

北京大学信息技术系列教材

BEIJING DAXUE XINXI JISHU XILIE JIAOCAI



本书附光盘

从HTML到XML

■ 主编 蔡翠平

■ 编著 周宏滔 任吉治



北方交通大学出版社

<http://www.press.njtu.edu.cn>



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



2HT-43

178
TP 312 HT-43
C13
北京大学信息技术系列教材

从HTML到XML

主编 蔡翠平

编著 周宏滔 任吉治

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载,
也可到视听部复制

北方交通大学出版社

Northern Jiaotong University Press

清华大学出版社

Tsinghua University Press

北京·BEIJING

目 录

第 1 章 HTML 简介	(1)
1.1 网络基础知识	(1)
1.2 HTML 基础	(5)
1.3 HTML 语言的基本构成	(7)
1.4 编写 HTML 代码的工具	(10)
习题	(11)
第 2 章 HTML 的基本标志	(12)
2.1 HTML 文档的结构	(12)
2.1.1 HTML 版本信息——<!DOCTYPE...>	(12)
2.1.2 <HTML> 标记——<HTML>...</HTML>	(12)
2.2 HTML 中的颜色设置	(16)
习题	(16)
第 3 章 常用标记的使用	(17)
3.1 添加网页标题和常用文字标记的使用	(18)
3.2 水平分割线的加入和常用文字布局标记	(27)
3.3 添加表格及其标记的使用	(31)
3.4 添加列表及其标记的使用	(37)
3.5 在网页中插入图像和 HTML 中的图像应用	(41)
3.6 添加超链接和网页中超链接的应用	(46)
习题	(51)
第 4 章 框架	(53)
4.1 框架定义及实例	(53)
4.2 框架的常用标记及其属性	(57)
4.3 框架的高级设置	(58)
习题	(65)
第 5 章 表单	(66)
5.1 表单的基本概念	(66)
5.2 表单的建立	(67)
习题	(80)
第 6 章 XML 概述	(81)
6.1 XML 的起源	(81)
6.2 什么是 XML	(82)
6.3 支持 XML 的公司和它们的开发工具	(84)
6.4 XML 的主要特性	(85)
6.5 XML 的其他应用	(87)

习题	(89)
第 7 章 XML 文档简介	(90)
7.1 Hello XML	(90)
7.2 简单分析 XML 文档	(91)
7.3 XML 标记的意义	(92)
7.4 XML 文档的样式表	(92)
7.5 将样式表附加到 XML 文档上	(93)
7.6 XML 的逻辑结构	(94)
习题	(96)
第 8 章 XML 的文件规则	(97)
8.1 独立的 XML 文档	(97)
8.2 元素和字符数据	(98)
8.3 Well-Formed 的 XML 文档	(103)
习题	(107)
第 9 章 XML 的实体引用	(108)
9.1 什么是实体	(108)
9.2 内部通用实体	(109)
9.3 外部通用实体	(111)
习题	(113)
第 10 章 文档类型定义	(114)
10.1 文档类型定义	(114)
10.2 文档类型声明	(115)
10.3 Valid 的 XML 文件	(116)
10.4 使用 DTD 的优缺点	(118)
10.5 元素 (Element) 声明	(118)
10.6 DTD 中的注释	(125)
10.7 在文档间共享通用的 DTD	(126)
10.8 结合 DTD 后的实体参考	(131)
10.9 多个外部 DTD 的使用	(133)
习题	(134)
第 11 章 DTD 中的属性声明	(135)
11.1 什么是属性	(135)
11.2 在 DTD 中声明属性	(135)
11.3 声明多个属性	(136)
11.4 指定属性的默认值	(137)
11.5 属性类型	(138)
11.6 预定义属性	(143)
习题	(144)
第 12 章 级联样式表 (CSS)	(145)

12.1 样式表 (Style Sheet)	(145)
12.2 级联样式表	(146)
12.3 CSS 在 HTML 中的应用	(146)
12.4 CSS 在 XML 中的应用	(149)
12.5 选择元素	(151)
12.6 在 CSS 样式表中添加注释	(157)
12.7 CSS 中的单位	(158)
12.8 CSS 中的常用属性	(159)
习题	(166)
第 13 章 XSL	(167)
13.1 XSL 的基本知识	(167)
13.2 使用 XSL 的基本步骤和简单示例	(168)
13.3 XSL 与 HTML 的合作	(171)
13.4 XSL 中常用的元素及其属性	(173)
13.5 CSS 与 XSL	(186)
习题	(187)
第 14 章 XML 的相关技术	(188)
14.1 命名空间 (Namespaces)	(188)
14.2 XLink 和 XPointer	(190)
14.3 XHTML	(195)
14.4 其他技术	(196)
习题	(199)
附录 1 HTML 4.0 简明参考手册	(200)
附录 2 CSS 参考手册	(218)

第 1 章 HTML 简介

本章要点:

- ☒ 网络基础知识
 - ☒ HTML的基础知识
 - ☒ HTML语言的结构
 - ☒ 编辑HTML文档的常用工具
-

1.1 网络基础知识

1. WWW概述

20世纪40年代以来人们就梦想能拥有一个世界性的信息库。在这个数据库中数据不仅被全球的人们存取,而且应该能轻松地链接其他地方的信息,以使用户可以方便快捷地获得重要的信息。

随着科学技术的迅猛发展,人们的这个梦想已经变成了现实。目前正在使用的最流行的系统叫“World Wide Web”,中文可译为万维网,缩写为WWW,或W³,W3。它是在全世界范围内所有位于HTTP服务器上相互链接的超文本文档。简而言之,WWW是一个以Internet为基础的计算机网络,它允许用户在一台计算机上通过Internet存取另一台计算机上的信息。从技术角度上说,万维网是Internet上那些支持超文本传输协议HTTP(Hyper Text Transport Protocol)的客户机与服务器的集合,透过它可以存取世界各地的超媒体文件,内容包括文字、图形、声音、动画、资料库以及各式各样的软件。

Web首先是在瑞士的欧洲粒子物理实验室(CERN)研究中心开发出来的。其目的是想给CERN的物理学家提供一种共享他们的工作和利用集团信息的一种工具。在万维网之前就有许多超文本系统的执行程序。而Berners-Lee在CERN与其他人合作,制定了基于Internet的体系结构,它是开放的、公开的标准规范并附免费的样本执行程序。因为是公开的,所以都可以建立Web客户机或Web服务器;因为是免费的,所以一些开发人员能够选择编制或定义系统的每一部分。也正因为这两个因素促使其他人员都可加入这一项目。WWW的发展使得全世界范围内的人都可协同工作。

理论上说来,万维网包含所有的Web站点、FTP档案库、Telnet公共存取账号、News新闻讨论区等等。所以万维网可以说是当今全世界最大的电子资料世界,已经可以把World Wide Web当成是Internet的同义词了。

位于WWW上的文档称为页面或Web页面,它用HTML语言编写,并使用URL进行标识,URL指明了特定的计算机和路径名,用户通过它可对文件进行访问,并在HTTP协议下将文件在节点间进行传输,直至传输到最终用户。

在WWW世界中,它的资源可以互相链接,全世界目前大概有数万个Web站点,每个Web

站点都可以通过超链接(Hyper Link)与其他Web站点链接,任何人都可以设计自己的主页(Homepage),放到Web站点,然后在主页上面产生链接,与其他人的主页链接,或是连到其他的Web站点。别人也一样可以连到你的主页,或是你的Web站点,整个信息网就这样编织起来了,形成一个巨大的环球信息网。本节将简要介绍一下万维网的一些概念:浏览器、服务器、协议等。

2. 浏览器与服务器

先撇开互联网络的技术问题,来思考一下谁是网络中最重要的参与者。无疑,他们是“网页提供者”与“上网者”。网页提供者将其制作好的网页放在服务器上,而上网者则是准备好上网的机器,然后通过互联网来浏览网页制作者所提供的网页。因此,把放置网页的部分称为服务器(Server),而上网浏览的部分则称为浏览器(Browser)。

浏览器是一个需要某些东西的程序,而服务器则是提供某些东西的程序。也就是说,浏览器是请求资源的,而服务器是提供资源的。浏览器是一种客户端应用程序,它允许用户查看位于WWW、另一网络以及用户计算机上的HTML文档;允许用户沿着文档中的超链接进行浏览或传输文件。服务器是对浏览器的请求做出反应的计算机或程序。这与通常所说的“客户”与“服务”的概念之间有一定的相似性。这就像商家与消费者之间的关系,商家提供资源,提供服务;消费者享受(或接受)资源,享受(或接受)服务。消费者可以去不同的商家,商家也给许多不同的消费者提供服务。一个浏览器可以向许多不同的服务器请求,一个服务器也可以向多个不同的浏览器提供服务。通常情况下,一个浏览器启动与某个服务器的对话,也就是请求服务。服务器通常是等待浏览器请求的一个自动程序。浏览器通常是作为某个用户请求或类似于用户的某个程序提出的请求而运行的。浏览器的作用就是把从服务器传回的超媒体信息展现在用户面前,它知道如何去解释和显示在WWW上找到的Hypertext(用HTML语言编写),HTML语言本身包含了各种格式化超文本的方法,从而允许浏览器根据它格式化每一种文本类型,以获得WWW页面(Web Page或HomePage)设计者当初设计时的屏幕显示效果。此外,大多数浏览器都可以自动调用其他应用程序(Helper Applications Program),以显示特殊类型的文档,如audio或者video的文件格式。协议是浏览器请求服务器和服务器如何应答请求的各种方法的定义,它就好像消费者与商家之间的交易规则,定义了如何请求服务与如何提供服务。WWW中最常用的协议就是HTTP协议,它负责管理超文本转换协议,HTTP是一个客户机/服务器协议,通过HTML接收和传送请求,它是支持WWW上信息交换的Internet标准,是定义Web服务器如何响应文件请求的Internet协议。

浏览器(Browsers)是为了使用Web,需要一个Web客户端程序,这一程序能够解释并显示超文本文件,它知道如何找到并显示由链接指向的文件。第一个WWW浏览器,是一种文本行式浏览器,它是由CERN中的小组完成的。Mosaic,是由伊利诺斯大学的NCSA开发的浏览器,由马克·安德森和艾瑞克·比拉开发的。它的出现对Web的增长起到了巨大的推动作用。Mosaic第一个版本是基于UNIX系统下的X Windows上开发的。随着Internet用户群体的增加,NCSA扩展了Mosaic的开发成果,开发出了基于Microsoft Windows和Macintosh版本的浏览器。NCSA还允许为商业软件开发商发放Mosaic源代码。结果,从Mosaic演变并出现了许多商业WWW浏览器,包括微软的Internet Explorer。

常用的浏览器主要包括Netscape Navigator、Internet Explorer(也就是通常所说的IE)等。

在Web中,浏览器的任务是:

- ☞ 帮助制作一个请求(通常在单击某个链接点时启动)。
- ☞ 将请求发送给某个服务器。
- ☞ 呈交HTML文档和传递各种文件给相应的“观察器”(Viewer),把请求所得的结果进行报告。

一个观察器是一个可被WWW客户机调用而呈现特定类型文件的程序。当一个声音文件被某个浏览器查阅并下载时,它只能用某些程序(例如Windows下的“媒体播放器”)来“观察”。

网络浏览器为网络服务器提供了一个图片似地、以文本为基础的终端界面。这种终端方法在用户与网络服务器间提供了一个界面,网络浏览器负责把网络服务器送来的HTML在浏览器内转换成图形用户界面。

通常浏览器不仅可以向Web服务器发出请求,还可以向其他服务器(例如FTP、邮件服务器)发出请求。

网络服务器有一些责任,即所有的中心必须围绕把HTML传送到请求的客户浏览器上。一个Web服务器的任务是:

- ☞ 接受请求。
- ☞ 请求的合法性检查,包括安全性屏蔽。
- ☞ 针对请求获取并制作数据,包括Java脚本和程序、CGI脚本和程序、为文件设置适当的MIME类型来对数据进行前期处理和后期处理。
- ☞ 把信息发送给提出请求的浏览器。

3. 统一资源定位符(URL)

World Wide Web是一个信息资源的网络。Web依靠3种结构来使这些资源为各类用户做好准备:

- ☞ 单纯的命名方案——提供在Web上进入资源的统一的方法和路径(Uniform Resource Locator, URL)。
- ☞ 协议——允许在Web上交换已命名的资源(HTTP、FTP)。
- ☞ 超文本——供在资源之间易于引导(HTML)。

在Web上的任何可用资源——HTML文档、图像、视频、程序等等——都有一个地址可被统一资源定位符解码。“资源”(Resource)是指在网络上所能获取的文字、图像、声音、动画等资料的统称。这些资料实际上都是以各种不同格式的“文件”(File)类型存在,分散于各地的电脑主机中;而“定位符”(Locator)的目的就是要指出这些资料的所在处。

统一资源定位符是互联网上资源的地址。Web浏览器通过使用统一资源定位符(URL)来对互联网上的资源进行定位。在单机系统中,定位一个文件需要路径和文件名,对于遍布全球的Internet,显然还需要知道文件存放在哪个网络的哪台主机中才行。URL与单机系统不同的是:在单机系统中,所有的文件都由统一的操作系统管理,因而不必给出访问该文件的方法;而在Internet上,各个网络,各台主机的操作系统都不一样,因此必须指定访问该文件的方法。一个URL包括了以上所有的信息。URL包含了3个部分:在Web上传输资源所使用的协议名称、数据所在的主机名称、资源本身的路径。它的构成为:

protocol:// machine-name[:port] / directory / filename

其中, protocol是访问该资源所采用的协议, 即访问该资源的方法, 下面显示了几种协议, 它可以是:

- ☞ HTTP: 超文本传输协议, 该资源是HTML文件。最常用的是HTTP, 主要用来连接远端的WWW服务器。例如: `http://www.pku.edu.cn`, 即可连接到北京大学的WWW服务器。
- ☞ FTP: 文件传输协议, 用FTP协议访问该资源, 主要是用来取得FTP服务器上的文件资源。例如: `ftp://ftp.pku.edu.cn`, 即可显示北京大学的FTP服务器内容。
- ☞ Gopher: Gopher协议, 该资源是Gopher文件, 表明该资源是网络新闻。

machine-name是存放该资源主机的IP地址, 通常以字符形式出现, 如`www.pku.edu.cn`。

port, 端口号, 是服务器在该主机所使用的端口号。一般情况下端口号不需要指定。只有当服务器所使用的端口号不是默认的端口号时才指定。

directory和filename是该资源的路径和文件名。

一个典型的URL为`http://www.pku.edu.cn`, 它表示北京大学主页。再考虑这样一个网页, 它是北京大学的学校概况, 其URL为: `http://www.pku.edu.cn/about/about.htm`。这个URL可以这样阅读:使用HTTP协议并且通过`www.pku.edu.cn`的机器传输, 而文件是`/about/about.htm`。

URL还可能携带更多的信息, 我们再来看一个URL地址, 它的形式如下所示:
`http://andyzu:dream@www.dreamworks.com:81/network/intro.html#faq`, 其中, “`http://`”表明这个地址中的数据是HTTP协议的, 需用WWW方式访问; “`www.dreamworks.com.cn`”是这个地址, 其中最后的“`cn`”是国家标识, 二级域名“`com`”代表公司的含义, “`andyzu:dream`”表明这个地址需要用户名和密码登录才能访问, 而完全公开的网址无需使用用户名和密码; “`:81`”说明这个服务不在WWW服务的缺少端口80, 而是将服务改在81端口上。“`network`”是子目录名, “`intro.htm`”是文件名。WWW服务器一般会指定一个默认文件名, 当访问一个目录并且没有明确给出文件名的时候, 服务器会自动帮助用户调出默认文件, 一般是`index.htm`、`default.htm`等。WWW服务器上的起始html文件(文件具体存放的路径及文件名)取决于该WWW服务器的配置情况。当一个HTML文本比较长的时候, 还可以使用“锚”, 如上面的“`#faq`”, 这样可以快速地定位到这个文本的任何一部分。对于使用CGI程序或ASP程序的URL, 在文件名后面可以跟随任意形式的参数, 如:

“`http://www.pku.edu.cn/try.asp?user=andyzu&city=beijing`”等形式。

与单机系统绝对路径、相对路径的概念类似, 统一资源定位符也有绝对URL和相对URL之分。上文所述的是绝对URL。相对URL是相对于用户最近访问的URL。比如一个用户正在观看一个URL为`http://www.pku.edu.cn/index.html`的文件, 如果想看同一个目录下的另一个文件`intro.html`, 可以直接使用`intro.html`, 这时`intro.html`就是一个相对URL, 它的绝对URL为`http://www.pku.edu.cn/intro.html`。

4. 超文本

超文本文件是与其他数据有关联(links)的数据。包含与其他文档链接的文档; 选择链接时自动显示第二个文档。超文本文件的一个简单例子是大百科全书。假设用户正在读“树”这个条目, 在文章的最末有一个参考这样写到, “相关信息参见‘植物’”。这最末一行就是一

个关联,从“树”到“植物”这个条目。当然这只是一个简单例子,全球网是基于一个远远复杂得多的超文本文件。特别是在文件的任何地方都可能有关联,不仅是在末尾。举一个想像的例子。假设正在用全球网阅读“树”的超文本文件,每次提到一种新的“树”就有一个关联,每个关联都以某种方式标识起来以显得突出。例如,有关联的字可能被加亮或者加了下划线,或者标注了一个数字。如果跟着关联走,将会转到那种类型的“树”的条目去。在主条目里还有其他题目的关联,如“雨林”或“木”。这些关联又引出完整的条目。用户或许还能找到与技术术语的关联,如“落叶科”和“针叶树”。如果沿着这些关联就会找到其定义。在全球网的语言里,一个超文本文件的文件就是由一些数据组成,而这些数据又可能与其他文件相关联。

5. WWW上常用的文件格式

用户在网上查询信息时,经常会遇到不同的文件格式。这些文件中,有一些格式可以直接使用浏览器进行处理,还有一些需要另外的软件来进行处理。

(1) .html或.htm

它是Internet中超文本文档格式文件,在浏览器中可直接浏览。

(2) .au

它是Internet中常见的声音文件格式。

(3) .jpeg或.jpg

它是Internet中常见的图像文件格式,可在浏览器中直接浏览。

(4) .gif

它是Internet中常见的图像文件格式,可在浏览器中直接浏览。

(5) .mov

它是Internet中常见的影视文件格式,可用Quicktime或Realplay等软件播放这类软件。

(6) .gz或.z

它是UNIX中一种常见压缩文件格式,可使用Winzip对这类软件进行解压缩。

(7) .txt

它是一种常见的文本文件,可在浏览器中直接进行浏览。

1.2 HTML 基础

World Wide Web(简称WWW)是由无数的商业、教育和娱乐资料充斥着的日益庞大的信息空间网。这些可以通过Internet访问的以超媒体文档形式构成的,资料可以放置在这个世界上任何地方,但是,无论这些资料怎样产生,绝大多数Web文档都是由超文本标记语言(HTML)编写的。在WWW的世界中,HTML(超文本标记语言,Hyper Text Markup Language)扮演着一个不可或缺的角色。HTML是构成网页最“基础”的要素。

HTML是在1989年由Timothy Berners-Lee所制定出来的。随后,由NCSA推出Mosaic浏览器,将HTML语言推广并得到越来越多用户的欢迎。

HTML是一种描述性语言,是由W3C组织(World Wide Web Consortium)推出的。它是一个按SGML(标准通用标记语言,Standard Generalized Markup Language)定义的语言,和其他

标识语言一样,采用作标记定义文本的特殊格式。标记是对文本的一段进行语义标记。在浏览器解释HTML时,标记本身并不呈现在浏览器上,浏览器按标记的意义来显示被标识的部分。HTML用标记来定义文档中的文本及图像等元素,来指示网络浏览器如何显示这些元素,以及如何响应用户的行为(如通过按键或单击鼠标来激活一个链接)。用HTML可以创建那些能在互联网上传输的信息页——即通常所称的主页(Homepage)或网页。SGML可以通用在各类应用领域的文件上,它允许使用者根据数据结构与形态的需求,制定出实用的文件格式定义(Document Type Definition, 简称为DTD),所以不同领域中的文件,其包含的数据项目与形态可能差异性很大,但只要分别定义出各自的DTD,就可以被各类文件建立时所引用与遵循,以保证同一类的文件都有相似的文件结构。网页本身也就是一种文件,也是互联网中的一种应用领域,所以有关网页的文件结构,当然也可以采用SGML来定义网页适用的DTD。这就是HTML的由来,也是SGML的一种应用,主要是采用SGML的规范来制定网页适用的DTD,而这种特别为网页量身定做而产生的标记语言就称为HTML(Hyper Text Markup Language),中文可译为超文本标记语言。在本书的后面讲到XML时,将专门讲述DTD的定义与撰写方式,读者就可以体会到上面这段话的含义了。

➡ 说明:

- ① SGML英文全名为Standard Generalized Markup Language,中文可以为标准通用标记语言。SGML最主要的目的是在提供一种描述电子文件的规范,也就是提供一种对文件进行结构化的法则,而当文件采用这种法则来进行结构化的处理后,该SGML文件就可以被广泛地传递与使用,且有关该文件的制作、存取、应用都可以通过计算机来做最有效的处理。
- ② W3C英文全名是World Wide Web Consortium,中文可以译为全球信息网协会,该组织主要负责有关全球信息网应用各类标准的制定,像HTTP、HTML、XML等都是该协会制定出来的。且该协会主要是由对全球信息网有兴趣的公司所组成的,成员包括Microsoft、HP、IBM、AT&T、Netscape等,所以一般通过该组织推荐的标准都能受到各家厂商的支持。

HTML语言使用描述性标记符(称为标记Tag)来指明文本的不同内容。标记是区分文本各个组成部分的分界符,用来把HTML文本划分成不同的逻辑部分(或结构),如段落、标题和表格等。标记描述了文本的结构,它向浏览器提供该文本的格式化信息,以传送文本的外观特征。用HTML语言编写的页面是普通的文本文件(ASCII文件),不含任何与操作系统和计算机硬件相关的信息,所以HTML文件可以被任何文本编辑器读取。

HTML是一种语言,但并不算是“程序”语言。HTML所定义的范畴仅局限于如何表现文字、图片,以及如何建立文件之间的链接。而程序则是经过规划的一连串命令(或称为“语句”),而这样的命令可用来驱动操作系统或应用程序执行某些操作。

HTML文本包含两种信息:页面本身的文本和表示页面元素、结构、格式、及其他超文本链接的HTML标记。HTML标记规定了HTML文本的逻辑结构,并且控制其显示格式,也就是说,设计者可以用标记定义HTML文本的逻辑结构,但是文本的实际显示则由浏览器来负责解释。可以使用HTML标记来设置链接、标题、段落、列表和字符加亮区域等等。

大部分HTML标记是以“<标记名>相应内容</标记名>”形式出现的,标记的名字用尖括号括起来。HTML标记一般有起始标记与结束标记两种,分别放在它起作用的文本两边。

起始标记与结束标记极相似，只是结束标记在“<”号后面多了一个斜杠“/”。后面将会看到，某些HTML元素只有起始标记而没有相应的结束标记，例如换行标记
，由于换行不包括相应的内容，所以只使用一个标记就可以了。还有一些元素的结束标记是可以省略的，如列表项结束标记“”、段落标记“</P>”和表格行结束标记“</TR>”等等。标记名不区分大小写。

起始标记中可以包含属性(Attribute)，其位置是从标记名之后空一格的地方开始，在结束符“>”之前结束，如 文字。属性向客户端提供了关于页面元素内容以及如何处理的附加信息，用户可以在这个区域中对文本的一些具体属性如大小、颜色、字体等信息进行设置。

1.3 HTML语言的基本构成

本节先以实际的例子来示范HTML语言的基本结构，下面的实例1-1就是一份HTML文件，该HTML文件经过浏览器解读以后，即可显示出如图1-1的网页内容。

例1-1:

```
<HTML>
<HEAD><TITLE>个人资料</TITLE></HEAD>
<BODY>
  <H3>姓名: 张华</H3>
  <H3>生日: 1978年10月4日</H3>
  <H3>职业: 国家公务员</H3>
  <H3>电子邮箱: lily@263.net</H3>
</BODY>
</HTML>
```

显示结果:

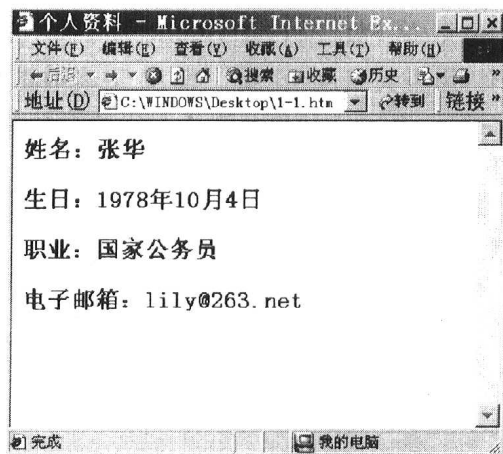


图1-1 例1-1的显示结果

观察说明

在上面的例子中可以发现<HTML>、<HEAD>、<BODY>、<H3>等等，这些就是在1.2节中所说的标记(Tag)，它们是HTML的控制标记，也可简称为标记。HTML文件是标准的ASCII文件，HTML语言即超文本标记语言，它看起来像是加入了许多标记(Tag)的特殊字符串的普遍文本文件。从结构上讲，HTML文件由元素(Element)组成，组成HTML文件的元素有许多种，用于组织文件的内容和指导文件的输出格式。绝大多数元素是“容器”，即它有起始标记(Start Tag)和结束标记(End Tag)，在起始标记和结束标记中间的部分是元素体。每一个元素都有名称和可选择的属性，元素的名称和属性都在起始标记内标明。

比如体元素(BODY)：

```
<BODY BACKGROUND= "background.gif" >
```

```
<H2>First</H2>
```

```
<P>这是我的第一个HTML文档。</P>
```

```
</BODY>
```

第一行是体元素的起始标记，它标明体元素从此开始。因为所有的标记都具有相同的结构，所以这里将仔细分析该标记的各个部分，以便读者对标记的写法有大概的了解。起始标记开始是元素名称，由于元素和标记一一对应，所以元素名也叫标记名。需要注意的是“<”和“body”之间不能有空格。元素名称不区分大小写。

“BACKGROUND”是属性名。一个元素可以有多个属性，属性及其属性值不区分大小写。本属性指明用什么方法来填充背景。等号右边引号内的部分指明了属性值，“background.gif”即属性值，表示用background.gif文件来填充背景。“属性名=属性值”合起来构成一个完整的属性，一个元素可以有多个属性，各个属性用空格分开。

“>”标志起始标记结束。

第二行和第三行是body元素的元素体，最后一行是body元素的结尾标记。结尾标记用“</”开始，随后是元素名，然后是大于号“>”。

从上面的例子中可以看出，一个元素的元素体中可以有另外的元素。(上面例子中第二行的标题元素<H2>...</H2>和第三行的分段元素<P>)。实际上，HTML文件仅由一个<HTML>元素组成，即文件以<HTML>开始，以</HTML>结尾，文件其余部分都是<HTML>的元素体。<HTML>元素的元素体由两大部分组成，即头元素<HEAD>...</HEAD>和体元素<BODY>...</BODY>和一些注释。头元素和体元素的元素体又由其他的元素和文本及注释组成。也就是说，一个HTML文件应具有下面的结构：

```
<HTML>  ←——HTML文件开始
  <HEAD> ←——文件头开始
    <!--文件头-->
  </HEAD> ←——文件头结束
  <BODY> ←——文件体开始
    <!--文件体-->
  </BODY> ←——文件体结束
</HTML> ←——HTML文件结束
```

说明:

需要说明的是, HTML是一门发展很快的语言, 早期的HTML文件并没有如此严格的结构, 因而现在流行的浏览器(如IE, Netscape等)为保持对早期HTML文件的兼容性, 也支持不按上述结构编写的HTML文件。各种浏览器对HTML元素及其属性的解释也不完全一样, 本书中所介绍的元素, 元素的属性及其输出是以IE 5.0浏览器为准的。

一般来说, HTML的元素有下列3种表示方法:

☞ <元素名>文件或超文本</元素名>

☞ <元素名 属性名=属性值>文本或超文本</元素名>

☞ <元素名>

第三种写法仅用于一些特殊的元素, 比如分段元素“P”, 它仅是通知WWW浏览器在此处分段, 因而不需要界定作用范围。

● 实际上, 分段元素“P”也具有结束标记“</P>”, 只是由于WWW浏览器具有很高的容错性, 因此即使省略了它的结束标记, WWW浏览器仍然能够正确显示。但是这对一个规范的文档来说, 是不允许的。在以后将要学到的XML语言中, 它的起始标记和结束标记是一一对应的, 是绝对不允许省略的。

HTML文件中, 有些元素只能出现在头元素中, 绝大多数元素只能出现在体元素中。在头元素中的元素表示的是该HTML文件的一般信息, 比如文件名称, 是否可检索等等。这些元素书写的次序是无关紧要的, 它只表明该HTML文件有还是没有该属性。与此相反, 出现在体元素中的元素是次序敏感的, 改变元素在HTML文件中的次序会改变该HTML文件的输出形式。

技巧

(1) 如何查看Web页的HTML源文件

用浏览器(假设为Microsoft Internet Explorer(即IE))随便打开一个网页, 从屏幕上看到的是精心编制的生动的页面, 而不是令人眼花的枯燥的HTML代码。那么怎样才能看到隐藏在其中的HTML源代码呢? 其实很简单, 用鼠标单击“查看”菜单的倒数第二个选项“源文件”, 执行后便会自动弹出一个记事本, 上面写的就是该Web页的HTML源代码, 如图1-2所示。

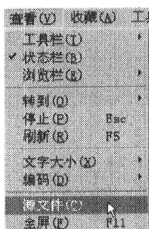


图1-2 查看HTML源代码

(2) 怎样用HTML源代码编写一个最简单的网页?

首先打开记事本, 建立一个空白文件。其方法有两种: 一种方法是在Windows中通过“开始→程序→附件→记事本”来打开; 另一种方法是打开其他网页的源文件, 再在该记事本中通过“文件→新建”来打开。

然后在记事本上写下如下代码:

```
<HTML>
```

```
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

接下来,在“<TITLE>”与“</TITLE>”之间写上需要编写的网页的标题,再在“<BODY>”与“</BODY>”之间写上一些文字,这样一个非常简单的源文件就做好了。但现在它还不是网页的形式,因为这个源文件属于文本文件,具有.txt的扩展名,要想使它成为可浏览的网页,需要将其扩展名改为.html或.htm。因此,需选择“文件→保存(或另存为)”,将文件以.html或.htm为扩展名进行保存即可。

最后,就可看到打开刚才所保存的文件(在找到它时,就可发现其图标已是Web文件的标志),就可看到编写的第一个最简单的网页。

若仔细看这个Web页,就可发现浏览器最上方的标题栏所列出的在“<TITLE>”与“</TITLE>”之间写的网页标题,而Web页的页面上正是在“<BODY>”与“</BODY>”之间所写的文字。

1.4 编写HTML代码的工具

目前,可以用来编辑HTML文档的软件很多,可以分为以下3种。

(1) 一般文本编辑器

其实HTML程序本身是一个文本文件的格式,用户可以使用任何的文本编辑器,像微软所附带的记事本(Notepad)、写字板;还有Word、WPS、UltraEdit、EditPlus等。图1-3就是UltraEdit的工作界面。但要注意的是在存盘时,必须选择文本格式(也就是TXT格式)来存盘,然后再把扩展名改成.htm或.html。

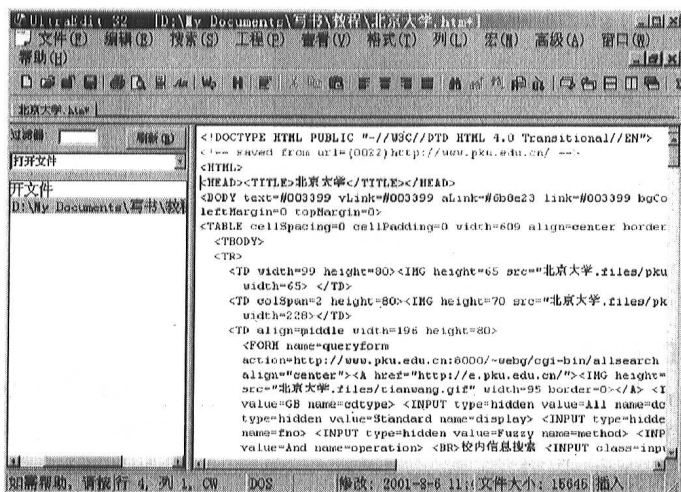


图1-3 UltraEdit的工作界面

(2) 专门撰写HTML的编辑器

目前市面上有许多专门用来编写HTML的编辑器，如HotDog、1st Page、40tude HTML等都是很好的HTML编辑器，这类专门编辑HTML文件的软件，有一个共同的好处就是无需设计者背记控制标记，只需单击一下软件中的控制标记按钮就可以将其显示出来。

(3) 自动产生HTML文件的设计软件

这类设计软件是采用一种WYSIWYG(What You See Is What You Get)的设计模式，设计者直接将文字与图像置入，并在屏幕上调整排版，存盘后就会自动转成HTML文件，设计者使用这种软件可以完全不认识HTML的控制标记，这类软件有FrontPage、Dreamweaver等。

如果用户是进行网页的设计，完全可以采用第三类软件，以加快开发速度。如果是想学习编写HTML语言的话，那就最好采用前两种软件，这样才有助于用户更好地学习。

习题

1. 简述Web中浏览器与服务器的基本任务。
2. 分析URL为ftp://test:done@ftp.lib.pku.edu.cn:21/user/course.txt各部分的含义。
3. 简述HTML文件的基本结构。

第2章 HTML的基本标志

本章要点:

- ☑ HTML文件的结构
- ☑ HTML文档里的颜色设置

2.1 HTML文档的结构

一份HTML文档通常由三部分组成：一条包含版本信息的语句，一个引导描述段以及一个含有具体内容的主干。HTML文档的主要结构可分为以下几个部分。

2.1.1 HTML 版本信息——<!DOCTYPE...>

SGML DOCTYPE结构声明了文档使用HTML的哪个版本组成。这是标识一个遵守HTML的特定版本规范的HTML文档的部分，作者应当在每一份文档的第1行包括一个如下的类似声明：<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Draft//EN">。声明的最后两个字母指HTML DTD的语言，在此例为英语("EN")。作者可以根据与它们的文档相宜的HTML的版本使用其他形式的描述。HTML 4.0建议的文档形式为：

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Draft//EN">，指出文档遵循HTML 4.0草案规格。

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Final//EN">或<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">，则指出了文档遵循HTML 4.0最后规格。

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Strict//EN">则指明文档遵守详尽的HTML 4.0规格。不过，这一部分即使没有写出也是无关紧要的。通常用HTML编写网页时，也是可以省略这个部分的。

2.1.2 <HTML>标记——<HTML>...</HTML>

这是定义HTML文档开始与结束的部分，也是HTML文档中最先出现的标识，表明这个文件的内容是用HTML语言来实现的。<HTML>和</HTML>必须成对出现，分别表示HTML文件的起始和结束。在<HTML>...</HTML>容器中又包括以下三个部分。

1. <HEAD>和</HEAD>标记

这一部分用来说明文档标题以及该页面的其他信息，<HEAD>和</HEAD>构成HTML文档的开头部分，在此标志对之间可以使用<TITLE></TITLE>、<SCRIPT></SCRIPT>等标志对，这些标志对都是描述HTML文档相关信息的标志对，<HEAD></HEAD>标志对之间的内