



前言

我们的HTML

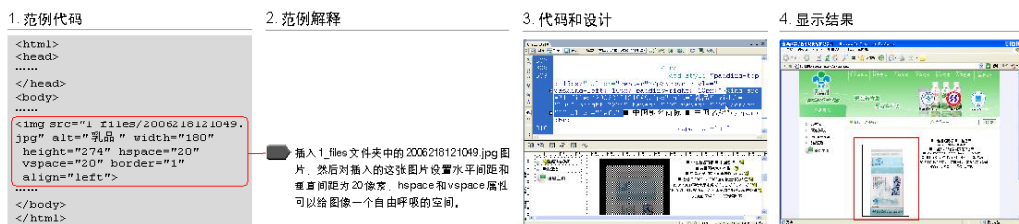
HTML 给人更多的感觉是代码，不过它与其他的程序语言有着很大的不同，它是 Hypertext Markup Language，也就是“超文本标记语言”，请大家不要忽略“标记”，它才是HTML的精华所在，也就是<>中的内容。每个标记都有它特定的含义，精彩的Web页面就是靠它们描述出来的。大多数标记成对出现，中间是属性等实质性内容，后面的标记加上一个“/”，表示结束部分。这样看起来你会觉得HTML也不是什么高难的语言，仅此“标记组合”而已。

本书弥补了以往HTML图书的不足，将真实的网页设计案例融入到培训专家的讲义中，铸造了一本真正意义上从入门到精通的专家级HTML参考书。

本书主要特色

全书包括13章及附录，每个章节都配有完整的制作实例，使读者能够在真实的学习环境中掌握HTML语言。

- 起点较低，收获丰富——真正将HTML代码编写放入Dreamweaver 8中结合讲解，读者能直接体验从入门到精通的学习过程。
- 上手容易，轻松学习——100%网页设计真实案例讲解，让学习代码像学习可视化操作一样简单、容易。
- 代码注解，时时提示——添加注释，对每段代码中的词汇进行细致的解释，避免理解歧义的发生，让读者真正领会代码的所有含义。
- 四位一体，全面讲解——范例代码、范例解释、代码和设计、显示结果四位一体的讲解过程，把代码和制作效果直接联系在一起。



- 公开代码，方便查阅——所有实例代码及效果文件放在附赠光盘中，读者可以随时学习使用、借鉴发挥。
- 配送丰富，物超所值——16进制RGB值与命名对照表、HTML元素速查表、HTML字符集识别等内容随附在光盘中。

看后可以学会

HTML 语言是任何高级网页制作技术的核心与基础，本书通过几十个真实的网站实例讲解了上百个语法知识点的应用。主要包括：

- HTML 语言的基本概念和特点，对 HTML 语言有一个初步的理解。
- 使用记事本和 Dreamweaver 编写 HTML 页面，迅速了解到网页的制作流程。
- HTML 网页的结构、头部信息、页面的整体设置。
- 通过 HTML 语言对字体、大小、颜色、底纹、边框等设置文本的属性。
- 使用 HTML 语言制作页面中的超级链接。
- 图片、音频、视频、动画、滚动字幕等多媒体效果在网页中的应用。
- 表单、表格、框架结构的使用，让网页更加完美。
- CSS 样式表、JavaScript 脚本的使用，是动静 Web 设计的最佳选择。

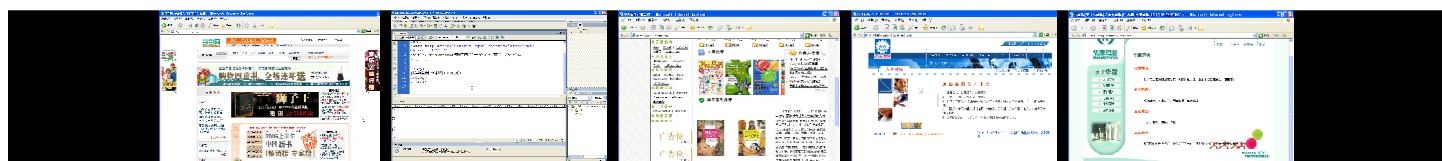
光盘附赠

1. 全书所有实例的原始文件以及完成文件。
2. 8款Dreamweaver中使用的源代码相关插件。
3. 8款优秀源代码编辑器。
4. HTML 语言电子学习手册。
5. HTML 语言授课教学幻灯片。
6. JavaScript特效代码集。
7. 网页颜色表。
8. 语法简明速查表。

作者要说

作者在3个月的写作时间里，一直坚持不懈地投入很多精力，可以看到本书在内容的原创性、体例的特殊性和版式的创新性上下了很大的功夫，目的是希望为学习网页制作的读者提供一本讲述HTML语言、CSS样式表的经典图书。从这里开始，HTML将不再是高不可攀的代码技术，而是正在变得愈加和蔼可亲起来。

作者
2007年1月



目 录 CONTENTS

Chapter 1

初识HTML语言	1
WWW 及其发展现状	2
WWW 的基本概念	2
WWW 的工作机制	2
WWW 的应用	3
WWW 的站点建设	4
HTML 简介	5
HTML 基本概念	5
HTML 的发展历史	6
HTML 4.0 的特点	6

Chapter 2

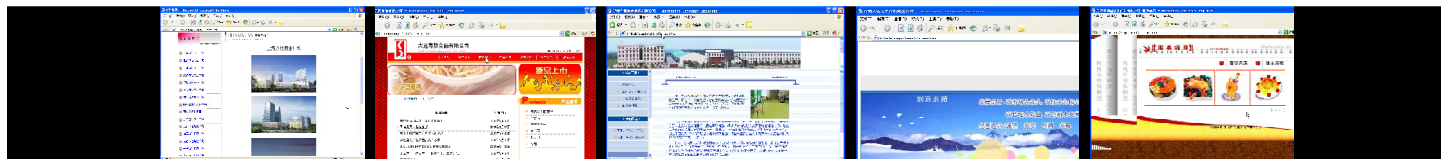
快速编写HTML文件	9
使用记事本编写HTML页面	10
使用编辑器编写HTML页面	11
HTML 文件的基本结构	14
HTML 中的标签	15
HTML 中的属性	16

Chapter 3

文档的顶级信息	19
根元素	20
Html 标签	20
Head 标签	22
Body 标签	23
头部元素	29
Title 标签	29
Base 标签	30
Basefont 标签	32
Meta 标签	33
实例制作	41

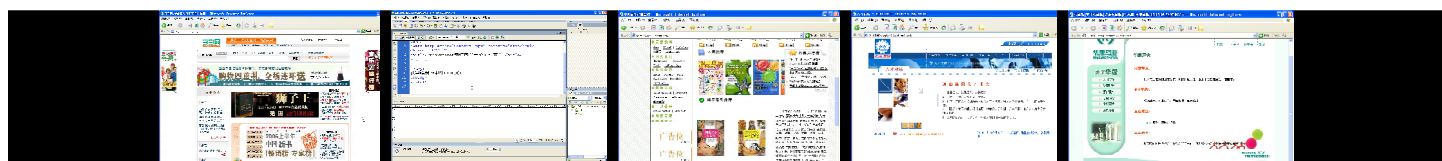
Chapter 4

文字和段落	43
文字内容	44
空格	44
特殊字符	45
注释	46



Chapter 5

显示HTML代码标签	47
文字标记	48
Font标签	48
Hn标签	51
文字的修饰	53
B和Strong标签	54
斜体字标签	55
Sup标签	56
Sub标签	57
Big标签	58
Small标签	59
U标签	59
S和Strike标签	60
Address标签	61
Code和Samp标签	63
Kbd标签	64
Tt标签	64
Var标签	65
Rt和Ruby标签	66
段落的排布	67
P标签	68
Br标签	71
Nobr标签	72
Div标签	73
Span标签	76
Center标签	77
Blockquote标签	79
实例制作	80
水平线和列表	83
水平线	84
Hr标签	84
列表	89
无序列表	89
有序列表	93
目录列表	98
定义列表	99
菜单列表	101

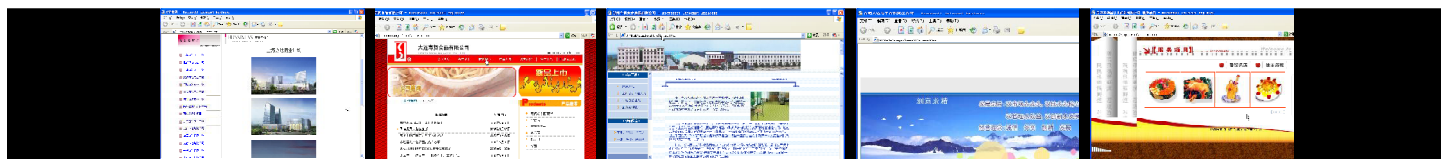


Chapter 6

Chapter 7

Chapter 8

实例制作	103
插入图片	107
网页图片的基本格式	108
GIF图像	108
JPEG 图像	109
插入图片	109
标签	109
实例制作	116
页面中的超级链接	119
关于超级链接	120
URL	120
制作超级链接	120
外部链接	121
访问News 新闻组	127
发送Email	128
内部链接	129
书签链接	130
空链接	132
脚本链接	132
制作图像映射	133
实例制作	137
多媒体效果	141
页面中的背景音乐	142
Bgsound 标签	142
插入图片	143
Embed 标签	143
嵌入Flash 动画	144
嵌入Mp3 音乐	146
嵌入Mpg 视频	146
利用Img 标签插入AVI 视频	147
插入AVI 视频	147
制作滚动字幕	150
Marquee 标签	150
其他嵌入式内容	158
使用Object 标签	158
使用Applet 标签	159

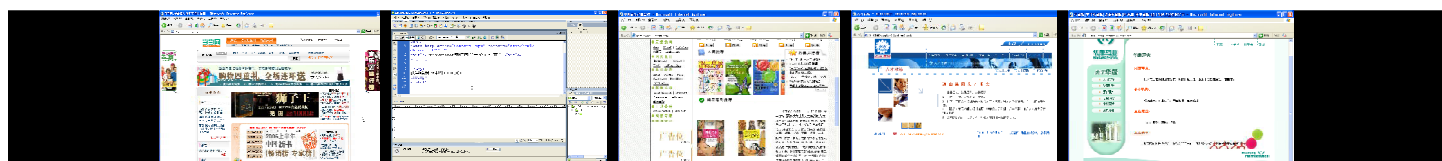


Chapter 9

Chapter 10

Chapter 11

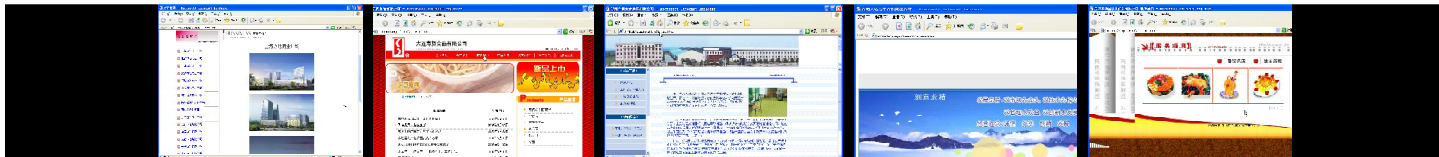
使用Param标签	160
实例制作	163
表单元素	167
什么是表单	168
Form标签	168
表单元素	171
Input标签	171
Textarea标签	184
Select标签	186
Option标签	187
Optgroup标签	190
表单元素分组	192
Label 标签	192
Fieldset 标签	194
Legend 标签	194
实例制作	197
使用表格排版网页	203
表格的整体结构	204
Table 标签	204
Tr 标签	205
Td 标签	206
Th 标签	207
Caption 标签	209
表格的属性	211
Table 标签的属性	211
Tr 标签的Valign 属性	225
Td 和 Th 标签的属性	226
Caption 标签的属性	231
高级表格标签	234
Thead 标签	234
Tbody 标签	237
Tfoot 标签	239
实例制作	241
框架结构页面	249
框架的整体结构	250
Frameset 标签	250



Chapter 12

Chapter 13

Frame标签	251
Noframes标签	252
框架的属性	255
Frame标签的属性	255
Frameset标签的属性	263
制作内联框架	266
Iframe标签	266
Iframe标签的属性	268
实例制作	275
使用CSS样式表将结构和样式分开	279
什么是样式表	280
样式表的基本语法	280
样式表类型	280
样式表语法	284
样式表的属性	289
字体属性	290
颜色和背景属性	296
文本属性	303
边框属性	311
方框属性	315
列表属性	323
滤镜属性	327
鼠标滤镜	340
实例制作	341
在页面中使用JavaScript脚本	345
什么是JavaScript脚本	346
在页面中加入JavaScript脚本	346
Script标签	346
使用内在事件调用	348
脚本链接	349
JavaScript内在事件	350
onBlur事件	351
onChange事件	352
onClick事件	352
onFocus事件	353
onLoad事件	354
onMouseOver事件	355



附录

onMouseOut 事件	356
onReset 事件	357
onSelect 事件	358
onSubmit 事件	359
onUnload 事件	360
实例制作	361
附录	365
附录1	366
附录2	367
附录3	378
附录4	381



Chapter

1

初识HTML语言

HTML（超文本标识语言）是能生成活动文档的代码系统，活动文档就是使用户和别人能够进行交互作用的文档。HTML的普通应用带来了超文本技术——从一个主题转入另一主题，而不是线性地寻找和阅读信息。如今人们不再需要逐页翻阅信息，而可以直接单击链接取得所要的信息，通过E-mail与别人联系，填写表单并联机提交，以及访问巨大的数据库和信息资源等。而这些正是HTML使文档的访问、阅读及使用变得如此快捷和有效。

```
meta http-equiv="Content-Type" content="text/html" charset="gb2312">
<link href="1_files/css.css" rel="stylesheet" type="text/css">
```



1

WWW 及其发展现状

作为一种新的信息传播与处理技术，WWW 极大地推动了 Internet 的发展及信息的交流与共享。

WWW 的基本概念

WWW (World Wide Web) 最初起源于 CERN (欧洲粒子物理实验室) Tim Berners-Lee 提出的一项名为“分布式超媒体系统”工程，其最初目的是使科学家们能容易地查阅同行们的文章。后来人们发现这项工程中的超文本和多媒体技术非常适于信息的发布与共享，于是被广泛应用到 Internet 上，并由此成为 Internet 发展的推动器。后来在 MIT 专门成立了一个 W3C 组织，负责 WWW 的技术研讨、标准发布以及应用推广等工作。

在 1990 年 WWW 出现以前，Internet 上的应用大多采用文字终端，其使用对象为少数计算机专业人员。自从 WWW 出现后，Internet 开始摆脱 UNIX 的局限和实验室环境，走进寻常百姓家。在 1993 年 2 月，以 NCSA 开发的 WWW 浏览器 Mosaic 为标志，WWW 初具规模并走向成熟，Internet 上的应用开始转向图形用户界面并提供多媒体功能。

WWW 主要由一些应用软件以及一系列协议和约定组成，它使用超文本和多媒体技术在计算机网络上进行信息的发布和浏览。从概念上来说，WWW 可以应用在所有的计算机网络上，而并非仅 Internet：可以是基于 TCP/IP 协议的 Intranet，也可以是基于其他网络协议（如 IPX/SPX）的网络。对基于不同网络协议的网络，WWW 的实现需要相应的应用软件来支挂。目前 WWW 主要应用在基于 TCP/IP 协议的网络上。

WWW 的工作机制

WWW 服务器端有大量的原始数据（如文本、图像、视频、音频等）和超文本文档（用于组织管理原始数据），这些信息在存放方式、访问模式、数据格式等方面差异很大。如何将这些信息有效地组织起来，让用户方便地访问是 HTML 语言研究的重点之一。WWW 通过 HTML 文件来组织各类原始数据，这种文档可以包含各种数据类型（经 MIME 定义）的数据。其实 HTML 文档中本身并不包含这些数据，而仅仅包含这些数据文件的链接指针。这些链接指针分别指向存放在不同地方的、不同类型的数据。WWW 管理和传输的数据一般是 HTML 文档和原始数据。这些数据通过 HTTP 协议来进行传输。

WWW 采用 Client/Server 模式，WWW 服务器软件（可能还包括其他相关软件，如网关、Proxy Server、扩展应用软件）负责处理客户端的请求，并将有关数据（HTML 文档和原始数据）通过网络返回给客户端。客户端软件（一般是 WWW 浏览器）负责初始化用户的请求后提交给 WWW 服务器，并对服务器返回的数据进行处理和显示（有时还需要相关的软件协助处理）。因此对 HTML 文档的解释处理、对各种网络数据的获取、对各种多媒体数据的处理，都是通过 WWW 浏览器来完成的。浏览器是 WWW 实现过程中的重要组成部分。目前主要有两种 WWW 浏览器：一是 Netscape 公司的 Navigator，一是 Microsoft 的 IE。

在技术方面, WWW 主要由 HTML (超文本标识语言)、HTTP (超文本传输协议)、CGI (通用网关接口)、URL (统一资源定位符)、MIME 等技术组成。其中 HTTP 协议是 WWW 的基础协议, WWW 服务器和客户端软件都是基于 HTTP 协议实现的; WWW 通过 HTTP 协议传输管理的数据主要是超本文档信息, 超本文档是用 HTML 语言编写的; 在超本文档中, 可以包含许多指向不同类型的网络资源的超文本链接, 这些链接用 URL 表示; CGI 是基于 WWW 服务器的一种标准扩展技术; MIME 是用来定义 Internet 数据类型的一种标准。除了 HTTP 协议外, WWW 还涉及其他一些相关协议, 如 FTP 和 Telnet 等, WWW 能兼容这些协议所具有的功能。

WWW 的应用

应该说 WWW 是基于 Internet 的一种应用, 只不过已成为 Internet 上最主要的应用, 实际上它已经涵盖了大部分传统的标准应用。其实人们平时所说的 Internet, 主要指的是其硬件设施和网络本身提供的基本通信服务方面 (从学术上说, 它处于 TCP/IP 协议组的应用层以下, 如物理层、链路层、IP 层和 TCP 层); 而基于它的网络应用则很多。这种网络应用一般分为标准应用 (一般有相应的 RFC 推荐标准) 和专业领域应用 (一般有行业标准或自定义协议标准)。标准应用一般包括文件传输、终端仿真、电子邮件、WWW、Gopher 等应用, 专业应用则随具体的应用需求而异, 在实际应用中, 它们常互为补充。

WWW 之所以被广泛接受, 主要在于其实施简单 (可充分利用以前的系统设备)、管理和维护方便 (可远程维护管理并且采用统一的多媒体界面)、高效率 and 多媒体功能。

WWW 为网络用户提供了一种全新的信息查询、信息浏览和信息发布手段。作为一种信息处理技术, 其应用领域可谓无所不及。当然, 它最主要的应用还在于 Internet/Intranet。这些站点提供的信息和服务涉及到社会生活的各个领域, 常见的应用如下。

1. 信息发布

借助于 Internet 遍布全球的优势, 在网上作广告、宣传、电子出版等。例如宣传公司的产品、图文并茂的风景名胜、人文地理、投资环境介绍等。



2. 数据查询

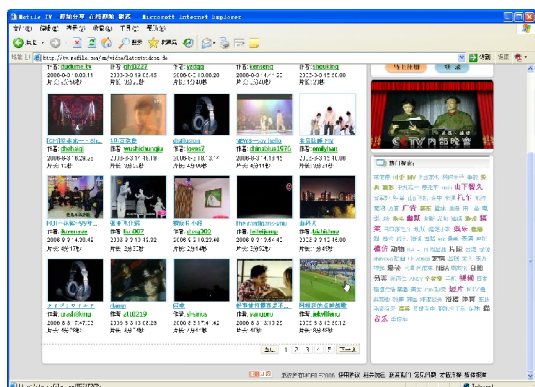
通过搜索引擎, 如 Yahoo、Infoseek, 查询网上的各种信息等。





3. 实况转播

将最新信息实时地在网上传播。在网络传输速率较高的地区，可以实现有限的含图像和声音的转播，例如通过Internet进行围棋、桥牌等转播。



4. 电子商务

通过 WWW 平台进行远程购物、电子商场、电子货币、电子银行等商务活动，是目前 WWW 发展的一个主要方向。



5. 专业应用

通过 WWW 平台进行各个领域的专业应用，是许多行业建立 WWW 站点的主要因素。



WWW的应用不仅在Internet，也广泛用于Intranet（基于TCP/IP的企业内部网）。在Intranet上也可以实现与Internet相同的应用，只是它对信息的传播与访问进行了一定的限制。传统的企业MIS大多版本各异、维护管理困难，但是借助于WWW技术，我们可以建立基于Web的企业MIS系统，并且更加灵活有效。在实际的应用系统中，WWW本身提供的功能是有限的，例如动态交互，就需要根据实际需求开发应用软件，与WWW集成在一起提供服务。开发专门的WWW扩展系统可以借助于Java、CGI等技术来实现。

WWW的站点建设

建设WWW站点首先需要以下相关设备和条件。

- 硬件设施：WWW服务器的硬件平台一般选择PC服务器或工作站，相对而言前者具有较好的性能价格比。但如果对处理效率、网络吞吐量等方面有较高要求，则宜选择工作站。
- 软件平台：主要是服务器操作系统和WWW服务器软件。可选的操作系统主要有UNIX和Windows等；可选的WWW服务器软件很多，如IIS、Apache等。此外为方便管理与调

试, 还应有 WWW 浏览器、主页编辑器、图像处理等辅助性工具。UNIX 和 Windows 各有优势, 从本身实现机制和性能方面来看, UNIX 的要好一些; 但 Windows 与普通用户的距离似乎更近一些。

- 网络平台: 如果尚无网络平台, 还需要建立网络环境(可能需要购买路由器、访问服务器以及集线器等网络设备)。此外, 如果建立的 WWW 站点要提供 Internet 访问, 申请 IP 地址是必须的。IP 地址可以从电信局或拥有剩余 IP 地址的相关公司申请。申请 IP 地址的数量取决于实际的需求, 建立一个 WWW 站点最好拥有 3 个以上的 IP 地址。
- 维护和开发人员: 这方面人员的数量和技术水平应根据应用需要决定。如果建立的 WWW 站点要提供大量的经常变化的信息, 则必须拥有一定数量的维护人员, 负责网页的制作和各种原始数据的处理。如果站点还要提供 HTML 等描述性语言(包括 JavaScript 等语言)不能实现的服务, 如股市行情、新闻信息发布检索系统以及网络沙龙等需要动态数据交换的系统, 则需要采用 ASP、Java 等技术进行专门的应用开发。
- 信息采集与处理: WWW 站点需要提供大量的信息, 有些信息可能还需要特定的系统进行处理。因此, 信息的采集与处理工作很重要。这是站点建立后比较主要的工作。
- 站点安全管理: 如果 WWW 站点与 Internet 相连, 那么安全对某些用户来说就显得很重要了。安全管理不仅涉及到系统硬件和软件方面, 还涉及到公司机密等信息的安全管理。目前, WWW 在技术和应用方面已比较成熟, 大部分企业内部网都引入了 WWW 信息处理机制。对于 ISP 来说, WWW 站点的建设成为其重要任务。对于软件开发商而言, 存在两大方面的机会: 将传统的信息系统改造移植到基于 Web 的应用上来; 另外一个则是开发基于或扩展 WWW 的应用, 例如新闻信息动态出版与浏览系统、基于 WWW 的虚拟现实会议系统、基于 Web 的全文检索系统、数据库网关接口以及传输层网关等中间件等。

2

HTML 简介

Internet 的飞速发展导致了成千上万互联网站的创建, 我们浏览这些网站的时候, 看到的是丰富的影像、文字、图片, 而这些内容都是通过当 HTML 语言表现出来的。对于网页设计和制作人员, 尤其是开发动态网站的编程人员来讲, 制作网页的时候, 如果不涉及 HTML 语言, 几乎是不可能的。

HTML 基本概念

要把信息发布到全球, 就必须使用能够被大众接受的语言, 也就是使用一种大多数计算机能够识别的发布语言。在 WWW 上, 通常使用的发布语言是 HTML, 即超文本标识语言。HTML 语言(HyperText Markup Language, 中文通常称作超文本标识语言或超文本标记语言)是一种解释执行的文本类标记语言, 它是 Internet 上用于编写网页的主要语言。用 HTML 编写的超文本文件称为 HTML 文件。

HTML 语言是一种简易的文件交换标准, 有别于物理的文件结构, 它旨在定义文件内的对象和描述文件的逻辑结构, 而并不定义文件的显示。由于 HTML 所描述的文件具有极高的适应性, 因此特别适合于 WWW 的发布环境。





HTML 是纯文本类型的语言，使用 HTML 编写的网页文件也是标准的纯文本文件。我们可以用任何文本编辑器，例如 Windows 的“记事本”程序打开它，并查看其中的 HTML 源代码，也可以在浏览器打开网页时，通过单击相应的“查看>源文件”命令查看网页中的 HTML 代码。HTML 文件可以直接由浏览器解释执行，而无需编译。当用浏览器打开网页时，浏览器读取网页中的 HTML 代码，分析其语法结构，然后根据解释的结果显示网页内容，正是因为如此，网页显示的速度同网页代码的质量有很大的关系，保持精简和高效的 HTML 源代码十分重要。

HTML 的发展历史

HTML 最初由 Tim Berners-Lee 开发——当他在 CERN 时，因为 NCSA 开发的 Mosaic 浏览器而流行起来；在 1990 年由于 Web 的爆炸性增长而得到成功。这期间，HTML 以多种不同方式被扩充，并且网页作者和零售商共享 HTML，因此这就促进了他们合作制定 HTML 的规范。在 Internet Engineering Task Force (IETF) 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）。HTML+（1993 年）和 HTML 3.0（1995 年）在内容上做了进一步的丰富。尽管从来没有收到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 年的努力结果是 HTML 3.2（1997 年 1 月）。时至今日，HTML 已经发展到了比较成熟的 HTML 4.0 版本，在这个版本的语言中，规范更加统一，浏览器之间的兼容性也更加完好了。HTML 已经发展成各种方式的设备都可以利用网上的信息：蜂窝电话、手持设备、快速输入 / 输出设备、计算机等。

HTML 4.0 的特点

HTML 4.0 通过以下机制扩展了 HTML：样式表、脚本、框架、嵌入对象、改进的从右到左的支持和混合方向文本、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。

国际化：这个版本的 HTML 在设计过程中得到了国际化领域专家的帮助，因此可以用各种语言写文档并易于在世界范围内发布。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。它是处理国际化字符、文本方向、发音和其他语言问题的最具包容性的标准。HTML 文档内提供对不同人类语言的强大的支持。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、更好的连字符连接等。

可访问性：网络社区在增长，而它的成员的能力和技巧各不相同，那么底层技术满足一些特殊需要是很重要的。HTML 已经被设计成可以被有身体缺陷的人访问的网页。HTML 4.0 便于访问的特性包括以下几个方面。

- 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、按语意分类表单控件、Select 选项和活动标签。
- 标记作为内嵌对象的文本描述的能力（使用 Object 元素）。
- 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 元素内）和图像映射（包含在 Area 元素内）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不仅可以方便访问者，还便于设计更容易地采用新技术。

表格：通过表格，设计者可以更好地控制结构和布局。定义好列宽度后可以逐渐地显示表格，而不必等到全部表格绘制完。

混合文档：HTML 提供了在 HTML 文档中嵌入常见媒体对象和应用程序的标准机制。Object 元素（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：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 文档内或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，制作者很少能控制页面绘制过程。HTML 3.2 包括一些属性和元素，它们提供了对对齐方式、字体大小和文本颜色的控制。制作者也使用表格和图像进行页面布局。用户需要较长时间才能升级他们的浏览器，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 协会将逐渐淘汰很多 HTML 的表现元素和属性。

脚本：通过脚本，制作者可以制作动态网页，例如使用 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。

打印：有时，制作者希望阅读者不仅是只打印当前文档。当文档成为一个工程的一部分时，它们之间的关系可以被描述为使用 HTML Link 元素或使用 W3C 的资源描述语言 (Resource Description Language, RDF)。





Chapter

2

快速编写HTML文件

在编写第一个HTML文档之前，网页设计者不需要花费太多的时间去学习详细的语法，因为简单的HTML页面非常容易编写。学会本章的基本编写方法后，会增强读者的学习信心，这种信心来自于很短的时间内就可以创建出优秀的网页页面。

```
meta http-equiv="content-type"
charset="gb2312"
<link href="1_files/css.css" rel="stylesheet" type="text/css">
```



1 使用记事本编写HTML 页面

创建第一个HTML 文档，只需要两个基本工具：HTML 编辑器和Web 浏览器。

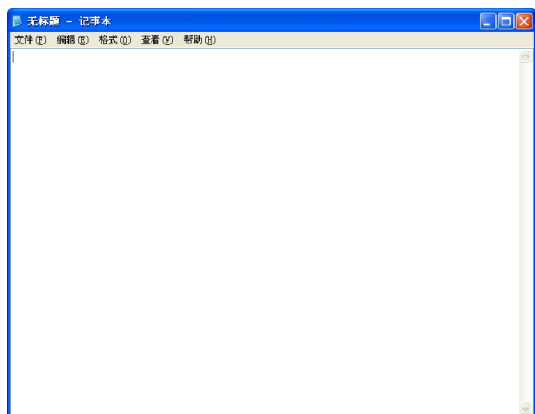
- HTML 编辑器就是用于生成和保存HTML 文档的程序。
- Web 浏览器就是用于浏览和测试HTML 文档的程序。

一般而言，HTML 编辑器分为两类：基于文本或代码的编辑器，在创建文档时只能看到HTML 代码；所见即所得编辑器，在格式化文档时即显示出类似于浏览器窗口的结果。

由于HTML 语言编写的文件是标准的ASCII 文本文件，因此可以使用任何文本编辑器来打开并编写HTML 文件，如Windows 系统中自带的记事本。下面就使用记事本来快速编写第一个HTML 文件。

范例文件	Sample\Ch02\1\1.htm
学习要点	使用记事本编写页面

1 单击“开始>程序>附件>记事本”命令，打开记事本程序。



2 在记事本中直接输入下面的HTML 代码（不区分大小写）：

```
<html >
<head >
<title>使用记事本编写html 文件</
title >
</head >
<body >
<p align="center">
快速编写html 文件
</p >
</body >
</html >
```

3 输入了代码后，记事本中显示出了代码的内容。

