链接时的颜色
14:  .vLinkColor="#0000FF"          '设置单击超链接以后的颜色
15:'使用 Document 对象的方法
16:  .Write "<B>Links 属性输出本页超链接的个数: </B>" &_
17:      Document.Links.Length & "<P>"
18:  .Write "<B>Anchors 属性输出本页超链接的个数: </B>" &_
19:      Document.Anchors.Length & "<P>"
20:  .Write "<B>Title 属性输出本网页的标题: </B>" &_
21:      Document.Title & "<P>"
22:  .Write "<PRE>"                 '预格式文本标识
23:  .Write "<H2>两幅自勉联</H2>"
24:  .WriteLn "发愤识遍天下字, (WriteLn 方法)"
25:  .WriteLn "立志读尽人间书。(WriteLn 方法)"
26:  .Write "</PRE>"
27:  .WriteLn "板凳要坐十年冷, (没能实现 WriteLn 方法)"
28:  .WriteLn "文章不写一句空。(没能实现 WriteLn 方法)"
29:  End With                       'With 语句结束
30:'使用 Window 对象的 open 方法打开窗口
31:  Window.Open "http://www.cctv.com", "windo1", "toolbar=yes, location=no,
32:      status=yes,menubar=no, resizable=1, width=450, height=300"
33:  Window.Open "L5-1.htm", "windo2", "toolbar=yes, location=no, status=yes,
34:      menubar=no,resizable=1, width=500, height=250"
35:  winName=Window.Name            '使用 Window 对象的 Name 属性, 获取窗口名
36:'使用 Window 对象的 Status 属性, 在浏览器的状态栏中显示指定信息
37:  Window.Status="今天是: "&date&" 时间是: "&time 'date 和 time 是 VBScript 的函数
38:'使用 Window 对象的 Alert 方法输出
39:  Window.Alert "本窗口的名称为: " & winName
40:  CloseWin=MsgBox("是否关闭弹出窗?",32+1)
41:  If CloseWin=VbOk Then
42:      Window.Close                '使用 Window 对象的 Close 方法关闭当前窗口
43:  End If
44:</SCRIPT>
45:</HEAD>
46:</HTML>

```

说明: 本例综合使用了 Window、Document 对象的属性、方法, 有关详细内容可以查阅 MSDN 的帮助文档或其他参考书籍。第 9 行的 With 语句使代码更为简洁, 但要注意它的使用格式, 也可以在 JavaScript 中使用 With 语句, 但要注意英文字母的大小写。第 31、32 行是一条语句, 输入源代码时不要换行, 否则会出错; 第 33、34 行同理。执行第 37 行, 在状态栏显示时间、日期, 图 5.66 由于正处在单击超链接的时刻, 所以状态栏中显示的是超链接的网址。还有几点注意:

- 使用 Anchors 属性时, <A>标识中必须含有 name 或 id 属性, 否则 Anchors 取不到值。
- 使用 Links 属性时, 无需遵守上条要求。
- <PRE>是预格式文本标识, 若不用它, 则 WriteLn 不会换行, 比较第 24、25 行与第 27、28 行, 后者没有执行换行操作。
- 使用 Window 对象的 Open 方法打开窗口, 第一个参数是 URL, 第二个参数是窗口名称, 若打开的窗口与当前窗口不同名, 则打开新窗口执行 URL 指定的页面。
- 可以使用 Window 对象的 Close 方法关闭当前窗口。



图 5.66 使用 Window 和 Document 对象

例 5.23 VBScript 中函数的使用, 运行效果见图 5.67 (L5-23.htm)。

```
<HTML>
<HEAD><TITLE>使用切分函数切分字符串</TITLE></HEAD>
<BODY>
<SCRIPT language="VBScript">
    Public MyArray           '声明公共变量
    MyCp="丁香花 百头 千头 万头 "
    Document.Write ("您要找的下联 (拆字联) 在此: ") & MyCp & "<P>"
    Document.Write ("用空格切分后为: ") & "<BR>"
    MyArray=split(MyCp," ")   'split 是 VBScript 的函数
    MaxNo=ubound(MyArray)
    For i=0 to MaxNo-1
        Document.Write (MyArray(i)) & "<BR>"
    Next
</SCRIPT>
</BODY>
</HTML>
```

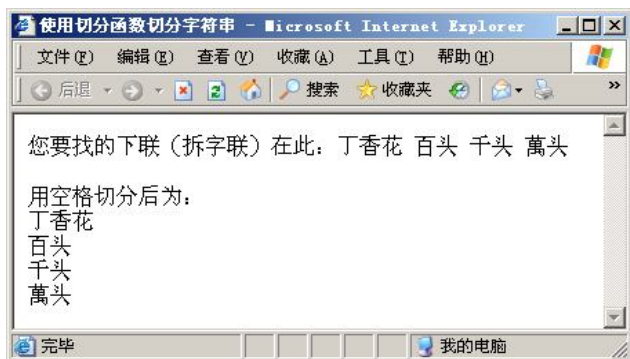


图 5.67 使用 split 函数切分字符串

说明：该例是将一个字符串赋值给一个变量，split 函数的作用是将一个字符串切分成数个字符串，并返回由这些子字符串组成的一维数组，本例以空格切分字符串。注意汉、英切分字符串的区别（特别是“拆字联”两端的圆括号，应该是汉字字符），而且在字符串 MyCp 的最后还应加一个空格，若不加此空格，则不能显示“万头”。ubound()函数是返回数组某一维的最大边界值。

5.6.3 JavaScript 脚本语言

JavaScript 是一种高级的脚本描述语言，不依赖于特定的机器和操作系统，而且独立于操作系统平台，设计它的目标是为了尽可能少地占用系统和网络资源。其原名叫 LiveScript。Netscape 公司在引入 Sun 公司有关 Java 的程序设计概念后，重新作了设计并更名为 JavaScript，其大部分语法与 Java 相似。

JavaScript 与 VBScript 都是解释性语言，由浏览器解释完成，不需要对它们进行编译等操作。对于 JavaScript 有两点必须说明：第一，Microsoft 公司在 Netscape 公司发布的 JavaScript 基础上，独立开发了自己的 JavaScript 语言版本 JScript。JScript 除了可以实现 JavaScript 的所有功能以外，还增加了许多功能。第二，不同的浏览器对脚本语言或其部分功能的支持可能有所不同，但高版本的浏览器（如 IE 5.0 以上、Netscape 6.0 以上）对脚本语言的支持要好得多。一般来说，Netscape (NS) 浏览器对 JavaScript 的支持要强一些。JavaScript 和 VBScript 同属于脚本语言，在许多地方非常类似。和 VBScript 一样，JavaScript 同样是一种嵌入 HTML 文档的，以事件驱动的程序设计语言。但是它是基于对象的脚本程序设计语言，不依赖于具体的计算机平台（虽然有一些限制），只与解释它的浏览器有关，它更像一种功能强大的程序设计语言。JavaScript 的操作符几乎和 C 语言完全一致，语法、语句也非常类似。

例 5.24 在 JavaScript 中使用 window 对象的 alert、confirm、prompt、close、find 方法和 document 对象的 write 方法完成输入、输出（L5-24.htm）。

```
<HTML>
<HEAD>
<TITLE>JavaScript 中使用 alert、confirm 和 prompt</TITLE>
<SCRIPT language="JavaScript">
    char1=window.confirm("单击「确定」继续，「取消」终止。");
    if(char1)
        {var char2=window.prompt("请输入购车型号：","");
```

```

    if (char2=="")
        window.alert("您没有输入数据！");
    else
        window.alert("您输入的是: "+char2);}
else
    {window.alert("您取消了本次操作！");
    window.close();}
document.write("<H2>您使用的是 JavaScript 对象的方法。</H2>");
document.write("若浏览器状态栏显示‘网页上有错误’或其他，请使用 Netscape。");
window.find("Script",true,true);           //IE 浏览器不支持 find 方法
</SCRIPT>
</HEAD>
</HTML>

```

说明：**confirm** 方法弹出一个确认对话框，**prompt** 方法为输入数据弹出对话框，**alert** 方法可以为输出数据弹出一个提示框，**close** 方法关闭当前浏览器窗口，**find** 方法是在当前页面中查找指定的字符串，如"Script"（这个方法 IE 浏览器不适用，查找结果见图 5.68）；**write** 方法与 VBScript 下 **document** 对象的 **write** 方法相同。注意 JavaScript 对大小写字母敏感。



图 5.68 find 方法查找的结果

5.6.4 JavaScript 应用

JavaScript 的代码由一行中的一个或多个符号组成，每一行开始一条新的语句，分号 (;) 是 JavaScript 的终止字符；最好是在每一条语句结束时使用分号。见例 5.24。用大括号 “{}” 括起来的一组 JavaScript 代码行称为一个语句块。在函数和条件语句中常常使用大括号将特定的代码行作为一个语句块括起来。下面从 JavaScript 的函数、事件、对象三个方面，介绍它的应用。

例 5.25 应用内置函数计算数据，并检测结果 (L5-25.htm)。

```

01:<HTML>
02:<HEAD>
03:<TITLE>应用内置函数</TITLE>
04:<SCRIPT language ="JavaScript">
05:    function push1(dx1)
06:        {dx1.pull1.value=eval(dx1.bds1.value)}
07:    function push2(dx2)
08:        {dx2.pull2.value=parseFloat(dx2.bds2.value)}

```

```

09:         if (isNaN(parseFloat(dx2.bds2.value)))
10:             {alert("非数值型数据返回 NaN! ");}
11:     }
12:     function push3(dx3)
13:     {dx3.pull3.value=parseInt(dx3.bds3.value)
14:     if (isNaN(parseInt(dx3.bds3.value)))
15:         {alert("非数值型数据返回 NaN! ");}
16:     }
17:     function push4(dx4)
18:     {dx4.pull4.value=isFinite(dx4.bds4.value)}
19:     </SCRIPT>
20:</HEAD>
21:<BODY>
22:<H2>应用内置函数</H2>
23:<FORM name="sheet1">
24:    eval 函数运算: <INPUT type="text" name="bds1" size="20">
25:    <INPUT type="button" value="执行结果" onclick="push1(this.form)">
26:    <INPUT type="text" name="pull1" size="10"></P>
27:</FORM>
28:<FORM name="sheet2">
29:    parseFloat 函数运算: <INPUT type="text" name="bds2" size="20">
30:    <INPUT type="button" value="执行结果" onclick="push2(this.form)">
31:    <INPUT type="text" name="pull2" size="10"></P>
32:</FORM>
33:<FORM name="sheet3">
34:    parseInt 函数运算: <INPUT type="text" name="bds3" size="22">
35:    <INPUT type="button" value="执行结果" onclick="push3(this.form)">
36:    <INPUT type="text" name="pull3" size="10"></P>
37:</FORM>
38:<FORM name="sheet4">
39:    isFinite 函数运算: <INPUT type="text" name="bds4" size="22">
40:    <INPUT type="button" value="执行结果" onclick="push4(this.form)">
41:    <INPUT type="text" name="pull4" size="10"></P>
42:</FORM>
43:</BODY>
44:</HTML>

```

说明：第 5、7、12、17 行是 4 个 JavaScript 自定义函数，分别获取对应表单对象的数据。在 4 个文本框内输入数据，单击各自的“执行结果”按钮，输出计算结果。isFinite 函数认为，除了数值型数据以外的任何数据均返回 false，数值型表达式也不例外，见图 5.69。4 个自定义函数分为 4 组运算分别执行不同的内置函数，由鼠标事件激活，见第 25、30、35、40 行，比较例 5.21 中“事件处理过程②”的语句形式。如果在 parseFloat 或 parseInt 函数中输入非数值型数据将有警示窗弹出。



图 5.69 内置函数应用

例 5.26 当触发鼠标事件 click 时, 可以将 event 对象的属性值传送到事件处理函数中, 同时按下 Shift 键并单击鼠标的结果见图 5.70 (L5-26.htm)。

```
<HTML>
<HEAD>
<TITLE>获取 event 对象的属性值(适用 NS 浏览器)</TITLE>
<SCRIPT language="JavaScript">
    function getEvent(event){
        document.write("<H3>当触发 click 事件时, 事件处理函数获取的 event 对象属性值: </H3>");
        document.write("获取的触发事件类型是: "+event.type+"<BR>");
        document.write("鼠标相对于当前层的水平位置 X 是: "+event.layerX+"<BR>");
        document.write("鼠标相对于当前层的垂直位置 Y 是: "+event.layerY+"<BR>");
        document.write("鼠标相对于当前页的水平位置 X 是: "+event.pageX+"<BR>");
        document.write("鼠标相对于当前页的垂直位置 Y 是: "+event.pageY+"<BR>");
        document.write("鼠标相对于当前屏幕左上角的水平位置 X 是: "+event.screenX+"<BR>");
        document.write("鼠标相对于当前屏幕左上角的垂直位置 Y 是: "+event.screenY+"<BR>");
        switch (event.which)      //返回按下鼠标键的信息
        {case 1: document.write("您按下的是鼠标左键, 返回值为 1. <BR>");break;
        case 2: document.write("您按下的是鼠标中间键, 返回值为 2. <BR>");break;
        case 3: document.write("您按下的是鼠标右键, 返回值为 3. <BR>");break;
        default: document.write("按键有误<BR>");
        }
        if(event.modifiers & Event.SHIFT_MASK)
            alert("哈哈! 您在单击鼠标时, 肯定按下了 Shift 键!");
        return true;
    }
    document.write("***请在屏幕的任意位置点鼠标或按下 Shift 键时点击鼠标***")
    document.onmousedown=getEvent;
</SCRIPT>
</HEAD>
</HTML>
```

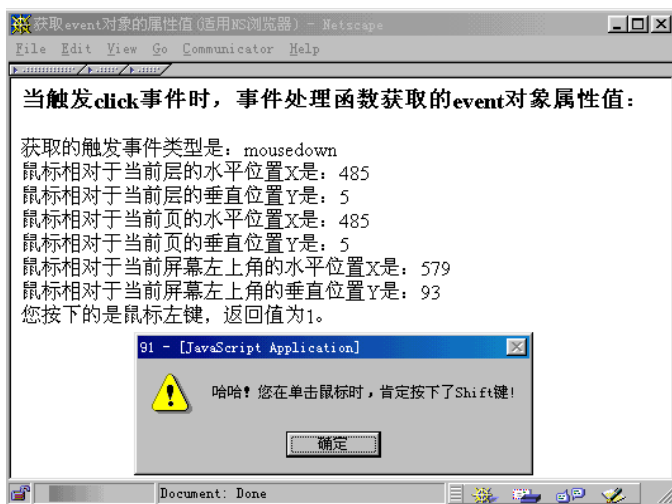


图 5.70 event 对象的部分属性值

说明：NS 和 IE 浏览器对 JavaScript 的支持有所不同，本例适用于 NS 浏览器。请用 Netscape 4.0 以上的版本，但 Netscape 7.2 不支持 Event.SHIFT_MASK，注意英文字母的大、小写。event 对象提供对 JavaScript 事件的各种处理信息，本例在鼠标事件发生时，解释器会自动把事件信息填充到 event 对象的对象实例中，并作为一个参数传送给鼠标事件处理函数。

例 5.27 在 IE 浏览器下获取 event 对象属性值 (L5-27.htm)。

```
<HTML>
<HEAD>
<TITLE>获取 event 对象的属性值(适用于 IE 浏览器)</TITLE>
<SCRIPT language="JavaScript">
    function getEvent(et){
        document.write("<H3>当触发 click 事件时，事件处理函数获取的 event 对象属性值： </H3>");
        document.write("获取的触发事件类型是： "+event.type+"<BR>");
        document.write("鼠标相对于当前窗口的水平位置 X 是： "+event.clientX+"<BR>");
        document.write("鼠标相对于当前窗口的垂直位置 Y 是： "+event.clientY+"<BR>");
        document.write("鼠标相对于当前屏幕左上角的水平位置 X 是： "+event.screenX+"<BR>");
        document.write("鼠标相对于当前屏幕左上角的垂直位置 Y 是： "+event.screenY+"<BR>");
        if(event.altKey)
            alert("哈哈！您在单击鼠标时，肯定按下了 Alt 键!");
        return true;
    }
    document.onmousedown=getEvent;
</SCRIPT>
</HEAD>
<BODY>
***请在屏幕的任意位置单击鼠标或按下 Atl 键时单击鼠标***
</BODY>
</HTML>
```

说明：IE 与 NS 支持 event 对象的属性有所不同，本例适用于 IE 浏览器。

例 5.28 当触发键盘事件时，事件处理函数将获得 event 对象属性值 (L5-28.htm)。

```

<HTML>
<HEAD>
<TITLE>获取 event 对象的属性值(适用 IE 浏览器)</TITLE>
<SCRIPT language="JavaScript">
    function getEvent() {
        document.write("<H3>当键盘事件发生时，事件处理函数获取的 event 对象属性值：</H3>");
        document.write("获取的触发事件类型是：" + event.type + "<BR>");
        document.write("鼠标相对于当前窗口的水平位置 X 是：" + event.clientX + "<BR>");
        document.write("鼠标相对于当前窗口的垂直位置 Y 是：" + event.clientY + "<BR>");
        document.write("鼠标相对于当前屏幕左上角的水平位置 X 是：" + event.screenX + "<BR>");
        document.write("鼠标相对于当前屏幕左上角的垂直位置 Y 是：" + event.screenY + "<BR>");
        if(event.keydown != "")
            alert("您按键的返回键值是：" + event.keyCode);
        return true;
    }
    document.write("***将鼠标放在屏幕的任意位置，在键盘上按任意键***")
    document.onkeydown=getEvent;
</SCRIPT>
</HEAD>
</HTML>

```

说明：本例使用 IE 浏览器。JavaScript 中键盘事件发生时，解释器会自动把事件信息填充到 event 对象的对象实例中，并作为一个参数传送给键盘事件处理函数。在键盘上按任意键触发事件，并输出该键的键值。

例 5.29 时间日期对象的应用，运行结果见图 5.71 (L5-29.htm)。

```

01:<SCRIPT language="JavaScript">
02:    var timeStr,dateStr;
03:    function myclock()
04:        {now=new Date();                                //取当前时间、日期
05:        hours=now.getHours();
06:        minutes=now.getMinutes();
07:        seconds=now.getSeconds();
08:        timeStr="" + hours;
09:        timeStr+=((minutes < 10)? ":0" : ":") + minutes;    //条件操作运算
10:        timeStr+=((seconds < 10)? ":0" : ":") + seconds;
11:        document.myclock.mytime.value=timeStr;
12:        date=now.getDate();
13:        month=now.getMonth()+1;
14:        year=now.getYear();
15:        dateStr="" + month;
16:        dateStr+=((date < 10) ? "/0" : "/" ) + date;
17:        dateStr+="/" + year;
18:        document.myclock.mydate.value=dateStr;
19:        setTimeout("myclock()",1000);
20:    }
21:</SCRIPT>
22:</HEAD>

```

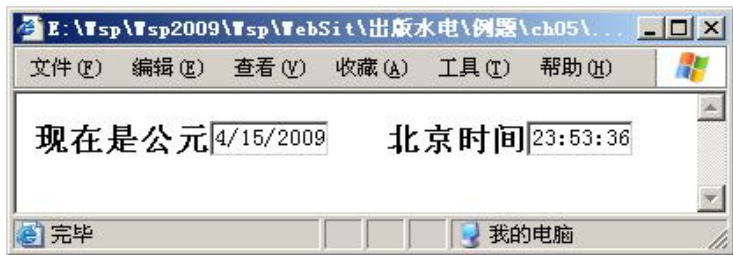
[illegible]

图 5.71 Date 对象应用

说明：在 IE 和 NS 中显示的时间、日期值稍有不同。Date 对象处理日期和时间的存储、转化和表达。第 9、10、16 行使用了 JavaScript 条件操作符，其语法结构是“(表达式)?[语句 1]:[语句 2]”，又称为三元操作符，即表达式为 true 时，执行语句 1；为 false 时，执行语句 2。getFullYear、getMonth、getDate、getHours、getMinutes、getSeconds 分别是 Date 对象获得当前年、月、日、时、分、秒的方法。

例 5.30 使用 JavaScript 脚本语言，在 HTML 文档中调用 XML 数据 (L5-30.htm)。

```

01:<HTML>
02:<HEAD>
03:<SCRIPT language="JavaScript" for="window" event="onload">
04:    var xmlCars=new ActiveXObject("Microsoft.XMLDOM");    /* 使用.NET 的 XMLDOM 对象 */
05:    xmlCars.async="false";                                /* 不允许异步下载 */
06:    xmlCars.load("L5-29.xml");
07:    nodes=xmlCars.documentElement.childNodes;              /* 确认根节点, 传回节点清单 */
08:    cname.innerText=nodes.item(0).text;
09:    cseri.innerText=nodes.item(1).text;
10:    ctype.innerText=nodes.item(2).text;
11:    carea.innerText=nodes.item(3).text;
12:    price.innerText=nodes.item(4).text;
13:    cdate.innerText=nodes.item(5).text;
14:</SCRIPT>
15:<TITLE>调用 XML 数据</TITLE>
16:</HEAD>
17:<BODY bgcolor="#FFFFFF">
18:    <B>品牌: </B><SPAN id="cname"></SPAN><BR>
19:    <B>系列: </B><SPAN id="cseri"></SPAN><BR>
20:    <B>型号: </B><SPAN id="ctype"></SPAN><BR>
21:    <B>产地: </B><SPAN id="carea"></SPAN><BR>
22:    <B>总价: </B><SPAN id="price"></SPAN><BR>

```

```

23:  <B>出厂日期: </B><SPAN id="cdate"></SPAN><BR>
24:</BODY>
25:</HTML>

```

说明：本例使用了.NET的XMLDOM对象（其他细节请参阅相关书籍），并用JavaScript脚本语言实现在HTML中调用XML数据。也可以使用汉字作为对象名，对应例5.30的语句如下（相应例5.31的语句也要改变，请见例5.31的说明）：

```

08:  品牌.innerText=nodes.item(0).text;
09:  系列.innerText=nodes.item(1).text;
10:  型号.innerText=nodes.item(2).text;
11:  产地.innerText=nodes.item(3).text;
12:  总价.innerText=nodes.item(4).text;
13:  出厂日期.innerText=nodes.item(5).text;

18:  <B>品牌: </B><SPAN id="品牌"></SPAN><BR>
19:  <B>系列: </B><SPAN id="系列"></SPAN><BR>
20:  <B>型号: </B><SPAN id="型号"></SPAN><BR>
21:  <B>产地: </B><SPAN id="产地"></SPAN><BR>
22:  <B>总价: </B><SPAN id="总价"></SPAN><BR>
23:  <B>出厂日期: </B><SPAN id="出厂日期"></SPAN><BR>

```

例 5.31 对应例 5.30 的 XML 数据 (L5-29.xml)。

```

<?xml version="1.0" encoding="GB2312"?>
<cars>
  <cname>一汽大众</cname>
  <cseri>新宝来</cseri>
  <ctype>1.6L 自动豪华</ctype>
  <carea>长春</carea>
  <price>10.7 万元</price>
  <cdate>2008.9</cdate>
</cars>

```

说明：若将例 5.30 的两段语句（第 8~13 行和第 18~23 行）换成汉字描述，则本例对应的标识元素也要作相应的变换。其对应程序如下：

```

<?xml version="1.0" encoding="GB2312"?>
  <轿车>
    <品牌>一汽大众</品牌>
    <系列>新宝来</系列>
    <型号>1.6L 自动豪华</型号>
    <产地>长春</产地>
    <总价>10.7 万元</总价>
    <出厂日期>2008.9</出厂日期>
  </轿车>

```

5.6.5 脚本语言技巧

例 5.32 程序出错处理方法（对预定义常量赋新值将导致错误）(L5-32.htm)。

```

01:<HTML>
02:<HEAD><TITLE>VBScript 错误处理</TITLE>

```

```

03:<SCRIPT Language="VBScript">
04:<!--
05:  Sub ChangeF(C)
06:      VbCrLf="MyNumber"          '改变预定义常量的值将导致错误
07:      ON Error Resume Next        '通知计算机在发生错误时先不要停止执行
08:      VbCrLf="MyNumber"          '改变预定义常量的值将导致错误
09:      DegreeF = C * 1.8+32
10:      If Err.Number = 0 Then        '程序运行正常时 Err.Number 的值等于零
11:          MsgBox "输入的摄氏温度: " & C & "°C, 转换为华氏是: " & DegreeF & "°F。"
12:      Else
13:          MsgBox "程序有错!!! "
14:      End If
15:  End Sub
16:  DegreeC = InputBox("请输入摄氏温度: ", "摄氏→华氏温度转换")
17:  ChangeF DegreeC
18:-->
19:</SCRIPT>
20:</HEAD>
21:<BODY></BODY>
22:</HTML>

```

说明: 预定义常量 VbCrLf 是不允许重新赋值的, 否则将会出错。第 7 行告诉计算机即使出现错误也不要停止运行。第 6 行和第 8 行是相同的语句, 第 6 行在此预先被注释掉, 读者可交替注释第 6 行、第 8 行观察执行结果。仅当程序出错时 Err.Number 的值才不等于零。

例 5.33 利用 CSS 滤镜 wave 制作形同波浪的文字 (L5-33.htm)。

```

01:<HTML>
02:<HEAD><TITLE>形同波浪的文字</TITLE></HEAD>
03:<BODY bgColor="#6495ED" onload="if (document.all)test()">
04:  <SCRIPT language="JavaScript">
05:    <!--
06:      var step=3;
07:      var xstep=0;
08:      var msg="山是眉峰聚 水是眼波横";
09:      document.write("<CENTER><DIV id='water' style='position:relative;
10:        width:380px;height:100px;font-family:隶书;font-size:60px;
11:        color:#FFA500">"+msg+"</DIV></CENTER>");          //显示文字
12:      function test(){
13:        document.all.water.style.filter='wave(freq=4, strength=7,
14:          phase='+xstep+', lightstrength=55, add=0, enabled=1)'; //设置 CSS 滤镜
15:        xstep+=step;          //x 方向偏移量
16:        setTimeout('test()',100); }
17:    //-->
18:  </SCRIPT>
19:</BODY>
20:</HTML>

```

说明: 本例适用于 IE 浏览器。第 9~11 行是一条语句, 编写代码时应在同一行内输入,

第 13~14 行同之。

例 5.34 在标题栏中交替闪烁的文字 (L5-34.htm)。

```
<HTML>
<HEAD>
<TITLE>标题栏文字</TITLE>
<SCRIPT language="JavaScript">
    var words1="★★★★★★★★★1";
    var words2="☆☆☆☆☆☆☆☆☆2";
    var speed=350;                                //设置字符交替闪烁速度
    var control=1;
    function change()
    {if (control==1)
        {document.title=words1;control=0;}
      else {document.title=words2;control=1;}
      setTimeout("change();",speed);
    }
</SCRIPT>
</HEAD>
<BODY>
    <P align="center">
        <B><FONT size="6" color="#6A6D32">标题栏文字</FONT>
        <FONT size="4" color="#6A5ACD">——文字在标题栏中交替闪烁</FONT>
    <P align="center">
        <A href="javascript:change()">看着标题栏单击我</A></P>
    </B></P>
</BODY>
</HTML>
```

说明：本例适用于 IE 和 NS 浏览器。还可以非常容易地将此功能应用于状态栏，处理方法是 将 document.title 换成 window.status。

例 5.35 为保护页面安全，进入网页时，要求输入密码登录 (L5-35.htm)。

```
<HTML>
<HEAD>
    <TITLE>密码保护页面</TITLE>
</HEAD>
<BODY>
    <SCRIPT language="JavaScript">
        function password() {
            var testV=1;
            var pass1=prompt('请输入密码(密码是 Welcome):'); //弹出输入密码对话框
            while (testV < 3) {
                if (!pass1)
                    history.go(-1);
                if (pass1 == "Welcome") { //检查输入的字符
                    alert('密码正确!欢迎进入! ');
                    window.open("L5-31.htm"); //进入页面
                }
            }
        }
    </SCRIPT>
```

```

        break;                                //跳出循环
    }
    testV+=1;
    var pass1=prompt('密码错误!请重新输入:');
    }
    if (pass1!="password" & testV ==3)
        history.go(-1);
    return " ";                                //重新输入
}
document.write(password());
</SCRIPT>
</BODY>
</HTML>

```

说明：本例适用于 IE 和 NS 浏览器，但执行效果有差别。



本章介绍了 HTML 的常用标识，还可以参阅其他参考书学习或上 W3C 网站查阅。使用 HTML 时还需注意以下几点：

- (1) 一般来说，HTML 标识是成对出现的。
- (2) 超文本标识语言的扩展名可以是 .htm 或 .html。
- (3) 标识中大小写字母可以通用，推荐统一字母书写。
- (4) 不应交叉使用标识（但交叉使用浏览器一般也不会显示出错）。
- (5) 一般不嵌套使用标识。
- (6) 可以在 FrontPage 2003 下单击“代码”视图，打开 HTML 源文件学习 HTML。

FrontPage 2003 具有“所见即所得”的强大开发能力，提供了网站设计、开发、制作与管理等多项功能。

应用层叠样式表可以使网页页面更加形式多样、丰富多彩，使用时注意选择器、样式规则的代码格式组成。

XML 的核心是以标准化的方式建立数据表示的结构，而将具体标识的定义留给用户。XML 的可扩展性使其可以满足各种不同领域的描述需求，并可对计算机之间交换的任何数据进行编码。但 XML 本身并不解释它所标识的标识含义，也不描述数据，而是交由第三方（如 XSLT、CSS、DTD 等）解释，并最终送至浏览器描述。

制作网页时应考虑下载速度，通常的概念是 60/30，即页面的内容小于 60KB，下载速度少于 30 秒。

Visual Studio 2005 是一个全面的集成开发工具，用于快速构建面向 Microsoft Windows 和 Web 并连接 Microsoft .NET 的应用程序，提供了一个强大而完善的端到端工具，用以设计、开发、调试和部署用于 Microsoft Windows 和 Web 的安全的应用程序，这些应用程序强健而且易于使用。应用 Visual Studio .NET 可帮助构建最复杂的企业级应用程序和在最小的设备上部署应用程序，极大地提高开发人员的效率。

脚本语言自身并不是独立的编程语言，只是作为 Web 页面的一部分依靠 HTML 而存在。

脚本语言源代码是纯文本，只能把它放在 HTML 文档中才能执行。

VBScript 是 Visual Basic 的一个子集，具有 Visual Basic 的类似功能；而 JavaScript 并不是 Java 的子集，不具有语言的通融性。

VBScript 和 JavaScript 都不是真正面向对象的语言，两者都没有表现出一种真正面向对象语言的全部性，例如，两种语言都不支持继承和多态。由于 JavaScript 是基于对象的脚本语言，所以它利用对象的能力比 VBScript 要强一些。

IE 浏览器与 Netscape Navigator 浏览器对 JavaScript 的支持有所不同，调试程序应分别使用不同的浏览器。若要保证程序在两种浏览器中均能正常显示，可使用程序识别、判断浏览器的类型。

使用 JavaScript 时应注意字母的大小写，特别是对象、方法、属性等的名称，这些名称有大小写之分，书写不同将引起错误，这一点与 VBScript 大不一样。



1. 通常在网页中使用的图像格式是_____和_____。
2. FrontPage 2003 主窗口提供的主要视图有_____、_____、_____和_____。
3. HTML 是一种_____的标识语言；XML 是一种_____的标识语言。
4. XSL 在转换 XML 文档时分为两个过程，一是_____结构；二是_____输出。
5. XML 的四大主要特点是_____、_____、_____和_____。
6. 为什么说“超文本标识语言不是程序语言”？
7. HTML 是什么语言？适用于制作哪种网页？是什么类型的文件？
8. 用哪些工具可以编辑 HTML？
9. 用记事本、Word 2003、FrontPage 2003 分别制作同一个网页，请比较源程序。
10. 标识的含义是什么？
11. 写出 HTML 文档的组织结构。
12. 什么是图片的文本替换？请写出一条可运行的语句。
13. 写出在超链接的文本中附带摘要信息的语句。
14. 什么是表格？什么是表单？各自有何差异？各自的主要功能是什么？
15. 框架结构适用于什么样的网站？
16. 简述 FrontPage 2003 的主要功能和特征。
17. 使用 FrontPage 2003 写出制作透明图像的过程。
18. 简述在 FrontPage 2003 下怎样制作框架。
19. 写出层叠样式表（CSS）的主要功能，怎样使用层叠样式表？
20. 利用 style 写出一条内含 CSS 控制文字的正确 HTML 语句。
21. Visual Studio 2005 的功能可以细分为哪几类？
22. Visual Studio 2005 的代码编辑器智能体现在哪几个方面？
23. Visual Studio 2005 的智能感知主要包括哪几项内容？
24. Visual Studio 2005 的项目是怎么组织的？

25. 请列举 Visual Studio 2005 中几个最常用的窗口, 并说明它们各自的作用。

26. 请用 XML/XSL 语言编制程序, 显示并输出下述数据。

```
<汽车类>
  <汽车>
    <产地>国内</产地>
    <品牌>东风日产</品牌>
    <系列>骐达</系列>
    <型号>自动尊贵型</型号>
  </汽车>
  <汽车>
    <产地>国外</产地>
    <品牌>宝马</品牌>
    <系列>X6 车型</系列>
    <型号>xDrive 50i</型号>
  </汽车>
</汽车类>
```

27. 利用数据岛捆绑习题 26 中的数据, 并用表格显示输出。

28. 自设一幅具有两个图像热点的网页, 并对其中一个热点设置超链接, 所设地址为 <http://www.csasp.net/>。

29. 上网浏览不同风格的网站, 例如:

<http://www.microsoft.com>

<http://www.ibm.com>

<http://www.cctv.com>

<http://www.chinaasp.com>

<http://www.xinhua.org>

<http://www.sina.com.cn>

<http://www.computerworld.com.cn>

<http://www.csasp.net/>