

面向 21 世纪高职规划教材

孙永道 主编

电子商务设计实务



机械工业出版社
China Machine Press

513.36
132
面向 21 世纪高职规划教材

电子商务设计实务

主 编	孙永道		
副主编	潘维芹	刘岳兰	
参 编	任 文	李冀蜀	杨 平
	张 岚	杨荣琴	
主 审	张国勋	贾建中	

机 械 工 业 出 版 社

本书共分 12 章,由基础篇、高级篇和应用篇 3 大部分组成。基础篇从第 1 章到第 5 章,主要包括 HTML 语言、CSS 样式表设置、文档对象模型、脚本语言和动态 HTML 等开发电子商务应用必须具备的基础网页设计知识;高级篇从第 6 章到第 8 章,包括 ASP 技术、ADO 技术、Web 服务器配置等具有一定难度的内容,是从事电子商务开发和开发交互式网站必须掌握的高级技能;应用篇从第 9 章到第 12 章,包括电子商务网站的模式、安全认证、产品目录、购物篮等电子商务的主要组成部分,并结合实际开发例子,对电子商务的主要模块进行了模拟。

本书是 21 世纪高职高专规划教材,其特点是:内容全面,技术含量高。

本书主要适合于高职高专电子商务、计算机应用、网络等专业的电子商务开发课程的教材,建议在学生已经学习过网页设计基础知识和电子商务基本理论知识之后再开设这门课程较为适宜。本书也可供从事电子商务网站开发的专业人员参考,还可作为高职高专、函大、成人学院等大专层次学校的教材和培训教材。

由于各个院校的课程开设可能不尽相同,实际应用时可以根据自己的实际情况选择部分或全部内容。

图书在版编目 (CIP) 数据

电子商务设计实务/孙永道主编. —北京:机械工业出版社, 2002.5

面向 21 世纪高职规划教材

ISBN 7-111-10017-4

I. 电... II. 孙... III. 电子商务—网站—设计—高等学校: 技术学校—教材 N. TP393.092

中国版本图书馆 CIP 数据核字 (2002) 第 022964 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑: 邓海平 王世刚 版式设计: 张世琴 责任校对: 李秋荣

封面设计: 陈 沛 责任印制: 路 琳

北京机工印刷厂印刷·新华书店北京发行所发行

2002 年 7 月第 1 版·第 1 次印刷

1000mm×1400mm B5·11.875 印张·459 千字

0 001—4 000 册

定价: 28.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话 (010) 68993821、68326677-2527

封面无防伪标均为盗版

前 言

未来的世界将是一个以网络技术和知识经济为核心的信息化世界。随着计算机技术的迅速发展及网络技术的广泛普及,电子商务将成为最主要、最核心的经营方式。电子商务是人类社会经济、科技文化发展的必然产物,电子商务是未来社会信息化的主要标志,是信息化社会商务的主要模式。

目前,许多大中专院校争相开设电子商务专业,但有关电子商务的教程多数是有关电子商务理论方面的。笔者在从事电子商务教学的时候也深深地体会到需要有一本有关电子商务平台开发的综合性教程。本书正是从这方面出发,编写了这本电子商务开发教程。

本书作为面向 21 世纪高职规划教材,其特点是没有过多的讨论电子商务的理论,而将重心放在电子商务应用的开发上。书中从最基础的 HTML 语言讲起,覆盖了样式设计、脚本语言、后台数据库和电子商务基本组成部分的开发技术。本书的内容分为基础知识、高级技术和应用技能三个部分。基础知识包括 HTML 语言、样式设计等基础知识;高级技术包括 ASP 服务器端应用开发、后台数据库技术等难度较高的内容;应用技能包括电子商务主要组成部分的开发实例等电子商务中必须具备的组成部分。

本书主要适合于高职电子商务、计算机应用、网络等专业的电子商务开发课程的教材,建议在学生已经学习过网页设计基础知识和电子商务基本理论知识之后再开设这门课程较为适宜。本书也可作为高职、函大、成人学院等大专层次学校的教材和培训教材,并可供从事电子商务网站开发的专业人员参考。

本书由邢台职业技术学院孙永道、辽宁仪器仪表工业学校潘维芹、湖南工业职业技术学院刘岳兰、甘肃国防科技工业学校任文编写。本书由孙永道主编,潘维芹和刘岳兰担任副主编。张国勋和贾建中担任主审。

在本书的编写过程中,我们的同事给了我们很大的帮助,在此对他们的支持表示衷心的感谢!另外,也感谢我们的几位学生(安新利、程富强、阮冬娥、王燕、赵静等)在排版、校对过程中对我们的帮助!

由于编者水平有限,且电子商务是一门新的学科,书中一定存在不少的错误和疏漏,请各位读者批评指正!

编 者
于河北邢台

目 录

前言

第 1 部分 基础篇

第 1 章 HTML 语言 1

1.1 什么是 HTML 语言 1

1.1.1 什么是 HTML 语言 1

1.1.2 HTML 的工作环境 3

1.2 HTML 基本结构 5

1.2.1 HTML 的结构框架 5

1.2.2 HTML 标记的组成 10

1.3 HTML 中常用的标记 10

1.3.1 超级链接 11

1.3.2 列表 15

1.3.3 段落与分隔符 20

1.3.4 字体 23

1.3.5 表格 27

1.3.6 框架 38

1.3.7 表单 42

习题 48

第 2 章 CSS 级联样式表 50

2.1 什么是级联样式表 50

2.1.1 CSS 与 HTML 50

2.1.2 浏览器对 CSS 的支持 51

2.1.3 CSS 的优点 51

2.2 CSS 的创建 51

2.2.1 样式的基本格式和用法 51

2.2.2 类选择器 53

2.2.3 ID 选择器 54

2.2.4 上下文选择器 54

2.2.5 建立样式文件 55

2.3 CSS 的引用 55

2.3.1 整页套用样式 55

2.3.2 样式定义中引用样式 文件 56

2.3.3 样式文件的引入 57

2.3.4 使用局部样式 58

2.3.5 样式的混合使用 59

2.4 CSS 的继承性 59

2.5 CSS 的优先顺序 60

2.5.1 不同引用方式的优先 顺序 60

2.5.2 不同选择器的优先顺序 61

2.6 在脚本语言中使用样式 61

2.7 CSS 滤镜特效 62

2.7.1 滤镜的创建 64

2.7.2 外观滤镜 64

2.7.3 过渡效果滤镜 71

习题 74

第 3 章 浏览器文档对象模型 75

3.1 IE 中的文档对象模型 75

3.1.1 什么是文档对象模型 75

3.1.2 IE 中的对象模型结构 76

3.2 Window 对象 76

3.2.1 Window 属性 76

3.2.2 Window 对象的方法 78

3.2.3 Window 的事件 82

3.3 Document 对象 83

3.3.1 Document 对象的属性 83

3.3.2 Document 对象的方法 87

3.3.3 Document 对象的事件 88

3.4 History 对象 89

3.5 Navigator 对象 90

3.6 对象模型的事件处理机制	92
3.6.1 事件类型	93
3.6.2 事件处理方法	94
习题	95

第4章 脚本语言 96

4.1 脚本语言基础	96
4.1.1 什么是脚本语言	96
4.1.2 JavaScript 简介	97
4.1.3 JavaScript 与 Java 的区别	100
4.2 JavaScript 基本数据结构	101
4.2.1 基本数据类型	102
4.2.2 函数与方法	108
4.2.3 流程控制	118
4.2.4 事件处理	123
4.3 基于对象的 JavaScript	127
4.3.1 对象的基础知识	128
4.3.2 常用对象的属性和方法	129
4.3.3 创建新对象	132
4.3.4 使用内部对象系统	138
4.3.5 窗口对象	141
4.4 Web 页面信息的交互	145
4.4.1 表单的基础知识	145
4.4.2 利用框架实现交互功能	146
习题	150

第5章 DHTML 与 XML

语言	151
5.1 什么是 DHTML 语言	151
5.1.1 DHTML 的结构	151
5.1.2 DHTML 的特点	153
5.1.3 浏览器对 DHTML 的扩展	156
5.2 XML 语言简介	156
5.2.1 HTML 的不足	156

5.2.2 XML 的产生	157
5.2.3 XML 的特点	158
5.2.4 XML 文档的基本结构	159

第2部分 高级篇

第6章 ASP 技术基础 162

6.1 ASP 简介	162
6.1.1 什么是 ASP	162
6.1.2 ASP 的运行机制和运行环境	163
6.2 如何使用 ASP	163
6.2.1 如何编写和运行 ASP 网页	163
6.2.2 ASP 中使用多种脚本语言	164
6.2.3 用 HTML 和 ASP 混合编程	166
6.2.4 如何在 ASP 中使用变量和常量	168
6.2.5 如何在 ASP 中编写脚本函数	171
6.2.6 使用组件和对象提高 ASP 的性能	174
6.2.7 使用集合管理对象和变量	175
6.2.8 ASP 网页中对象和变量的作用域	178
6.2.9 使用 ASP 指令处理特殊任务	181
6.3 ASP 高级技巧	182
6.3.1 如何向浏览器发送信息	182
6.3.2 向浏览器发送客户端脚本	186
6.3.3 ASP 网页中包含其它文件	187
6.3.4 用 ASP 脚本访问表单	

元素	189
6.3.5 ASP 脚本的错误处理...	192
6.3.6 应用程序管理	193
6.3.7 管理会话	194
6.3.8 提高 ASP 应用的安全性	198
6.4 特殊的 Global.asa 文件	201
6.4.1 global.asa 中的应用 程序事件	202
6.4.2 global.asa 中的 session 事件	202
6.4.3 global.asa 中的<object> 声明	202
6.4.4 global.asa 中的 Type- Library 声明	203
6.5 服务器端指令	204
6.5.1 #config 指令	204
6.5.2 #echo 指令	204
6.5.3 #exec 指令	206
6.5.4 #flastmod 指令	206
6.5.5 #fsize 指令	207
6.5.6 #include 指令	207
6.6 ASP 中使用内建对象	208
6.6.1 Application 对象	208
6.6.2ObjectContext 对象	212
6.6.3 Request 对象	214
6.6.4 Response 对象	218
6.6.5 Server 对象	220
6.6.6 Session 对象	222
6.7 ASP 中使用外部对象 和组件	224
6.7.1 Ad Rotator 组件	224
6.7.2 Browser Capabilities 组件	228
6.7.3 Content Rotator 组件	231
6.7.4 Page Counter 组件	234
6.7.5 Counters 组件	236

习题	238
----------	-----

第 7 章 后台数据库技术

7.1 ADO 简介	239
7.1.1 ADO 开发 Web 应用...	239
7.1.2 ADO 数据库解决 方案	240
7.1.3 ADO 数据库编程 模型	240
7.1.4 Web 数据库应用 基础	243
7.2 Connection 对象	249
7.2.1 Connection 对象的 属性	250
7.2.2 Connection 对象的 方法	255
7.3 Command 对象	259
7.3.1 Command 对象的 属性	259
7.3.2 Command 对象的 方法	263
7.4 RecordSet 对象	266
7.4.1 RecordSet 对象的 属性	267
7.4.2 Recordset 对象的 方法	276
7.5 Field 对象	289
7.5.1 Field 对象的属性	290
7.5.2 Field 对象的方法	294
7.6 Property 对象	295
7.7 Error 对象	295
7.8 Parameter 对象	297
7.9 ADO 的集合	298
习题	299

第 8 章 Web 服务器的安装 与配置

8.1 Web 服务器基础知识	300
-----------------------	-----

8.1.1 什么是 Web 服务器 ...	300	习题	335
8.1.2 常见的 Web 服务器 介绍	301	第 10 章 安全认证	336
8.2 安装与配置 PWS	303	10.1 安全认证的必要性和 手段	336
8.2.1 PWS 的安装	303	10.1.1 电子商务的安全性 ...	336
8.2.2 PWS 的配置	307	10.1.2 电子商务安全性的解 决方案	337
8.3 IIS 的安装与配置	314	10.2 安全认证的实现	340
8.3.1 IIS 的安装	314	习题	341
8.3.2 服务器设置	314	第 11 章 产品目录	342
8.3.3 站点设置	319	11.1 建立产品目录的必要性 ...	342
8.3.4 虚拟目录设置	325	11.2 产品目录的实现实例 ...	343
习题	326	习题	354
第 3 部分 应用篇		第 12 章 购物篮的创建	355
第 9 章 商务网站的模式与构 建过程	327	12.1 网上的购物篮和超市中的 购物小车	355
9.1 商务网站的工作模式	327	12.2 利用 Session 变量创建购 物篮	355
9.2 构建商务网站的基本步骤 ...	330	习题	368
9.3 商务网站的基本组成	333	参考文献	369
9.3.1 电子商务的基本组 成	333		
9.3.2 电子商务网站的组 成	335		

第 1 部分 基 础 篇

第 1 章 HTML 语言

1.1 什么是 HTML 语言

1.1.1 什么是 HTML 语言

HTML 是 Hyper Text Markup Language 的缩写,意思是超文本标识语言,用来描述万维网 WWW (World Wide Web) 上超文本文件。它主要是在文本文件的基础上,加上一系列的标识符形成所谓的超文本文件,当用户使用浏览器下载文件时,就把这些标识符解释成相应的含义,按照一定的格式将被标识的文件显示,而 HTML 的标识符本身并不显示。

HTML 文件可以用两种方式产生,一种是用文档工具编写 HTML 文件。由于 HTML 文件是一种纯文本文件,故可用任何文本编辑器来编辑它。例如 WORD、EDIT、“记事本”等都用来建立或编辑 HTML 文件;另一种是使用专用的 HTML 编辑器,如 FrontPage、Dreamweaver 等,它们都具有方便快捷、所见即所得的特点。

值得注意的是,目前最常用的 Netscape 和 IE 浏览器并不完全兼容,故有的标识符只能被其中一种浏览器所识别。

1.1.1.1 HTML 语言的产生

超文本的概念并不新鲜,早在 20 世纪 40 年代,Vaneever Bush 就提出了这个术语。但超文本技术的实现则是最近几年的事。此外,将超文本概念引入网络并从分布的计算机系统上提供链接的信息,也是到最近几年才变成现实。

HTML 是 SGML (Standard Generalized Markup Language,标准通用标记语言)的子集,SGML 则是一种功能更为强大的文件表示、管理和编排的语言,其草稿公布于 1980 年,1986 年成为 ISO 作为文件标准的描述语言。

HTML 是一种超文本的标识语言。标识语言的描述叫做文档类型定义 (DTD),目前 HTML 的 DTD 支持基本的超媒体文档的创建及布局。HTML 的 DTD 有如下 4 种版本:

- HTML1.0 版 其版本主要用于制作超文本链接,它已被 HTML2.0 版代

替。

- HTML2.0 版 其版本是 Internet 工程任务组认可的标准，其说明书定义了允许用户显示内联图像和使用交互形式的特征。

- HTML3.0 版 其版本与许多浏览器销售商创建的最新 HTML 标签不相容。HTML3.0 版本也叫 HTML+，由许多附加部分组成，这些附加部分都是没有被认可的 HTML 标准。1995 年 6 月，3.0 版本走到了尽头。1996 年 5 月，3.2 版本的草案出现了，其中只包含对 HTML 最流行的扩充。

- HTML 4.0 版 其版本是 HTML 的最新版本，其代号叫“Cougar”。像 HTML3.2 版一样，HTML 4.0 版中加入了很多 Web 设计的常规实例，正规化了一些作为 HTML3.2 版扩展的指标和特性。

HTML4.0 版与较早版本的 HTML 最大的区别在于字符集。4.0 版已不限于仅使用有限的国际字符，它使用 Unicode（统一码）的字符集，从而引进了上千个不同字符。

1.1.1.2 HTML 语言的特点

HTML 文档独立于平台，对多个平台兼容。只要你拥有一个可以阅读和解释 HTML 的浏览器，就可以在任何平台上阅读 HTML 文件。HTML 虽然是一个标识性的语言，但是它学习起来非常容易，甚至比任何一种计算机语言都简单易学。

创建 HTML 文件并不需要特殊的软件，只要一个字符编辑器就可以了。当然，专门的 HTML 编辑器生成 HTML 文件会更加方便快捷。每一个 HTML 文件都不会太大，它能够尽可能快地通过网络来传输和显示，不需要加入字体或格式等其它各种控制信息。当然，如果需要的话，可以加入一些控制信息，对相关内容进行修饰，可使文本显得更加美观，但为此付出的代价是文件的下载速度和显示速度将会降低。

HTML 文件是用 ASCII 码写成的，它由内容和标识两部分组成。大多数 HTML 标签的书写形式为：

<标签名>文件内容</标签名>

标签名通常写在一对<>内。一般来说，HTML 的标签都包含起始标签和结束标签两部分，其差别在于结束标签前要加“/”。但并不是所有的 HTML 标签都有起始标签和结束标签，有些标签是没有结束标签的。对于 HTML 标签，大小写可混用，其效果是相同的。例如：<body>、<Body>和<BODY>都代表相同的标签。

1.1.1.3 HTML 文件的创建

HTML 文件是一种标准的文本文件，任何文字处理软件都可以用来创建此文件。例如，在 Windows 系统中，就可以用“记事本”创建 HTML 文件。

对 HTML 文件扩展名的规定为：在 DOS 操作系统下，其扩展名为 .htm；在

UNIX 操作系统下, 扩展名为 .html; 在 Windows 环境下, 以上两种扩展名均可使用。下面是一个最简单的 HTML 文件:

```
<html>
  <head>
    <title>这是一个最简单的 HTML 文件</title>
  </head>
  <body>
    HTML 文件的内容就从这里开始书写……
  </body>
</html>
```

创建 HTML 文件的方法非常简单, 可以用 Windows 的记事本创建该文件并保存扩展名为 htm 的一个文件, 启动 IE 并在地址栏中输入刚才创建的文件全名即可看到最终的结果了。

事实上, 借助浏览器本身的功能, 轻而易举地就可以查看浏览器中显示的 HTML 文件的原始代码, 方法是执行 IE 的“查看/源文件”菜单, 系统就会启动记事本并将网页的源代码装入其中, 但只能看到 HTML 代码, 而不能看到其他的服务器端的脚本代码。有关服务器端脚本代码的编写请参考第 6 章的内容。

1.1.2 HTML 的工作环境

1.1.2.1 URL

Internet 是一个相当庞大的网络系统, 当访问 Internet 上的资源时, 必须有一个标识符来唯一标识这些资源, 这个标识符就是 Internet 的统一资源定位器 (Uniform Resource Locator, URL), 也就是人们常说的网址。每个 URL 都包括一些特定的部分以标识网页所在的具体位置。URL 的格式为:

协议名称//主机名: 端口号/路径/文件名

协议名是传输资源时所用的协议标准, 常见的有 HTTP (超文本传输协议)、FTP (文件传输协议)、TELNET (远程登录协议)、NNTP (网络新闻传输协议) 和 mailto (访问邮件的传输协议) 等。

主机名是要访问资源所在服务器的域名, 例如 www.yahoo.com.cn 代表 Yahoo 公司的主机名。端口号是网络通信中的逻辑端口, HTTP 的端口号默认为 80。由于协议有端口号的默认定义, 所以端口号常常省略。

路径是所访问资源在服务器中的位置, 这个位置常常是虚拟路径 (相对于服务器的路径)。例如 PWS 的默认根路径为 C:\Interpub\wwwroot, 如果要访问的站点的相对路径为 myweb/web1, 则站点的物理路径为 C:\Interpub\wwwroot\myweb\web1\。

文件名是要访问资源 (比如网页文件) 的名称, 必须用全名 (包括文件名和

扩展名)。

注意：URL 要求用户在输入时不要有空格；另外，在 URL 书写当中的“/”，切记不能与 DOS 命令中的“\”相混淆。下面是一个完整的 URL 的例子：

`http://www.edmweb.com:80/steve/cyberdict.html`

其中，http 是超文本传输协议；www.edmweb.com 服务器的域名；80 是端口号；steve 是资源在服务器中的虚拟地址；cyberdict.html 是要访问的网页文件的名称。

实际上，可以更通俗地理解 URL 的含义。如前面的例子表示按照 HTTP 协议，通过计算机的 80 号端口，访问 Internet 当中的一台名为 www.edmweb.com 服务器中的虚拟地址为 steve 中的网页文件 cyberdict.html。

URL 的表示方式有两种，分别称作相对 URL 和绝对 URL。绝对 URL 是用如上所示的完整的表示方法表示资源的位置。当我们从一个网站访问另外一台远程计算机上的资源时，这种方法就是必须的。相反，相对 URL 表示的是同一站点中的资源位置。绝对 URL 和相对 URL 与资源管理器中的绝对路径和相对路径的作用是类似的。例如：我们要在网页 a.htm 中建立两个超链接，一个是访问该网页所在位置的子目录 Sub 下的文件 b.html（我的简历），另一个是 `http://www.yahoo.com.cn/index.htm`（雅虎主页），则两个超级链接分别：

`<a href="sub/b.html">我的简历</a>`

`<a href="http://www.yahoo.com.cn/index.htm">雅虎主页</a>`

1.1.2.2 Web 服务器

Web 服务器就是 WWW 用来储存 Web 资源的服务器，不同的 Web 资源存放在全球不同的服务器上。作为用户，完全不用考虑所需要的资源存在何处。用户浏览其上的内容是通过浏览器来进行的。

1.1.2.3 网页

网页 (Web Page) 是组成 Web 网站的基本元素，它是由超文本标记语言对各种多媒体信息的特殊格式化设置后组成的。网页中可以包含文字、图形图像、动画、视频、音频、脚本等信息。

1.1.2.4 网站

网站 (Web Site) 是相关网页的集合，如一个公司的网站，就是将公司的各种资源信息放在网页上，在由这些网页组成一个统一的、相关的整体，即网站。网站一般放在 Web 服务器上。

1.1.2.5 超文本与超媒体

超文本 (Hyper Text) 是包含超级链接的文本，即一旦文件内拥有超级链接的文件导引功能，我们称此文件为“超文本”。而超媒体 (Hypermedia) 突破了超文本超级链接只包括文本的限制，它可以链接到声音、图像、动画、图片和虚拟

现实等多种媒体信息。也就是说，如果文件内拥有超级链接功能的链接点是一些图片、动画、声音、图像和电影等多媒体元件，这个文件我们就称为“超媒体”，它实现了比超文本更强的功能。

HTML 是一种超文本标识语言，是用来描述 WWW 上超文本文件的。它主要是在原来文本文件的基础上，加上一系列的标识符号描述其格式，形成网络文件，当用户使用浏览器下载时，就把这些标识解释成相应的含义，按照一定的格式，将这些被标识的文件，显示在屏幕上。而 html 的标识符本身并不显示。

利用 HTML 语言，可以将 Internet 上连接的不同地区的服务器上的文件信息链接起来：有的标识是去链接另外一个文件，有的是形成一张表格，有的是接受用户信息等等。借助于这些标识，用户在浏览器上看到的将是五彩缤纷的画面，而不再是单纯呆板的文本。利用 HTML 语言也可以将声音、图像、甚至视频连接起来。此外，HTML 还可以和数据库进行连接，来满足用户的查询需求以及为用户交互的功能等等。

1.2 HTML 基本结构

本节通过一个简单的 HTML 文件，让大家对 HTML 的基本结构及其构成要素有一个基本的认识。

1.2.1 HTML 的结构框架

以下是一个简单的 HTML 文件：

```
<html>
  <head>
    <title> 网页的标题 </title>
  </head>
  <body>
    网页的内容，很多标记都放于此
  </body>
</html>
```

可以看出，HTML 文件的基本架构包含如下几部分：

```
<html>
  <head>
    <title>    </title>
  </head>
  <body>
    </body>
</html>      标示着一个 HTML 文件的结束
```

在此结构中,包含了4个最基本的标签:HTML、HEAD、TITLE 和 BODY。下面我们分别介绍它们的用法。

1.2.1.1 HTML 标签

在任何一个HTML文件里,HTML 标签总是最先出现的,它告诉浏览器,这是一个HTML文件,除此而外,该标签并无其它任何属性。<HTML>和</HTML>成对出现,整份文件处于标记<HTML>与</HTML>之间,内含两个结构标签:<HEAD>和<BODY>。

1.2.1.2 HEAD 标签

HTML 的文件分为两部分,开头和文本。由<HEAD>至</HEAD>称为开头。<HEAD>标签属于HTML文件的结构标签,开头部分用以加载关于文档的重要信息。写在<HEAD>和</HEAD>中间的文本,如果又被写在<TITLE>和</TITLE>标签的中间,则该文本作为网页的标题,被显示在这个网页窗口的标题栏上;如果不写在<TITLE>和</TITLE>标签的中间,则该文本作为正文的头部,显示在文本之前。一般<HEAD>标签中的文本内容应尽量精炼。

1.2.1.3 TITLE 标签

<TITLE>所标示的是文件的标题。该标题就是通常所说的网页标题,出现在浏览器标题栏上,其作用是说明这个网页的内容或用途。<TITLE>标签中的内容应该是对整个HTML文档简洁而明了的概括,要短小精炼,一目了然,让人一看就明白该网页的内容是关于哪方面的。一个网页只能有一个网页标题,网页标题要写在<TITLE>和</TITLE>标签之间,并且<TITLE>标签应包含在<HEAD>和</HEAD>标签之中,<TITLE>标签中不能含有其它标签。

1.2.1.4 BODY 标签

<BODY>至</BODY>间的内容是网页的主体,是网页的核心部分,主体被显示在浏览器窗口中,从某种意义上讲,它相当于高级语言的程序体部分。<BODY>标签中,内含所有文字、图片、表格和多媒体文件等组合成文件内容的组件,几乎所有关于文件内容的格式标签都放在该标签中,而<BODY>标签一定要在<HTML>标签之后。

除此之外,还有两个标签经常用到,它们是<META>标签和<COMMENT>标签,下面作一简单介绍。

1.2.1.5 META 标签

<META>标签在HTML各种标签中,算得上是一个比较特别的标签。其主要作用是定义一些其它HTML标签所没有定义的文件信息,这些信息并不是指文件的内容,而是文件本身的信息。<META>标签所定义的信息,能够在服务器端或浏览器端存取。例如HTTP通信协议的标头参数,这些参数能够使用HTTP-EQUIV 属性定义属性和值,并可以储存在HTTP标头,通过这个机制提供一个管

道能够将信息传送给其它的 HTML 文件。

在 IE 4.0 以上的版本中, <META> 标签还增加了一些 HTML 文件显示的特效, 不用编写任何脚本, 就可以轻松建立网页的过渡效果。

<META> 标签的属性有三个, 分别是 NAME、HTTP-EQUIV 和 CONTENT。其中 NAME 属性定义文件的名称, 如果未指定, 该名称可以看作是 HTTP-EQUIV 属性的名称。HTTP-EQUIV 属性定义需要传递参数的名称, 参数名称在命名上虽然不区分大小写, 但一定要注意, 不能和 Web 服务器默认的标头参数名称相同。另外, 如果只设定了 name 属性的值, 而没有设定 http-equiv 属性的值, 则该数据是不会新增到 HTTP 标头参数中, 而且此属性提供有默认的值。例如, 利用 refresh 可以轻松的建立转向、过场的网页的特殊效果。content 属性要和 name 或 http-equiv 属性相对应, 它就是参数名称的具体值。

<meta> 标签在使用时, 不仅能够单纯的定义资料, 而且配合 IE 4.0 或以上版本所提供的特效功能, 实现过场特效。

<meta> 标签的基本用法大约有如下几种:

(1) 定义 HTML 文件本身的属性和值 例如:

```
<meta name="generator" content="wordpad">
```

该设置表示由 generator 指定编辑 html 所使用的编辑工具, 编辑工具为“WordPad”写字板。

(2) 设置 HTML 文件默认的语系 HTML 文件内容因使用不同语系的文字而有所不同。假定你目前使用的是一个日文浏览器, 此时, 如果你再浏览中文的 HTML 文件, 看到的肯定是一堆乱码。如果要想浏览器能够正确的显示有关 HTML 文件的内容, 就必须在<meta> 标签中设定语系内码, 使用 http-equiv 属性传送 HTTP 通信协议的标头, 例如, 简体中文是 GB2312, 繁体中文为 big5。其设置方法如下:

```
<meta http-equiv="content-type" content="text/html; charset=GB2312">
```

```
<meta http-equiv="content-type" content="text/html; charset=big5">
```

(3) 新增 HTTP 通信协议的标头参数 如果需要新增 HTTP 标头参数, 则可以利用如下方法:

```
<meta http-equiv="expires" content="005-01-2001">
```

```
<meta http-equiv="from" content="http://www.163.com/">
```

通过这种方法, 便可将参数值、表单元素的值放在标头传送的下一页文件。通过上述的<meta> 标签设置了两个标头参数, 此时的标头参数就会多出两项内容: expires: 005-01-2001 和 from: http://www.163.com/。

(4) 自动转向到其它的 HTML 文件 如果网页中的文件路径有变更, 这时, 你有可能首先想到的是更改所有超链接的目的地, 事实上完全没有必要。我们可

以利用 http-equiv 属性的值 refresh, 在经过规定的时间后, 自动转向目的地的文件或网址。其设置方法如下:

```
<meta http-equiv="refresh" content="n";URL="URL">
```

这里, n 为秒数, 此设置的含义是: 经过 n 秒后, 将显示由 URL 指定的文件或网址。例如:

```
<html>
  <head>
    <title>自动转向其它 html 文件的演示</title>
    <meta http-equiv="refresh"
      content="10";url="http://www.gmdaily.com.cn">
  </head>
  <body>
    <center>
      <h2>请等待……</h2>
      网页位置发生变化! 10 秒后将自动到正确的网页!
    </center>
  </body>
</html>
```

(5) 定时更新 HTML 文件的内容 http-equiv 属性的值 refresh, 不但能在规定的时间之后自动转向其它的 HTML 文件或网址, 而且还能够在规定的时间后自动转向到设置的网址, 进行内容的自动更新; 如果未指定其它网址, 也可以在规定的时间内自动刷新自己。在聊天室中的聊天页面通常都要设置自动刷新功能, 一边聊天者能够及时得到聊天信息。设置刷新的格式为: <meta http-equiv="refresh" content="n">, 其中, n 为秒数, 即在经过 n 秒后便自动更新网页内容。下面的例子就是利用 <meta> 标签实现 HTML 文件的自动刷新。

```
<html>
  <head>
    <title>html 文件自动刷新演示</title>
    <meta http-equiv="refresh" content="5">
  </head>
  <body>
    <center>
      <h2>文件每隔 5 秒自动更新一次</h2>
      
      
```

```


</center>
</body>
</html>

```

(6) HTML 文件的过场效果 在 IE4.0 版及其以后各版本的浏览器中, <meta> 标签增加了网页切换功能, 就是利用 http-equiv 属性能够设定在进入或离开网页、进入或离开网站的时候, 可以使用不同的过场效果来显示下一页的 HTML 文件的内容。实现这一效果的基本语法如下:

```

<meta http-equiv="event" content="revealtrans
(duration=n,transition=no)">

```

这里, Event 为产生过场效果的值, 共有 4 个, 如表 1-1 所示; n 是秒数。用于定义经过 n 秒后才能完成全部显示过程; No 是过场效果的编号, 范围在 0 到 23 之间, 共有 24 个。各个代码的相应效果说明, 如表 1-2 所示。

表 1-1 过场事件表

Event 的值	说 明
Page-Enter	进入 HTML 文件 (网页) 时
Page-Exit	离开 HTML 文件 (网页) 时
Site-Enter	进入网站时
Site-Exit	离开网站时

表 1-2 过场效果

No	说 明	No	说 明
0	逐渐缩小长方形画面显示下一页	12	溶解式显示下一页
1	逐渐放大长方形画面显示下一页	13	从左右同时向中央逐渐显示下一页
2	逐渐缩小图形画面显示下一页	14	从中央同时向左右逐渐显示下一页
3	逐渐放大图形画面显示下一页	15	从上下同时向中央逐渐显示下一页
4	从下向上逐渐显示下一页	16	从中央同时向上下逐渐显示下一页
5	从上向下逐渐显示下一页	17	从右上向左下逐渐显示下一页
6	从左向右逐渐显示下一页	18	从右下向左上逐渐显示下一页
7	从右向左逐渐显示下一页	19	从左上向右下逐渐显示下一页
8	垂直像页窗显示下一页	20	从左下向右上逐渐显示下一页
9	水平像页窗显示下一页	21	随机水平线式逐渐显示下一页
10	垂直棋盘式显示下一页	22	随机垂直线式逐渐显示下一页
11	水平棋盘式显示下一页	23	随机选择 0~23, 每次都不同

1.2.1.6 HTML 中的注释

像很多计算机语言一样, HTML 文件也提供注释功能, 浏览器会忽略此标记