



深源圖書創作室
SHENYUAN TUSHU CHUANGZUOSHI

Internet 系列



CD-ROM



深源圖書創作室 編

HTML

JavaScript

CSS

Flash

网页特效教程

山東電子音像出版社

前 言

近几年来,计算机技术飞速发展,计算机应用日益普及,特别是 Internet 上提供的各种服务,深刻地影响着人们日常的工作、学习、交往、娱乐等各种活动。从更深的层次上讲,以计算机技术为核心的信息技术极大地改变了人们的思维方式和生活方式。我们的生活环境变得越来越数字化。时下,制作个人主页是十分流行的事,在因特网上拥有自己的网站已成为现实。对于每一个拥有自己网站的网虫而言,不断地精益求精、创造迷人的效果可能是最大的梦想之一。

本书内容分为 4 个部分,分别讲述制作网页及网页特效所需要的语言和工具 HTML、CSS、JavaScript、Flash 的有关知识,制作网页,最基本的语言是 HTML,但是 HTML 是一种简洁的语言,它没有过多地控制网页的格式和外观,只是定义了网页的结构和各元素的功能。要让我们的网页更加富于变化、更加精彩,就要求助于动态的 HTML——DHTML 以及其他一些工具。其中, DHTML 是一些语言和技术规范的总称,包括脚本语言 JavaScript、VBScript 和 Cascading Style Sheets(CSS 样式表)等;其他工具如 Flash 等。

CSS 样式表诞生于 1996 年底,1998 年 5 月 12 日, CSS2.0 成为了 W3C 的新标准。CSS 样式表是一组样式,简单地讲,它增加了更多的样式定义方式来辅助 HTML。CSS 样式表最精彩的一部分是它的滤镜,可以用来制作很多精彩的效果。很多网页特效都是 JavaScript 结合 CSS 样式表实现的。

JavaScript 语言是一种运行于客户端的脚本语言,浏览器可以直接解释执行 JavaScript 的程序。和 VBScript 相比, JavaScript 语言更成熟,对不同浏览器的兼容性也更好,所以制作网页特效的脚本语言一般都用 JavaScript。JavaScript 语言是一种面向对象的语言,它的语法并不复杂,常用函数,对象及对象的属性、方法都不多,读者可以参考实例很快地掌握这种脚本语言的用法。

Flash 是一个面向网页制作的动画软件。用 Flash 可以做出各种精美的动画,而且 Flash 的 ActionScript 可以通过编程实现交互动画, ActionScript 的语法类似 JavaScript,并不难学。

通过学习这些知识,你能够制作出各种网页特效效果,还能够为深入到高层次的网络编程技术打下基础。

本书内容新颖,可供初、中级用户借鉴参考,也适合广大网迷阅读。

本书配套光盘包含了书中的实例效果及源文件。

为读者服务、对读者负责是深源图书创作室的工作宗旨;严谨、创新、求实是深源图书创作室的工作态度;高品位、高质量是深源图书创作室孜孜以求的目标。尽管倾心相注、精心而为,但书中的错误和不足在所难免,恳请读者批评指正。

深源圖書創作室

联系电话: (0532) 3823831

目 录

第 1 章 HTML	1
1.1 HTML 4.0 概述	2
1.1.1 HTML4.0 的新特点	2
1.1.2 初识 HTML	3
1.1.3 HTML 文档中的注释	6
1.2 文档头	7
1.2.1 可在文档头部分使用的标识符	7
1.2.2 meta 标识符	8
1.3 网页的页面属性	9
1.3.1 定义背景色、背景图片和文字颜色	9
1.3.2 设置网页页边空白	10
1.4 链接	11
1.4.1 基本语法	11
1.4.2 在新窗口中打开链接	13
1.4.3 用链接实现发信	14
1.5 字体样式	15
1.5.1 标题字体	15
1.5.2 设置字号	15
1.5.3 设置物理字体和逻辑字体	16
1.5.4 设置字形	18
1.5.5 设置文字颜色	18
1.5.6 字体设置标识符的组合使用	18
1.6 网页布局	19
1.6.1 设置标尺线	20
1.6.2 行的控制	20
1.6.3 文字的对齐的方式	21
1.6.4 列表	22
1.6.5 预排格式化文本	27
1.6.6 标识符 和 	28
1.6.7 标识符<address>和</address>	29
1.6.8 标识符<blockquote>和</blockquote>	30
1.6.9 空白区域	31
1.7 图像	31
1.7.1 基本语法	32
1.7.2 图像在网页中的布局	33
1.7.3 设置图像的大小和边框宽度	34
1.8 表格	34

1.8.1 定义表格的基本语法	34
1.8.2 设置表格外观	35
1.8.3 控制表格内文字的布局	43
1.9 多窗口结构	44
1.9.1 基本语法	44
1.9.2 设置各窗口的尺寸	45
1.9.3 框架的嵌套	47
1.9.4 框架结构各窗口之间的关系	49
1.9.5 设置框架的外观	49
1.9.6 浮动窗口	51
1.10 表单	52
1.10.1 form 的基本结构	53
1.10.2 下拉菜单	54
1.10.3 复选框	58
1.10.4 单选按钮	59
1.10.5 提交和重置按钮	61
1.10.6 文字和密码输入	62
1.10.7 文本区域	63
1.11 嵌入多媒体文件	65
1.11.1 嵌入声音文件	66
1.11.2 嵌入电影文件	66
第 2 章 样式表 CSS	68
2.1 CSS 和 HTML 的关系	69
2.2 CSS 的基本结构	69
2.2.1 样式规则	70
2.2.2 分类和情景选择	72
2.3 将样式表加入网页中	74
2.3.1 把样式表嵌入到文档头中	75
2.3.2 链接到样式表	76
2.3.3 输入样式表	78
2.3.4 在行内加入样式	80
2.4 class、ID 属性和 span、div 元素	81
2.4.1 class 属性	81
2.4.2 ID 属性	82
2.4.3 Span 元素	82
2.4.4 div 元素	83
2.5 设置字体样式	84
2.5.1 按照字形系列名称调用字体	85
2.5.2 对字号的控制	85
2.5.3 字体风格	90

2.5.4 字粗细度	90
2.5.5 文字的特殊效果	91
2.5.6 文字修饰	91
2.6 控制图文布局	92
2.6.1 设置字间距及字母间距	92
2.6.2 设置行距	93
2.6.3 让文字互相重叠	95
2.6.4 文字对齐及缩行	96
2.6.5 间隙、边框及空白区域	98
2.7 设置颜色和背景	100
2.7.1 背景颜色	100
2.7.2 背景图像	101
2.7.3 控制背景图像	102
2.8 定位和叠放	103
2.8.1 绝对定位	104
2.8.2 相对定位	104
2.8.3 宽度、高度和可视性	104
2.8.4 叠放文字和图像	105
第3章 JavaScript	107
3.1 什么是 JavaScript	108
3.1.1 JavaScript 的基本特点	108
3.1.2 JavaScript 应用	108
3.2 在 Web 页中使用 JavaScript	109
3.2.1 嵌入 JavaScript	109
3.2.2 链接 JavaScript	111
3.3 JavaScript 基本数据结构	112
3.3.1 基本数据类型	112
3.3.2 表达式和运算符	114
3.4 语句、函数与事件	115
3.4.1 控制语句	115
3.4.2 函数	116
3.4.3 事件驱动及事件处理	117
3.4.4 示例	119
3.5 常用内部对象	121
3.5.1 对象的基础知识	122
3.5.2 String: 字符串对象	124
3.5.3 Math: 数学对象	126
3.5.4 Date: 日期及时间对象	127
3.5.5 系统函数	127
3.5.6 示例	128

3.6 创建新对象	133
3.6.1 对象的定义	133
3.6.2 创建对象实例	133
3.6.3 对象中方法的使用	134
3.6.4 JavaScript 中的数组	134
3.6.5 示例	136
3.7 浏览器的内部对象	142
3.7.1 浏览器中的对象树	142
3.7.2 navigator 对象	143
3.7.3 window 对象	143
3.7.4 history 对象	144
3.7.5 document 对象	144
3.7.6 示例	147
3.8 窗口输入输出	148
3.8.1 window 对象与输入	150
3.8.2 document 对象与输出	151
3.8.3 示例	152
第4章 Flash 5.0	156
4.1 Flash 基础	157
4.1.1 Flash 5.0 概述	157
4.1.2 我的第一个 Flash 动画——移动的球	158
4.2 Flash 基础知识	163
4.2.1 菜单栏与工具栏	163
4.2.2 面板	170
4.2.3 静态图形制作实例	178
4.3 Flash 进阶知识	187
4.3.1 场景与时间轴	187
4.3.2 Flash 的图层	189
4.3.3 Flash 中的帧	191
4.3.4 Flash 的组件	194
4.3.5 Flash 的图层应用之一	200
4.3.6 Flash 的图层应用之二	205
4.4 Flash 的高级功能	207
4.4.1 Flash 的动作(Action)	207
4.4.2 Flash 动作(Action)应用实例	212
4.4.3 Flash 必备知识	218
附录1 HTML 4.0 速查手册	222
附录2 CSS 速查手册	237
附录3 Flash 5 快捷键	243

第 1 章

HTML

本章概述

HTML 在众多的 Web 技术中恐怕是最显得微不足道的了，但 HTML 的确确是精彩的 Web 世界必不可少的基石。使用 HTML 可以创建静态的、内容丰富的、有趣的 Web 页，这在 Internet 世界的初期可说是一个创举。

本章介绍了 HTML 的相关知识，包括 HTML 的特点、文档头、链接、字体样式、网页布局、图像、表格、多窗口结构、表单以及嵌入多媒体文件等知识。

1.1 HTML 4.0 概述

HTML 由版本 1.0 已经发展到今天的 4.0。HTML 4.0 通过以下机制扩展了 HTML：样式表、脚本、框架、嵌入对象、对从右到左文本方向和混合文本方向的支持、功能更强的表格、增加的表单，并且为残疾人提供了访问的可能。

下面我们了解一下 HTML 4.0 的新特点。

1.1.1 HTML4.0 的新特点

1.国际化

HTML 4.0 可以使用各种语言编写网页文档，并易于在全世界内发布。HTML 4.0 设计过程中的一重要步骤是采用了 ISO/IEC: 10646 标准作为 HTML 的字符集，该标准在处理国际化字符、文本方向、发音和其他语言问题方面最具包容性。

HTML4.0 在文档内提供了对不同种类语言的更强大的支持。这就为搜索引擎提供了更有效的文档索引功能、更高质量的排版格式、更好的文本到发音的转换、更好的连字符连接。

2.可访问性

目前网络正在飞速发展，而网络上各个用户的能力和技术各不相同，所以在底层技术上满足他们的特殊需要是很重要的。HTML4.0 涉及到的便于访问的方面包括：

- 文档的结构和表现更好地区分开来，这就鼓励使用样式表而不用 HTML 表现元素和属性。
- 更好的表单，包括附加的存取键、按语意分类的表单控件、Select 选项和活动标识符。
- 标记作为内嵌对象的文本描述的能力(使用 Object 标识符)。
- 一个新的客户端图像映射机制(map)，允许把图像和文本链接集成在一起。
- 替代文字与图像(包含在 img 标识符内)和图像映射(包含在 area 标识符内)一起出现。
- 支持所有元素的 title 和 lang 属性。
- 支持 abbr 和 acronym 元素。
- 与样式表一起使用的更多的目标媒体(tty, braille 等)。
- 更好的表格，包括标题、列组和实现不可见渲染的机制。
- 表格、图像、框架等长名描述。

在设计中考虑到可访问性因素还有其他的好处：把结构和表现分开的 HTML 文档更容易采用新技术。

3.表格

通过表格，设计者可以更好地控制结构和布局。定义列宽度的能力可以逐渐地显示表格，而不必等到全部表格被绘制完。

4.混合文档

HTML 现在提供在 HTML 文档中嵌入常见媒体对象和应用程序的标准机制。Object 标识符(与 img 和 Applet 一起)提供了在文档内包含图像、视频、声音、数学、特殊应用程序和其他对象的功能。

5.样式表

样式表简化了 HTML 标记,它使设计者和用户都可以控制文档的表现形式、字体信息、对齐方式、颜色等。

样式信息可以为单独的元素或一组元素定制。样式信息可以定制在 HTML 文档内,也可以以外部样式表的形式出现。

把样式表与文档联系起来的机制独立于样式表语言。

在样式表出现之前,制作者很少控制网页绘制过程。虽然 HTML 3.2 中包括了一些属性和标记,它们可以提供对齐方式、字体大小和文本颜色的控制,制作者也可以使用表格和图像进行网页布局,但用户需要较长时间才能升级他们的浏览器,这意味着对这些特征的使用还要持续一段时间。然而,既然样式表提供更强大的表现机制,WWW 协会将最终逐渐淘汰很多 HTML 的标识符和属性。

6.脚本

通过脚本,可以制作动态网页,或使用 HTML 建立网络应用程序。

把脚本包含在 HTML 文档中的机制独立于脚本语言。

1.1.2 初识 HTML

下面我们开始编写第一个 HTML 文档。使用一个你习惯的文本编辑器,比如 Windows 自带的写字板,编写如下代码:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>例 1</title>
</head>
<body>
<p>例 1: 初识 HTML</p>
</body>
</html>
```

将以上代码保存到硬盘上,注意文件的后缀名为.html 或者.htm,本例保存为 1.htm。双

击打开该文件，网页显示效果如图 1.1 所示。



图 1.1 初识 HTML

从上述程序中我们可以看出 HTML 文档的基本结构如下：



一个 HTML 文档由一系列标识符组成，这些标识符包含在尖括号< >中，而且大多成对出现。成对出现的标识符中后一个以</标识符>结束。一对标识符仅对包含在其中的文档部分发生作用。

一个 HTML 文档，总是由标识符<html>开始，然后是由<head>和</head>标识的文档头部分，文档头中可以包含文档的标题(<title></title>)、<meta>属性以及脚本代码等，然后是文档的正文部分，由<body>和</body>标识，最后由</html>结束。当然这些部分并非都是不可缺少的。文档的标题甚至整个文档头都是可以省略的，表示结束的标识符</body>和</html>也可以省略。

下面我们再看一个例子，使用文本编辑器编写一个 HTML 文档，代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>文档标题</title>
</head>
<body>
```

```
文档正文  
</body>  
</html>
```

下面做如下尝试:

- 设置文档标题为你想要的标题, 比如“我的主页”。方法是修改<title>和</title>之间的文字即可。
- 在文档体部分写一段文字。方法是将文字插入在<body>和</body>之间即可。
- 删去文档头部分, 即删去<head>和</head>以及它们之间的全部文字。然后保存起来再打开文件观察显示结果。
- 删去</body>和</html>标识符, 保存文件, 然后打开文件观察显示结果。

文档标题显示在浏览器菜单栏的上方最左端。执行以上 4 步操作之后的显示结果如图 1.2 所示。



图 1.2 HTML 文档

需要说明的是, HTML 文档的标识符并不区分大小写, 即大写、小写或者混合使用都可以。比如: <html>和</html>, <HTML>和</html>以及<HTMl>和</htMI>都是有效的。但是请注意以下 3 点:

- 对于很多浏览器, 如果是特殊字符, 则只能使用小写。例如版权字符是“©”, 如果写成“©”, 那么在网页中将显示为“©”, 而非版权符号。关于字符实体, 请参考本书附录 1。
- 如果在 HTML 文档中嵌套了脚本代码(使用<Script></Scritp>标识符), 要注意, 这些脚本代码(比如 JavaScript)通常对变量名是区分大小写的, 这时变量 Number 和变量 number 就是不同的两个变量。
- 在 HTML 文档中使用链接、图片等调用其他文件时, 最好注意大小写, 这样具有更好的可移植性。因为虽然 Windows 系统对文件名称不区分大小写, 但是 UNIX 系统对大小写是敏感的。如果不注意区分大小写, 当网页移植到使用 UNIX 操作系统的服务器上时就可能

会出现错误。

另外,在对标识符的属性赋值时,有时使用了引号,有时则没有。是否一定要使用引号呢?

一般来说,引号并不是必需的。而且,使用单引号、双引号皆可,但是必须前后配对,即如果左引号是单引号,则右引号也应该是单引号。

给标识符的属性赋值时,使用引号是一个好习惯,而且使用引号赋值是 HTML 规范的一个发展趋势。在使用网页制作工具自动生成的 HTML 代码中,都是使用引号的。另外,在使用脚本(如 JavaScript 语言)、样式表等的代码中,引号一般是必需的。

1.1.3 HTML 文档中的注释

在 HTML 文档中进行注释的方法有两种:

- 使用标识符`<!--和-->`可在 HTML 网页内插入注释性的说明文字,包含在这两个标识符之间的代码不会被浏览器处理和显示。

- 在脚本代码(使用标识符`<Script></Script>`进行标识)中,使用`“//”`进行注释。`“//”`之后的一行文字将不被浏览器处理和显示。

使用注释的目的常常有以下两种:

- 给出提示信息或者描述信息。
- 用于隐藏脚本代码。我们常常将脚本代码包含在`<!--和-->`标识符之间。这是因为一些低版本的浏览器有时不支持这些脚本语言,我们将其写在注释标识符之间,就不会被显示成为乱七八糟的网页。而支持这些脚本语言的浏览器仍然会执行这些脚本代码。

下面我们看一个 HTML 文件,原文件较长,删去了大部分内容,我们只需要看其中的中文注释部分。其中注释用来说明版权信息、隐藏脚本代码和样式表代码以及进行其他说明。代码如下:

```
<html>
<title>mouse track</title>
<meta name="description"content="有趣的鼠标跟踪脚本">
<script language="JavaScript">
<!--//hide
function
YY_Layerfx(yyleft,yytop,yyfnx,yyfny,yydiv,yybilder,yyloop,yyto,yycnt,
yystep)
{
if ((document.layers)||(document.all)){
.....
```

```
    }  
-->  
</script>  
<style type="text/css">  
  <!--  
    .pt9{ font-family:"宋体";font-size:9pt}  
    .....  
  -->  
</style>  
</head>  
<body bgcolor="#000000"text="#FFFFFF">  
  <!--1:1 广告交换的代码 400*40-->  
  <div align=center">  
    .....  
  <!--结束 1: 1 广告交换的代码--></div>  
  .....  
</body>  
</html>
```

1.2 文档头

习惯上,标识符<head>和</head>之间的部分称为文档头,文档头是用来描述网页的内容的,对网页的外观没有实际的影响。

1.2.1 可在文档头部分使用的标识符

- <title></title>指示文档的标题。
- <base>定义当前文件的基值。它有如下属性:
 - href 定义当前文档的基 URL。
 - target 为当前文档中所有链接定义一个默认目标窗口。
- <isindex>指示该文档是允许查找的网页文件。它有如下属性:
 - prompt="..." 查找字段的提示信息。
- <link>指示该文档与其他文档之间的关系。通常只有 HTML 生成工具使用。它有如下属性:
 - href 被引用的 HTML 文档的 URL。

- rel 正向关系。
- rev 逆向关系, 通常是文档作者的电子邮件地址。
- title 链接的标题。
- <nextID>指示该文档之后的下一个文档(可能是由顺序管理 HTML 文档的工具定义的)。
- <meta>用来在 HTML 文档中模拟 HTTP 协议的响应头报文。

1.2.2 meta 标识符

meta 标识符放在每个网页的<head>...</head>中, meta 标识符的用处很多, 比较常用的几种如下面的例子所示。

- <meta name="Generator"content="..."> 说明生成工具(如 Microsoft FrontPage 4.0)。
- <meta name="Key Words"content="..."> 说明关键词。
- <meta name="Description"content="..."> 对主页进行一定的描述。
- <meta http-equiv="Content-Type"content="text/html;charset=gb_2312-80"> 和 <meta http-equiv="Content-Language"content="zh_CN"> 说明常用文字及语言。

由上可见, meta 的属性有两种: name 和 http-equiv。

name 属性主要用于描述网页, 对应于 content, 以便于搜索引擎查找、分类(目前几乎所有的搜索引擎都通过查找 meta 值来给网页分类)。这其中最重要的是 Description(站点在搜索引擎上的描述)和 Key Words(搜索引擎用以分类的关键词), 应该给你的“每一页”都插入这两个 meta 值。当然, 也可以使用如下代码来决定是否让搜索引擎进行检索:

```
<meta name="Robots"content="all | none | index | noindex | follow | nofollow">
```

其中的属性说明如下:

- 设定为“all” 文件将被检索, 且页上链接可被查询
- 设定为“none” 则表示文件不被检索, 而且不查询页上的链接
- 设定为“index” 文件将被检索
- 设定为“follow” 则可查询页上的链接
- 设定为“noindex” 时, 文件不检索, 但可被查询链接
- 设定为“nofollow” 则表示文件不被检索, 但可查询页上的链接

http-equiv 属性的作用相当于 http 文件头, 可以直接影响网页的传输。比较直接的例子如下:

(1)自动刷新, 并指向新网页

如下代码的效果是 10 秒钟后刷新网页, 跳转到 <http://yourlink:>

```
<meta http-equiv="Refresh"content="10;url=http://yourlink">
```

(2)在网页转换时加入效果

```
<meta http-equiv="Page-Enter"
content="revealTrans(duration=10,transition=50)">
<meta http-equiv="Page-Exit"content="revealTrans(duration=20,transition=6)">
```

把以上代码加在一个网页中，进入和离开该网页时，能有一些特殊效果，这个功能即 FrontPage 即 format/Page Transition，不过所加网页不能是一个 frame 页。

(3) 强制网页不被存入 Cache 中

```
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="expires" content="wed,26Feb 1997 08:21:57 GMT">
```

离开网页后，就无法从 Cache 中再调出。

(4) 定义指向窗口

```
<meta http-equiv="window-target" content="_top">
```

可以防止网页被别人作为一个 frame 页调用。

下面给出一个范例，使用 meta 标识符来实现网页自动定时跳转到其他网页的效果，编写如下代码：

```
<html>
<head>
<title>示例</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body>
</body>
</html>
```

1.3 网页的页面属性

Web 网页具有如下属性：背景色、背景图片、网页文字颜色，以及网页页边空白。这些属性可以使用相应的标识符来设置，以控制网页的显示。

1.3.1 定义背景色、背景图片和文字颜色

定义背景和文字颜色的基本语法如下：

```
<body bgcolor=# text=# link=# alink=# vlink=#>
```

属性值：

- bgcolor 背景色
- text 非可链接文字的颜色
- link 可链接文字的颜色
- alink 正被单击的可链接文字的颜色
- vlink 已经单击(访问)过的可链接文字的颜色

其中“#”代表颜色值,使用16进制的“红-绿-蓝”(red-green-blue, RGB)值来表示,也可以使用颜色的英文名称。

定义背景图片的语法如下:

```
<body background="image-URL">
```

其中,“image-URL”指图片的位置。

下面给出一个范例。首先编写如下代码:

```
<html>
<head>
<title>示例</title>
</head>
<body bgcolor=#FFFFFF text="purple"
link="#0000FF" vlink="#00FF00" alink="#FF0000">
正文<br>
<a href="1.htm">链接 1</a><br>
<a href="2.htm">链接 2</a><br>
</body>
</html>
```

首先将以上代码保存起来,然后打开该网页,观察其中文字的颜色和链接的颜色。对链接的颜色,注意观察未单击时、单击时和单击过的链接的颜色的区别。并可自行修改各属性的颜色值,进行观察。该例中,背景色为白色,一般文字的颜色为紫色,链接在未单击时为蓝色,单击后为绿色,单击时为红色。

下面我们将<body...>一行修改成如下代码:

```
<body background="images/bg.jpg"
text="purple"link="#0000ff"vlink="#00ff00"alink="#ff0000">
```

上述代码的作用是使用图片 bg.jpg 作为背景,该图片位于该文档所在位置的同级目录 images 中。你可以将 bg.jpg 改为你想要用作背景的图片的名称,并正确书写它的位置。

对于 HTML 文档中使用的其他文件(HTML 文件、TXT 文件、图片文件等),既可以使用绝对地址(URL),也可以使用相对地址。实际上,使用得最多的是相对地址。使用相对地址时,以该 HTML 文档的位置为基准,可用“../”表示返回到上一层目录。

1.3.2 设置网页页边空白

设置网页页边空白(Margin)的基本语法如下:

- 设置网页左边的空白: <body leftmargin=#>
- 设置网页上边的空白: <body topmargin=#>

其中“#”为空白量,单位为像素。

下面给出一个示例，编写如下代码：

```
<html>
<head>
<title>示例</title>
</head>
<body leftmargin=200 topmargin=50>
    对于 HTML 文档中使用的其他文件 (HTML 文件、TXT 文件、图片文件等)，既可以使用绝对地址
    (URL)，也可以使用相对地址。实际上，使用得最多的是相对地址。使用相对地址时，以该 HTML 文
    档的位置为基准，可用“.../”表示返回到上一层目录。<a href="1.htm">链接 1</a><br>
</body>
</html>
```

上述代码在网页左边设置了大小为 200 像素的页边空白，在网页顶部设置了大小为 50 像素的页边空白。网页显示效果如图 1.3 所示。



图 1.3 设置网页边空白的效果

1.4 链接

当单击一个链接时，可以到达新的网页或者网页的其他部分，这是 Web 网页最重要的特征之一。在这一节将介绍链接的设置方法。

1.4.1 基本语法

链接(Link)是 HTML 文档的重要组成部分。它的基本语法如下：