



北京科海培训中心

网页设计·网站建设培训系列教材

CSS & DHTML

网页制作特效与实例



杨正华 编著

92

清华大学出版社



北京科海培训中心

► 网页设计·网站建设培训系列教材

CSS & DHTML

网页制作特效与实例

杨正华 编著

清华大学出版社

(京)新登字 158 号

内 容 提 要

动态 HTML(DHML)是网络发展进程中最振奋人心也最具实用性的创新之一。

本书以具体实用的示例,通过简明精练的讲解,介绍了 HTML 规范;以明晰的实例,讲述了通过层叠样式表(CSS)如何实现页面视觉效果的一切变化,如字体、信息、颜色、位置、大小、分层与重叠等;通过脚本语言介绍如何简便达到网页动态交互效果。

全书对动态网页制作过程中可能遇到的大大小小、方方面面的问题都给出了深入浅出的解答,并提供 55 个 JavaScript 和 CSS 精彩范例,展示了令人震撼的动态网页技巧,引导读者把握网页制作的精髓所在。

本书适合于各个层次的网页制作者。

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得进入各书店。

书 名: CSS & DHTML 网页制作特效与实例
作 者: 杨正华
出版者: 清华大学出版社(北京清华大学校内,邮编 100084)
网 址: <http://www.tup.tsinghua.edu.cn>
印刷者: 北京朝阳科普印刷厂
发 行: 新华书店总店北京科技发行所
开 本: 787×1092 1/16 印张: 25.75 字数: 626 千字
版 次: 2000 年 11 月第 1 版 2000 年 11 月第 1 次印刷
印 数: 0001~5000
书 号: ISBN 7-302-04131-8/TP·2435
定 价: 32.00 元



前言

正如 Homepage 一词由“家”(Home)和“页面”(Page)合成一样,主页是网民的网上家园。那么究竟什么是主页呢?从本质上讲主页是由一系列 HTML 标识符把文字、图像、数据和多媒体等内容组织起来的超级文档。

HTML 历史简介

HTML 最初由 Tim Berners Lee 开发。当他在 CERN(欧洲粒子研究所)时,因为 NCSA(美国国家超级计算机应用中心)开发了 Mosaic 浏览器而使 HTML 流行起来。在 1990 年,由于 Web 的爆炸性增长而使 HTML 得到成功。这期间,HTML 以多种不同方式被扩充。

在 Internet Engineering Task Force(IETF,因特网工程部任务组)的支持下,在 1994 年末通用的规则被制定为 HTML 2.0(1995 年 11 月制定)。其后 HTML+ (1993)和 HTML 3.0(1995)在内容上更加丰富。尽管从来没有得到对标准讨论的一致同意,但是这些草稿导致很多新特征的采用。WWW 协会的 HTML 工作组在 1996 年的努力结果制定出 HTML3.2(1997 年 1 月)。

多数人认为 HTML 文档应该在不同浏览器和平台上都工作得很好,只有这样才能取得互操作性、降低内容提供者的费用,因为他们只需开发文档的一个版本。如果不做这种努力,就会使 Web 发展成不兼容格式的私有世界,最终会削弱 Web 对所有参加者的商业潜力。

HTML 已经发展到各种设备都可以利用网上信息的阶段,这些设备包括:使用不同分辨率的 PC、蜂窝电话、手持式设备、快速输入输出设备、使用高带宽和低带宽的计算机等。

动态 HTML

动态 HTML(DHTML)是近年来网络发展进程中最振奋人心也最具实用性的创新之一。

什么是动态 HTML

动态 HTML 作为浏览器自带的功能,实际上只是一种概念,它并不属于一种专门技术(如 JavaScript, ActiveX),它不是标识符,不是 Plug-in(软插件),也不是浏览器。它是一种通过各种技术的综合发展而得以实现的概念(当然,不同的浏览器实现的程度也不

同), 这些技术包括 JavaScript, VBScript, Document Object Model (文档目标模块), Layers 和 Cascading Style Sheets (CSS, 层叠样式表) 等。

简而言之, 动态 HTML 是一种即使在网页下载到浏览器以后仍然能够随时变换的 HTML。比如, 当鼠标移至文章段落中时, 段落能够变成蓝色, 或者网页标题能够滑过电脑屏幕。HTML 中能够做的一切在网页下载后还能够再重复。

HTML 规范发展到 4.0 版本时, 已经可以称做是一种动态 HTML 规范了。

HTML4.0 通过以下机制扩展了 HTML: 样式表、脚本、框架、嵌入对象、改进的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、增强的表单, 并且为残疾人提供了访问的可能。

事实上, HTML4.0 主要包括以下三部分内容:

- 扩展的 HTML 基本标识符集。
- 对层叠样式表的支持。
- 嵌入脚本语言。

下面我们重点介绍后两部分。

通过层叠样式表 (Cascading Style Sheets, CSS), 能够实现页面在视觉效果上的几乎一切变化。通过改变页面元素的 CSS 属性 (如颜色、位置、大小), 我们几乎可以达到你的机器的带宽和处理器运行速度允许范围内的一切效果。

层叠样式表简化了 HTML 标记, 并且减轻了 HTML 对表现形式的责任。它们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、定位、分层与重叠等。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。这实在是一件非常美妙的事情。

而通过脚本 (主要是指客户端脚本语言, 比如 JavaScript 语言或者 VBScript 语言), 我们可以很简便地生成实际促成变化的程序, 不需要服务器端的 CGI (通用网关接口) 程序, 就可以达到动态交互的效果。比如, 我们让一幅图在鼠标滑过时产生变化, 这就是一个典型的用脚本语言实现的动态 HTML 的例子。

本书正是从以上 3 部分, 对动态网页制作过程中可能遇到的大大小小、方方面面的问题, 所以给出了深入浅出、精彩明晰的解答, 同时也给出了简明而形象的例子。

本书内容

在 HTML4.0 标识符部分, 本书包括了以下内容:

- HTML 基本概念
- 页面属性设置
- 链接设置
- 字体样式设置
- 页面布局控制
- 图像设置
- 表格设置
- 框架设置

- Form 设置
- 多媒体设置
- 移动的文字: Marquee
- <Script>、<Applet>和<Style>标识符

在JavaScript部分, 本书包括了以下内容:

- JavaScript 简介
- 基本数据结构
- 语句、函数与事件
- 常用内部对象: 串 (String) 对象、Math 对象、日期及时间 (Date) 对象
- 系统函数
- 创建新对象以及 JavaScript 中的数组
- 浏览器内部对象树: navigator 对象、window 对象、history 对象、Document 对象等
- 窗口及输入输出
- 使用 Form 及复杂交互

在样式表部分, 本书包括了以下内容:

- 样式表基本概念
- 样式表的分类和情景选择
- 样式表加入网页的 4 种方法
- Class, ID 属性和 Span, Div 元素
- 样式表中对字体的控制
- 样式表中对图文和布局的控制
- 样式表中对颜色和背景的设置
- 样式表中的定位和叠放
- 样式表与浏览器的兼容性

本书最后两章, 是“JavaScript 精彩范例”和“样式表精彩范例”, 共有 55 个例子, 实现了令人震撼的动态网页特效, 这是动态网页制作范例的一个宝藏。这些例子, 精彩、实用、全面, 你可以直接或者稍加改动后就能拿来嵌入到你的网页代码中, 以达到特殊的动态效果。其实, 即使你不懂 JavaScript 脚本语言和样式表, 也没有关系, 你只需要按书中的说明, “拿来”就是了。

在本书的编写过程中, 我们注意尽量减少冗长乏味的说明代之以具体实用的示例。通过简明精练的讲解, 精彩明晰的实例, 引导读者把握网页制作的精髓所在, 适合于各个层次的网页制作者。对初学者而言, 可迅速地设计出颇具水准的个人网页; 对有一定制作经验的读者, 也可以由此全面掌握动态网页制作 (尤其是脚本语言和样式表) 的制作方法。

“制作网页其实是一件很容易的事情”, 不论你是一个网页制作新手, 还是一个资深高手, 拥有本书, 你就会拥有这种感觉。

准备工作

为了使用本书，你需要一个支持动态 HTML 的浏览器。本书的所有例子都在 IE 4.0 下调试成功，IE 4.0、IE 5.0 及其更高版本支持本书的全部例子，它们都是我们推荐使用的浏览器。同时，你也可以安装 Netscape Navigator 4.0 或更高版本，以测试不同浏览器下的显示效果。

此外，你还需要一个顺手的编辑工具。事实上，如果仅仅是学习和测试本书中的范例效果，你需要一个文本编辑器就足够了。它可以是 Windows 9x 的记事本或者写字板。

然后，你就可以开始阅读本书。

然后，制作动态网页，一切都会容易起来。

第1章 HTML 4.0

主页是 Web 上信息发布的主要形式，它由一些 HTML 文本文件组成。HTML 即超文本标记语言 (Hypertext Markup Language)，是万维网上的标准语言，它把不同的计算机上的资源有机地联系在一起，形成一个整体 (Web 网页)，以方便用户访问。

HTML 规范定义了一系列标识符，这些标识符用于控制网页出版的文档格式、字符格式、段落格式等，还包括为达到网页交互效果而对脚本语言、Java 小程序、CGI 程序等的支持和调用的规则。这些标识符是非常简单易学的。HTML 文档就是一个文本文件，可以使用任何一个文本编辑器进行编写。它通过浏览器来解释执行。

动态 HTML (DHTML) 是近年来网络发展进程中最振奋人心也最具实用性的创新之一。什么是动态 HTML 呢？动态 HTML 作为浏览器自带的功能，实际上只是一种概念，它并不属于一种专门技术 (如 JavaScript, ActiveX)，它不是标签，不是插件，也不是浏览器。它是一种通过各种技术的综合发展而得以实现的概念 (当然，不同的浏览器实现的程度也不同)，这些技术包括 JavaScript, VBScript, DocumentObject Model (文件目标模块), Layers 和 Cascading Style Sheets (CSS, 层叠样式表, 简称样式表) 等。

目前，动态 HTML 主要包括 3 部分内容：HTML 规范、Script 语言 (JavaScript 和 VBScript, 其中 JavaScript 使用得更为广泛，本书将以 JavaScript 语言为例) 和 CSS。在本书中我们将结合范例为你详细讲解这些内容。在第 1 章里，将首先向你介绍 HTML 4.0 规范。

1.1 HTML 4.0 简介

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、增强的表单，并且为残疾人提供了访问的可能。下面我们介绍一下 HTML 4.0 的新特点。

1.1.1 HTML 4.0 的新特点

国际化

HTML 4.0 可以使用各种语言编写网页文档，并易于在全世界范围内发布。HTML 4.0 设计过程中的一个重要步骤是采用了 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。

HTML 4.0 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……。

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。HTML 4.0 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、框架等的长名描述。

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。

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。

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。

样式表

样式表简化了 HTML 标记，并且减轻了 HTML 对表现形式的责任。它们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，制作者很少能控制网页绘制过程。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。

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>
```

保存，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。

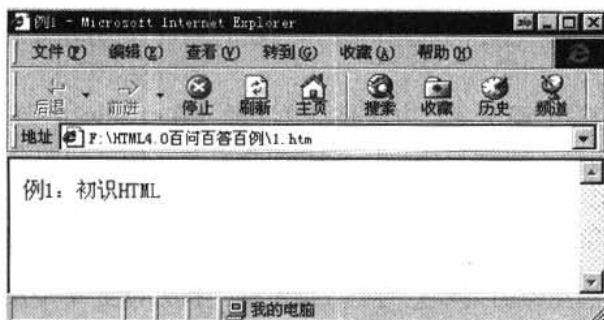
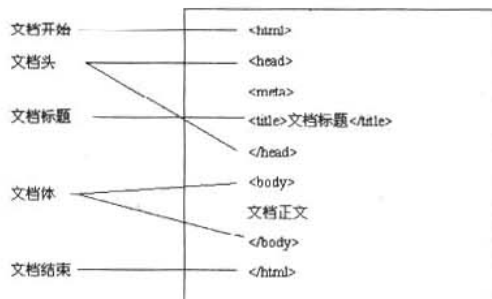


图 1.1 初识 HTML

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，这些标识符均包含在尖括号< >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，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，并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的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。

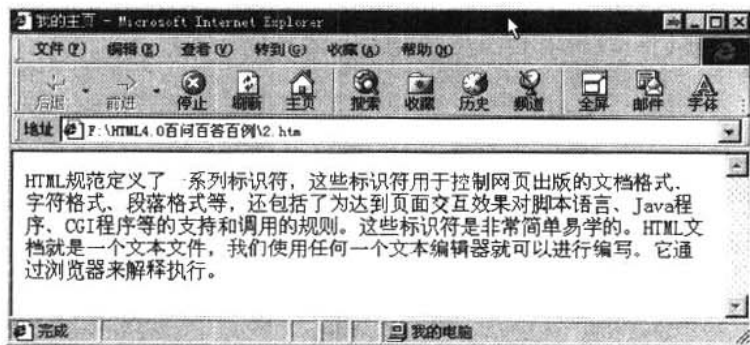


图 1.2 HTML 文档

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`</htMl>`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。
- 如果在 HTML 文档中嵌套了脚本代码（使用`<Script>`/`</Script>`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注释部分（已经使用黑体标识出来）。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>
```

```
<head>
<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>之间的文档头部分的内容是用来描述网页的内容的, 对网页的外观没有实际的影响。这一部分称为文档头。

1.2.1 可在文档头部分使用的标识符

在文档头部分可以使用的标识符有:

1 <title></title> 指示文档的标题。

2 <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。

3 <isindex> 指示该文档是允许查找的网关文件。它有如下属性:

- prompt="..." 查找字段的提示信息。

4 <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。

5 <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)。

6 <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">说明所用文字及语言。

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, 以便于搜索引擎机器人查找、分类 (目前几乎所有的搜索引擎都使用网上机器人自动查找 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 (搜索引擎籍以分类的关键词), 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”时, 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”时, 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 中的 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, 26 Feb 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。

meta 还有很多功能，你可以在以下站点进一步查询：

```
http://webdeveloper.com/categories/html/html\_metatag\_res.html
http://vancouver-webpages.com/meta/
http://www.nlc-bnc.ca/ifla/II/metadata.htm
```

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>例 7</title>
<meta http-equiv="refresh" content="20; url=otherpage.htm">
</head>
<body>
</body>
</html>
```

代码是如何工作的

代码中 Content 属性值中第一个值表示该网页停留的时间，单位是秒，URL 的内容是要转到的网页地址，可以是相对地址，也可以是绝对地址：

```
<meta http-equiv="refresh" content="20; url=otherpage.htm">
```

这个例子的效果是“在该网页停留 20 秒之后自动跳转到网页 otherpage.htm”，示例显示效果如图 1.3 和图 1.4 所示。



图 1.3 初始网页

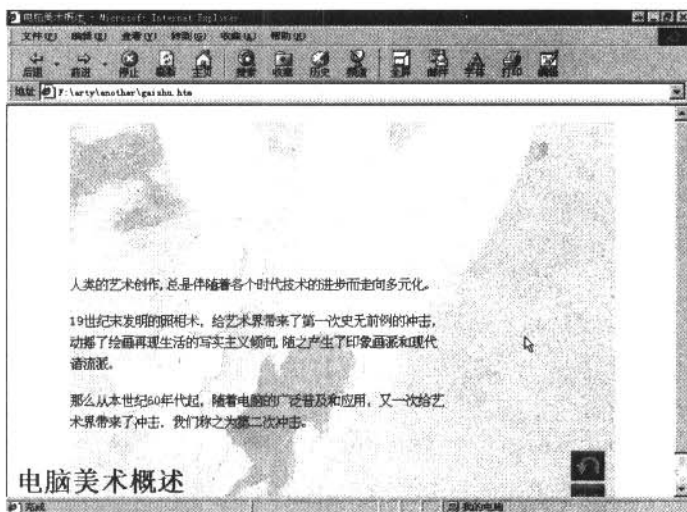


图 1.4 跳转到的网页

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

定义背景色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。

定义背景图像的语法如下：

```
<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>例 7</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>
```

首先保存，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不同。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。

以上代码对应的网页显示效果如图 1.5 所示。