

网络应用系列



Web程序设计 从入门到精通

鸿志创作组 编著

科学出版社

网络应用系列

Web 程序设计从入门到精通

鸿志创作组 编著

科学出版社

1998

内 容 简 介

本书介绍了 Web 开发所涉及到的四种语言: HTML, VBScript, JavaScript 和 Java。对于每一种语言,着重介绍了它的基本语法元素和语句结构,并列举相应的程序示例。本书叙述简洁扼要,是快速掌握 Web 开发语言的首选读物。

本书适合 Web 程序员和高校计算机专业师生参考。

图书在版编目 (CIP) 数据

Web 程序设计从入门到精通/鸿志创作组编著. -北京: 科学出版社,
1998. 7

(网络应用系列)

ISBN 7-03-006440-2

I. Web… II. 鸿… III. 万维网. 软件开发 IV. TP393. 4

中国版本图书馆 CIP 数据核字 (98) 第 27763 号

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

北京双青印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1998 年 7 月第 一 版

开本: 787×1092 1/16

1998 年 7 月第一次印刷

印张: 19

印数: 1—3 400

字数: 435 000

定价: 29.00 元

前 言

随着 Internet 的迅速兴起和广泛使用,软件运行环境正在发生迁移。传统上在各种操作系统下运行的程序正在逐步挪到 Web 上运行。这样,对于软件开发者,包括所有为 DOS, Windows 和 Unix 等操作系统编写应用程序和系统程序的开发人员,掌握 Web 程序设计技术成为当务之急,本书正是为此目的编写。

全书共分四部分,分别讲解 Web 最通用的编程设计语言:HTML,VBScript,JavaScript 和 Java,按从简到繁的顺序编排。HTML 是 Web 的基本语言,所有 Web 文档都是基于 HTML 编制的。在 HTML 部分中,主要讲解了 HTML 基本概念、页面布局、文本布局、字体设计、表单设计、表设计、帧设计、移动文字设计、图像设计等内容。VBScript 是 Microsoft 推出的 Web 脚本设计语言,用它编写的程序直接把源代码嵌入到 HTML 文档中,由 Web 浏览器解释执行。在 VBScript 部分中,主要介绍了 VBScript 的语言基础、变量和数组、函数和过程、程序流控制等内容。JavaScript 是 Netscape 推出的 Web 脚本设计语言,其与 VBScript 类似。不同的是,VBScript 是在 Visual Basic 的语言基础上发展而来,而 JavaScript 是在 Java 语言基础上发展而来。在 JavaScript 部分中,讲解了 JavaScript 语言、函数、事件、对象和数组、内置对象、窗口对象、表格对象、Cookie 等。在本书最后一部分,讲解了 Java 语言。尽管 Java 相对于 C++ 来讲,Java 比较简单;但相对于前三种语言,Java 是最难的。这是因为,在 HTML 文档中,不是直接插入 Java 源代码,而是插入 Java 源代码编译生成的 .class 文件的引用。另外,它的语法结构也要复杂一些。在 Java 这一部分中,主要讲解了 Java 语法、小程序编程、图形编程、用户界面设计、多线程设计等内容。

本书是集体劳动的结晶。创作组的成员包括:魏源源、方兴、欧阳蒙、魏泱、刘烨、秦瑞芳、郭子键、刘里华、方未、李纪鸿、陈前进、胡鼎、魏小华、刘日丹、何玻等。

编 者

目 录

前 言

第一部分 HTML

1 初识 HTML	(3)
1.1 什么是 HTML	(3)
1.2 用什么来编写 HTML 代码	(3)
1.3 HTML 编辑器	(3)
1.4 基本概念	(4)
2 HTML 页面布局	(6)
2.1 HTML 的基本结构	(6)
2.2 定制页面的属性	(7)
2.3 <BODY>标记中的属性	(9)
2.4 创建超链接	(10)
2.5 开一个新的(浏览器)窗口	(11)
2.6 水平线	(11)
2.7 有关颜色的设定	(12)
3 HTML 文本布局	(14)
3.1 段落和换行	(14)
3.2 创建列表	(14)
3.3 预定的文本格式	(17)
3.4 其他与文本有关的标记	(19)
4 HTML 字体设计	(22)
4.1 控制字体的大小	(22)
4.2 特殊效果的字體	(23)
5 HTML 表单设计	(27)
5.1 创建和配置表单	(27)
5.2 CGI 的基本认识	(28)
5.3 创建表单的 HTML 标记	(28)
6 HTML 表设计	(33)
6.1 表的基本概念	(33)
6.2 表的 HTML 标记简介	(33)
6.3 表的 HTML 标记示例	(35)
7 HTML 帧的设计	(37)
7.1 关于帧(frame)和帧集(frame set)	(37)
7.2 创建帧的 HTML 标记	(38)
8 HTML 移动文字(marquee)设计	(41)

8.1	什么是 marquee	(41)
8.2	HTML 中的<MARQUEE>标记及其属性	(41)
9	HTML 图像设计	(44)
9.1	在页面中插入图像	(44)
9.2	在页面中使用图像的 HTML 标记简介	(45)
9.3	客户端图像映射图(Client Side Image Map)	(46)
10	HTML 增强标记功能	(47)
10.1	HTML 的<SCRIPT>标记	(47)
10.2	HTML 的<APPLET>标记	(49)
10.3	HTML 的<OBJECT>标记和<PARAM>标记	(50)
10.4	使用<OBJECT>标记和<PARAM>标记	(52)
10.5	HTML 多媒体页面设计标记	(53)

第二部分 VBScript

11	VBScript 语言基础	(57)
11.1	回顾 HTML 的<SCRIPT>标记	(57)
11.2	访问 VBScript 数据类型	(59)
11.3	判断变体中的数据类型	(60)
11.4	数据类型转换	(62)
12	VBScript 变量和数组	(64)
12.1	声明变量	(64)
12.2	变量的作用范围	(64)
12.3	禁止使用未声明的变量	(65)
12.4	数组操作	(66)
13	VBScript 函数和过程	(69)
13.1	VBScript 内建函数简介	(69)
13.2	使用 VBScript 日期和时间函数	(70)
13.3	使用消息框和输入框与用户交互	(71)
13.4	声明自定义函数	(72)
13.5	声明定制过程	(74)
14	VBScript 程序流程控制	(76)
14.1	用 If 和 Select 进行条件判断	(76)
14.2	用 Do...Loop 在程序中进行循环	(77)
14.3	使用 For...Next 循环	(78)
14.4	使用 While...Wend 循环	(80)
15	VBScript 客户端程序实例	(81)
15.1	响应客户端事件	(81)
15.2	对用户进入站点和离开站点作响应	(83)
15.3	改变窗口中的 HTML 文档	(84)
15.4	利用客户端脚本对 Form 元素进行验证	(85)

第三部分 JavaScript

16	JavaScript 概览	(91)
-----------	----------------------------	---------------

16.1	JavaScript 语言	(91)
16.2	JavaScript 扩展 Web 网页	(92)
16.3	JavaScript 开发环境	(94)
17	JavaScript 语言	(95)
17.1	变量和常量	(95)
17.2	JavaScript 变量的数据类型	(96)
17.3	JavaScript 操作符	(97)
17.4	JavaScript 的控制语句	(101)
18	JavaScript 函数	(107)
18.1	函数是什么	(107)
18.2	参数的作用	(107)
18.3	函数的返回值	(108)
18.4	定义和调用函数	(108)
18.5	全局和局部(本地)变量	(109)
18.6	方法	(109)
18.7	系统函数	(109)
18.8	共享方法	(111)
19	JavaScript 事件	(114)
19.1	导航行为和事件	(114)
19.2	表格输入和事件	(114)
19.3	JavaScript 事件句柄	(116)
19.4	event 对象	(122)
19.5	JavaScript 事件小结	(123)
20	JavaScript 对象和数组	(126)
20.1	JavaScript 对象	(126)
20.2	数组(Array)对象	(130)
20.3	JavaScript 提供的对象	(133)
21	JavaScript 内置对象及其使用	(136)
21.1	string 对象	(136)
21.2	Math 对象	(140)
21.3	Date 对象	(141)
22	JavaScript 窗口对象和元素	(145)
22.1	JavaScript 对象层次	(145)
22.2	Navigator 对象	(145)
22.3	窗口(window)对象	(146)
22.4	位置(location)对象	(150)
22.5	历史(history)对象	(152)
22.6	文档(document)对象	(153)
22.7	link 对象	(156)
22.8	anchor 对象	(156)
22.9	form 对象	(157)
23	JavaScript 表格对象	(158)
23.1	理解表格行为和方法	(158)

23.2	表格元素介绍	(158)
23.3	创建一个 HTML 表格	(160)
23.4	使用 JavaScript 表格(form)对象	(162)
23.5	JavaScript 表格对象层次	(163)
23.6	使用 JavaScript 处理表格	(166)
24	使用 cookie	(172)
24.1	在 cookie 中存储永久数据	(172)
24.2	cookie 的参数	(172)
24.3	cookie 的限制	(173)
24.4	JavaScript 和 cookie	(173)
24.5	使用 cookie	(175)
25	使用插件	(179)
25.1	Netscape 插件概述	(179)
25.2	插件与辅助应用程序	(179)
25.3	插件和 MIME 类型	(180)
25.4	在 HTML 中使用插件	(181)
25.5	插件和 JavaScript	(183)

第四部分 Java

26	Java 语言概述	(189)
26.1	Java 语言的特性	(189)
26.2	Java 与 C++ 的区别	(190)
26.3	Java 语言的开发环境	(191)
26.4	Java 的应用程序和小程序	(192)
27	Java 语法简介(一)	(195)
27.1	Java 语言的关键字	(195)
27.2	Java 语言的注释	(195)
27.3	Java 语言的标识符	(196)
27.4	Java 语言的数据类型	(196)
27.5	运算符	(200)
27.6	类型转换约定	(202)
27.7	流程控制语句	(203)
28	Java 语法简介(二)	(208)
28.1	类	(208)
28.2	接口	(216)
28.3	包	(218)
28.4	小结	(220)
29	Java 小程序编程概述	(221)
29.1	小程序的代码特点	(221)
29.2	小程序安全性模型	(221)
29.3	小程序的生存周期	(222)
29.4	小程序中的重要方法	(222)
30	Java 的图形操作	(223)

30.1	Java 的图形坐标系统	(223)
30.2	基本绘图方法	(223)
30.3	颜色	(229)
30.4	字体	(232)
31	Java 的用户界面设计(一)	(235)
31.1	AWT 包概述	(235)
31.2	常用组件类编程	(236)
31.3	容器类	(248)
32	Java 的用户界面设计(二)	(251)
32.1	一个布局管理器的示例	(251)
32.2	布局管理类概述	(252)
32.3	类 BorderLayout	(253)
32.4	类 FlowLayout	(253)
32.5	类 CardLayout	(254)
32.6	类 GridLayout	(255)
32.7	类 GridBagLayout 和 GridBagConstraints	(256)
33	多线程	(259)
33.1	线程概述	(259)
33.2	一个多线程示例	(260)
33.3	多线程的创建	(261)
A	预定义颜色	(266)
B	VBScript 语言参考	(270)
B.1	VBScript 数据子类型	(270)
B.2	VBScript 运算符	(271)
B.3	VBScript 函数	(273)
B.4	VBScript 语句	(283)
B.5	VBScript 常数	(287)
B.6	VBScript 编码规则	(290)

第一部分 HTML

- 1 初识 HTML
- 2 HTML 页面布局
- 3 HTML 文本布局
- 4 HTML 字体设计
- 5 HTML 表单设计
- 6 HTML 表设计
- 7 HTML 帧的设计
- 8 HTML 移动文字 (marquee) 设计
- 9 HTML 图像设计
- 10 HTML 增强标记功能

1 初识 HTML

1.1 什么是 HTML

HTML是超文本标记语言(HyperText Markup Language)的缩写,它是在 World Wide Web上描述页面的内容和结构的标准语言。WWW 网上最基本的传输单位是 Web 页面,它们都是用 HTML 语言来编写的。

HTML 编程就是用一大堆各种各样的 HTML 标记把在页面中显示的内容组织起来,这些标记用来指定哪些文本应该放大,哪些文本应该向右对齐,哪些文本或图像应该插入超链接,在页面中的什么地方加入图像等等。HTML 编程的结果就是一个 HTML 文件,其中包含了各种各样的“标记”和文本,浏览器能够理解这些标记,并把一个 HTML 文件解译出来,显示出漂亮的 Web 页面。

某些 HTML 标记还包含了各种属性。属性提供了有关该标记的特殊的细节信息。比如说,要在页面中插入一个图像,那么在 HTML 标记中必须包含的一个属性是指示该图像文件的路径和文件名。

1.2 用什么来编写 HTML 代码

为了编写 HTML 代码,不需要使用任何具有高级功能的程序,而只需要使用一个简单的文本编辑器即可。如果你是一名 Microsoft Windows 的用户,那么 Notepad(记事本)程序、Windows 95 的 WordPad(写字板)或者 DOS 的 Edit 程序就足以应付。UNIX 用户则可以使用 vi 或者 pico。Macintosh 用户也可以使用他们特有的文本编辑器。

HTML 文件实际上是以 ASCII 格式存储的文件。ASCII 是计算机能够识别的一种最基本的格式。HTML 文件的后缀名必须是 htm 或者 html,因为只有这样,浏览器才能识别。一般的文本编辑器不提供直接保存成 HTML 文件类型的功能,可以先将文件保存成纯文本文件(*.txt),然后再将它改名为*.htm 或者*.html 文件。

虽然仅仅使用简单的文本编辑器就可以编写 HTML 代码,但是如果你曾经这样做过的话,你一定会发现这是一个非常沉闷和繁琐的工作。除了不可避免的输入错误之外,你必须不时地在文本编辑器和浏览器之间切换,以查看你所编写的页面的效果。

1.3 HTML 编辑器

为了满足越来越多的 Web 站点开发人员和页面制作人员的需要,现在也出现了许多专门用于编写页面的软件,这些软件被称作 HTML 编辑器。在表 1.1 中列出了现在比较流行的 HTML 编辑器及其软件提供商。

HTML 编辑器简化了输入 HTML 标记和符号的一些枯燥的工作,并且它们还能保证这种工作的准确性。现在有很多的 HTML 编辑器可用,如微软公司在 1997 年推出的 FrontPage

97 就是一个很受欢迎的 Web 站点设计工具。

表 1.1 流行的 HTML 编辑器

HTML 编辑器	软件厂商
Hot Dog Pro	Sausage Software
HoTMetaL Pro	SoftQuad
NaviPress	Navisoft
FrontPage	Microsoft

FrontPage 97 包括 FrontPage Explorer 和 FrontPage Editor 两个部分,其中 FrontPage Explorer 提供了对 Web 站点强大的管理功能,如对各种文件的管理、对用户权限的设置等等,另外还提供了多种实用工具帮助管理者查看 Web 站点的内容和链接。

FrontPage Editor 是一个所见即所得的 HTML 编辑器。也就是说,当在 FrontPage Editor 中编辑页面的时候,所见到的效果与该页面在浏览器中的效果是一样的。不仅仅可以在其工作窗口中输入文本、插入图像、创建超链接、插入脚本和其他各种复杂的对象,还可以控制页面的布局。

FrontPage Editor 的图形化界面与常见的字处理程序(如 Microsoft Word)非常相似,这意味着熟悉 Word 的人可以很快地转向 Web 开发,而不用花费太多的学习时间。

另外,Microsoft 为其 Office 应用程序也提供了扩展 Internet 功能的附件。例如,熟悉 Microsoft Word 的用户可以使用 Internet Assistant for Word 将一个文档转换为 HTML 格式的文档,这样它就可以在浏览器中显示了。

Microsoft Office 97 支持在文档中创建超链接,并且提供了将文档直接保存为 HTML 文件类型的功能,这对于页面的制作者来说的确很方便。他们可以利用在 Office 97 中强大的文档处理功能,然后选择另存为 HTML 的选项即可。

1.4 基本概念

在详细介绍 HTML 语言之前,有必要把相关的一些基本概念解释清楚。

1. 主页与页面

页面(page)是 World Wide Web 中最基本的文档,用 HTML 语言编写,它包含文本、HTML 标记、以及内嵌的命令等等。这些内嵌的命令提供有关页面的结构、外观和内容的信息。

有些 HTML 标记构建了某些页面项目的结构性描述,比如说表(table)和表单(form),它们由许多元素构成。浏览器使用这些结构性描述的信息来决定怎样把页面元素相关地显示出来。比如说,描述一个表的 HTML 标记指定了表所包含的行数,以及每一行所包含的表元个数。

主页(home page)是 World Wide Web 站点中顶层的页面,通常它包括对 Web 站点的介绍,以及指向站点中(甚至站点之外)的其他页面的超链接。当用户通过浏览器访问一个 Web 服务器时,如果不指定页面,通常浏览器是以默认的方式显示出主页。在大多数的 Web 服务器中,主页一般被命名为 index.html 或者 index.htm。比如说,如果在浏览器的 Address 或者 Location 区域中输入 <http://www.microsoft.com>,浏览器所获得并显示的就是该服务器中的主页。另外,某些 Web 服务器还支持多个主页。

由此可见,主页与页面的区别是非常明显的。图 1.1 大致显示出了一个 World Wide Web 站点中主页与其他页面之间的层次性关系。

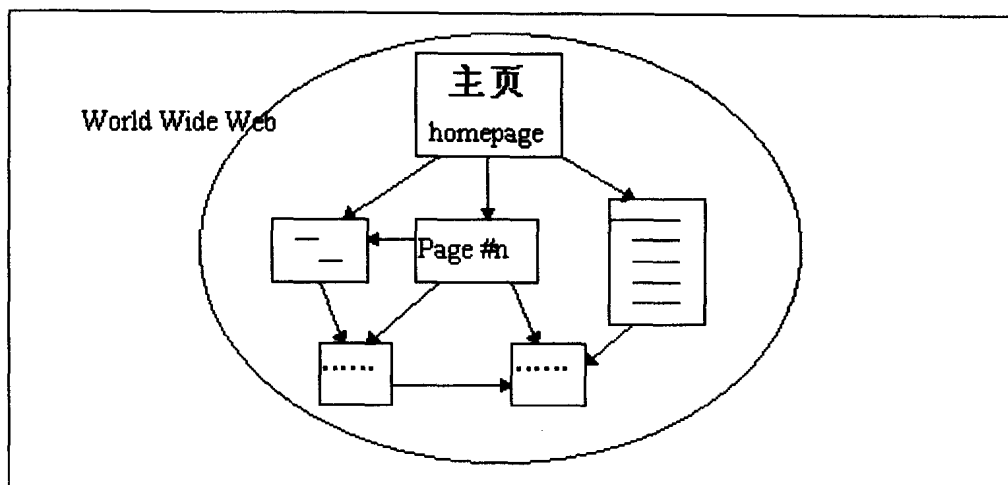


图 1.1 主页与页面

2. 超链接

World Wide Web 中页面的最重要特性是使用了超链接。超链接是从一个页面指向另一个页面或其他类型文件的链接。超链接的目的地通常是另一个 Web 页面,但也可以是一个多媒体文件甚至是一个程序。在 World Wide Web 中,超链接是在页面之间浏览或者在 Web 站点之间浏览的主要的手段。

超链接通常是嵌入到页面的文本或者图像中。当用户单击指向一个页面的超链接时,浏览器从该页面所在的服务器中获得该页面并显示出来;当用户单击指向其他类型文件的一个超链接时,浏览器从服务器上取得该文件,并且在与这种类型的文件相关联的一个应用程序中打开它。

2 HTML 页面布局

首先我们来看看 HTML 页面的基本结构及其布局设计。

2.1 HTML 的基本结构

HTML 文件基本布局由以下几个标记组成。

- `<html>...</html>`
- `<head>...</head>`
- `<body>...</body>`

HTML 文件的基本结构如下所示：

```
<HTML>
<HEAD>
    <title>, <base>, <link>, <isindex>, <meta>, <nextid>
</HEAD>
<BODY>
    HTML 文件的正文写在这里...
</BODY>
</HTML>
```

现在请打开你的计算机,启动一个文本编辑程序(MS Windows 的 Notepad 即可)来编写下面这段 HTML 文件：

```
<HTML>
<HEAD>
<TITLE>我的第一个 HTML 页面</TITLE>
</HEAD>
<BODY>
<H1>大家好</H1>
<HR>
<P>向大家问好!
</BODY>
</HTML>
```

将这个文件存成 *.htm 文件,然后用 Internet Explorer 或者 Netscape Navigator 把该文件读进来,就可以看到你的第一个 Web 页面,如图 2.1 所示。

HTML 文件都必须以 `<HTML>` 标记开头,以 `</HTML>` 标记结尾,其所有的内容都在这一对标记中被封闭起来,这样浏览器才能识别。

所有的 HTML 标记都放在“`< >`”符号里面,有些标记必须是成对使用,而后面的那个配对标记则必须加一个“`/`”符号。要注意的是:并不是每一种标记都需要有配对的标记。如果你觉

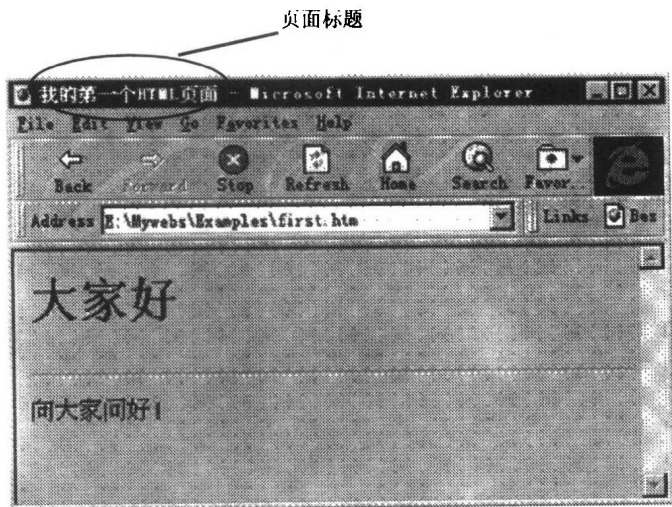


图 2.1 你的第一个 Web 页面

得你实在记不清楚哪个标记要成对使用的话,有一种简便的方法或许能帮助你记忆:如果一个标记是在一定的范围内起一定作用的话,那么它一般应该加上配对标记来标识其作用范围的结束。

再有一点,HTML 文件中的标记是没有大小写之分的。〈HTML〉和〈html〉对浏览器而言完全一样,它都能够识别。

在〈HTML〉…〈/HTML〉范围之内又分为〈HEAD〉…〈/HEAD〉及〈BODY〉…〈/BODY〉两大部分。

〈HEAD〉…〈/HEAD〉部分可以设定有关页面的一些信息,这些信息不是页面显示的主体内容,但有时对于浏览器而言是很有用的。这一部分中可以用来放置 TITLE, ISINDEX, BASE, NEXTID, LINK, META 等六种标记。刚开始时,你只要记在〈TITLE〉…〈/TITLE〉中加入页面的标题就可以了。标题显示在浏览器窗口的标题栏中,如图 2.1 所示。

〈BODY〉…〈/BODY〉标记之间包括页面的主体内容,所有将在页面中显示的内容一般都要放在这对标记之间。但是这并不是绝对的,在介绍 Frame 的时候你会认识到,它是一个例外。

以下介绍编写 HTML 页面经常要使用的 HTML 标记以及它们的属性。首先明确一点,所谓属性,是用来提供有关 HTML 标记的附加的信息,不同的标记一般具有不同的属性。属性必须与标记同时使用,其格式一般如下:

〈标记名 属性名 = 属性值 属性名 = 属性值…〉…〈/标记名〉

要注意属性名和属性值一样没有大小写之分。

一般为了使 HTML 源文件具有较好的可读性,可以让标记名和属性名都用大写字母,属性值用小写字母(数字),并且将属性值用引号括起来。

2.2 定制页面的属性

我们曾经提到过,在 HTML 文件中的〈HEAD〉…〈/HEAD〉部分可以设定有关页面的一些信息,这些信息不是页面显示的主体内容,但有时对于浏览器而言是很有用的。这一部分中可

以用来放置 TITLE, ISINDEX, BASE, NEXTID, LINK, META 等六种标记。下面就来详细介绍 META 标记, 以及如何加入该标记。

1. <META> 标记

该标记用来提供与文档相关的其他信息, 其中包括索引和目录信息, 还可以用它来做动态的页面。该标记必须在 <HEAD>...</HEAD> 标记之间, 它的一般格式为:

```
<META
HTTP-EQUIV = response
CONTENT = description
NAME = description
URL = url>
```

其中各属性的说明如下:

- HTTP-EQUIV 属性绑定该元素到一个 HTTP 响应页头, 该信息被读这个页头的应用程序所使用;

- CONTENT 属性定义关联到给定的名字或 HTTP 响应页头的元信息的内容, 该内容可以被 URL 属性所使用, 也可以是重新调入文档的指定时间间隔;

- NAME 属性指定文档的一个说明;

- URL 属性指定文档的 URL。

下面我们看几个例子。

例 2.1 如果一个文档包含如下的标记:

```
<META HTTP-EQUIV = "Expires"      CONTENT = "Tue, 04 Dec 1996 21:29:02 GMT">
<META HTTP-EQUIV = "Keywords"     CONTENT = "HTML, Reference">
<META HTTP-EQUIV = "Reply-to"     CONTENT = "anybody@microsoft.com">
```

那么服务器将包含以下的页头区域的信息, 作为对文档的 GET 或者 POST 请求的响应的 HTTP 页头的一部分:

```
Expires: Tue, 04 Dec 1996 21:29:02 GMT
Keywords: HTML, Reference
Reply-to: anybody@microsoft.com
```

例 2.2 <META> 标记可以用来制作动态的 HTML 页面, 如:

```
<HTML>
<HEAD>
<META HTTP-EQUIV = "REFRESH" CONTENT = 2>
<TITLE>动态的文档</TITLE>
</HEAD>
<BODY>
.....
<P> 该页面每两秒钟刷新一次。
</BODY>
</HTML>
```

本来 <HEAD>...</HEAD> 里面的标记并不会引起浏览器的任何反应, 因此必须用 HTTP