



- XHTML 介绍
- 文字、图像、表格的编排控制
- 超链接的编排控制
- CSS 排版样本
- 框架、多媒体的编排控制
- XML 的基础
- JavaScript 语言
- MIME 文件类型
- XHTML 标记索引

精通 XHTML

萧世文 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



精通 XHTML

萧世文 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

动态 XHTML 编程语言是 HTML 4.0 的改良版本, 现在已逐渐成为网页设计语言标准。本书以大量的实例程序详细介绍了 XHTML 的语法及应用。主要内容包括: XHTML 简介, 文字、图像、表格的编排控制, 超级链接的编排控制, CSS 排版样本, 框架、多媒体的编排控制, XML 的基础, JavaScript 语言, 表单与控件等以及附录的 XHTML 的相关资源。

本书适用于那些想创建主页, 在 Web 上发布信息的所有网络用户使用。

本书繁体字版名为《精通 XHTML》, 由文魁资讯股份有限公司出版, 版权属萧世文所有。本书简体中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可, 任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号: 图字 01-2000-2258 号

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: 精通 XHTML

作 者: 萧世文

责任编辑: 高志凤

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

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

印 刷 者: 北京市昌平环球印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 30 字数: 723 千字

版 次: 2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷

书 号: ISBN 7-900635-60-2

印 数: 0001 ~ 5000

定 价: 53.00 元(含 1 张光盘)

序 言

在进行网络程序设计时，直接编辑网页的文件内容是很重要的一环。网页文件是以 HTML（HyperText Markup Language，超文本标记语言）语言来编写的，HTML 经过数次改版，最终版本是 4.01。之后 HTML 就被 XHTML 取代，后者已成为新的网页设计语言标准。

XHTML 是 eXtensible HyperText Markup Language（可扩展的超文本标记语言）的缩写，是由 HTML 4 改进而成的版本。XHTML 的基础语法与 HTML 完全相同，如果您已经熟悉 HTML，那么学习本书将是驾轻就熟。

XHTML 与 HTML 的差别主要是，XHTML 加强了 XML 的技术。XML（eXtensible Markup Language，可扩展的标记语言）允许自定义标记和属性，让您不再局限于原有的 HTML 标记和属性。有了 XML 的帮助，网页设计人员便可以发挥 XHTML 更大的功能。

本书的内容除了 XHTML 的完整语法介绍之外，还介绍了最新的技术，例如 CSS 排版样本，XML 语言等。书中配有大量的范例程序使读者更容易了解，更容易进入 XHTML 设计的领域。

目 录

第 1 章 XHTML 简介	1
1.1 超文本标记语言——HTML	2
1.1.1 起源	2
1.1.2 规范机构及版本	2
1.2 HTML 文件的结构	3
1.2.1 文件开始与结束标记	4
1.2.2 表头开始与结束标记	4
1.2.3 主体开始与结束标记	5
1.2.4 标记的属性	6
1.3 HTML 4.0 版本的新特性	7
1.4 XML	8
1.5 XHTML	13
1.6 XHTML1.0 的 DTD	15
1.7 XHTML1.1 的 DTD	16
1.8 XHTML Basic 的 DTD	17
1.9 正确的 XHTML 格式	17
1.10 XHTML 与 HTML 的差别	17
1.11 XHTML 的相关软件	19
1.11.1 浏览器	19
1.11.2 XHTML 编辑器	21
1.12 将 HTML 转换成 XHTML	22
1.13 XHTML 基本元素详解	23
1.14 浏览器可辨识的颜色名称	35
第 2 章 文字的编排控制	37
2.1 设置标题文字	39
2.2 设置字体的外观	40
2.3 逻辑性标记	50
2.4 设置字体	61
2.5 使用段落与换行标记	65
2.6 画水平线	67
2.7 默认格式	68
2.8 使用区块	70

第 3 章 图像的编排控制	73
3.1 显示图像	74
3.2 图像的格式	78
3.3 图像的默认大小	78
3.4 改变图像的大小	79
3.5 给图像加上边框	81
3.6 给图像加上间隔	82
3.7 图像与文字的编排	83
3.7.1 垂直编排文字	84
3.7.2 水平编排文字	85
3.8 设置网页背景图片	86
3.9 拼排图片	87
3.10 重叠图片	89
第 4 章 表格的编排控制	91
4.1 表格的基本组成元素	92
4.2 为表格加上边框	103
4.3 设置单元格的间隔	105
4.4 设置单元格文字的对齐方式	108
4.5 设置特殊表格	110
4.6 设置颜色	113
4.7 设置表格与列行的宽与高	115
4.8 设置表格的对齐方式	116
4.9 在表格内使用图形	118
4.10 嵌套表格	119
第 5 章 超级链接的编排控制	121
5.1 创建文字超级链接	124
5.2 创建图像超级链接	125
5.3 创建文件书签超级链接	126
5.4 设置超级链接的颜色	129
5.5 设置 URL	130
5.6 设置链接信息	132
5.7 设置超级链接的快捷键	134
5.8 表格与超级链接	135
5.9 使用超级链接寄邮件	137
5.10 使用超级链接下载文件	138
5.11 创建项目列表	139
5.11.1 使用项目符号	139

5.11.2 使用项目编号	141
5.11.3 使用定义名称	142
5.12 图像映射	148
第 6 章 CSS 排版样本	153
6.1 排版样本的类型	154
6.2 排版样本的种类	157
6.2.1 外部 CSS 文件	157
6.2.2 内嵌式的 CSS 排版样本	159
6.2.3 行内 CSS 排版样本	161
6.3 排版样本的类别	162
6.3.1 样本类别的定义	162
6.3.2 使用 class 属性引用样本类别	163
6.3.3 使用 id 属性引用样本类别	165
6.4 排版样本规范	166
6.5 关于字体的属性	170
6.6 关于背景与颜色的属性	175
6.7 关于文字的属性	179
6.8 关于列表的属性	182
6.9 关于边框的属性	185
6.10 关于方框边界的属性	188
6.11 关于方框内缘的属性	190
6.12 关于超级链接的属性	191
6.13 关于组件定位的属性	193
第 7 章 框架的编排控制	199
7.1 创建框架与框架页	200
7.2 创建直排框架页	204
7.3 创建横排框架页	206
7.4 创建直排与横排混合框架页	208
7.5 为框架页取名称	209
7.6 打开新窗口	213
7.7 当浏览器不支持框架模式时	215
7.8 设置框架页的边框	216
7.9 设置框架页间的间隔	217
7.10 设置框架页边框的颜色	218
7.11 设置框架页的滚动轴	219
7.12 固定框架页的大小	220
7.13 设置浮动框架页	221

第 8 章 多媒体的编排控制	225
8.1 多媒体文件的类型	226
8.1.1 MIME 的格式	227
8.1.2 多媒体文件的工具程序	228
8.2 声音的类型	228
8.3 视频的类型	230
8.4 3D 模型的类型	232
8.5 多媒体制作工具	232
8.6 链接外部多媒体文件	233
8.7 链接背景声音文件	236
8.7.1 使用<bgsound>标记	236
8.7.2 使用<embed>标记	238
8.7.3 使用<object>标记	240
8.8 链接背景视频文件	244
8.8.1 使用标记	244
8.8.2 使用<embed>标记	245
8.8.3 使用<object>标记	246
8.8.4 显示下载信息图片	247
8.9 当浏览器不支持内嵌对象时	248
8.10 使用走马灯显示文字	249
第 9 章 XML 的基础	253
9.1 XML 初探	254
9.2 XML 的文件结构	255
9.3 DTD 文件类型的声明	258
9.4 DTD 的元素	259
9.5 DTD 的属性	263
9.6 Internet Explorer 对 XML 的支持	266
9.7 加载 XML 文件	267
9.7.1 使用 Microsoft.XMLDOM 对象	267
9.7.2 使用数据岛	269
9.7.3 读取元素的属性	271
9.8 浏览 XML 的节点	272
9.8.1 读取节点的类型	274
9.8.2 读取节点的属性	276
9.9 处理加载 XML 的事件	279
第 10 章 JavaScript 语言	283
10.1 开始使用 JavaScript	284

10.1.1	使用<script>标记.....	285
10.1.2	使用外部文件.....	285
10.1.3	<script>标记.....	287
10.1.4	<noscript>标记.....	288
10.2	服务器的 JavaScript.....	289
10.3	批注文字.....	290
10.4	数据类型.....	290
10.5	运算符.....	291
10.6	条件表达式.....	295
10.7	循环表达式.....	299
10.8	函数.....	301
第 11 章	JavaScript 的对象.....	305
11.1	对象.....	306
11.2	XHTML 元素的事件码.....	308
11.3	创建对象.....	315
11.4	String 对象.....	317
11.5	Array 对象.....	319
11.6	JavaScript 对象总览.....	321
11.7	决定浏览器的种类.....	322
11.8	document 对象.....	324
11.8.1	设置网页的属性.....	325
11.8.2	打开与关闭新网页.....	326
11.9	window 对象.....	332
11.10	location 对象.....	341
11.11	history 对象.....	342
11.12	cookie.....	343
11.12.1	创建 cookie.....	343
11.12.2	读取 cookie 的内容.....	345
第 12 章	表单与控件.....	347
12.1	控件的种类.....	348
12.2	创建表单及加入控件.....	369
12.3	使用 JavaScript 读取控件.....	370
12.4	传输表单的数据.....	372
12.5	使用下拉式列表.....	375
12.6	使用图片按钮.....	381
12.7	使用<fieldset>, <legend>与<label>.....	383
12.8	XForms.....	389

第 13 章 动态 XHTML	391
13.1 动态排版格式	392
13.2 选择排版样本	396
13.3 动态改变图片	398
13.4 使用 writeln 方法	400
13.5 使用 insertAdjacentHTML 方法	401
13.6 使用 innerHTML 与 outerHTML 方法	402
13.7 使用 createTextRange 方法	405
13.8 使用 createElement 方法	407
13.9 动态创建表格	408
13.10 使用 VML	410
13.11 视觉特效: Filters	411
13.12 视觉特效: Transitions	413
第 14 章 数据的处理	419
14.1 元素的拖放	420
14.2 数据链接	422
14.3 使用 MSHTML 链接数据	423
14.4 使用 MSHTML 创建表格	427
14.5 使用 TDC 链接数据	429
14.5.1 创建 TDC 文件	429
14.5.2 TDC 的 classid	430
14.5.3 使用<object>元素引用 TDC	430
14.6 使用 TDC 创建表格	433
14.7 使用 XML 链接数据	435
14.7.1 创建 XML 文件	435
14.7.2 读取 XML 文件	436
14.8 使用 RDS 链接数据	439
14.9 元素的 Behaviors	440
附录 A 特殊字符代码对照表	443
附录 B 颜色对照表	449
附录 C MIME 文件类型	453
附录 D XHTML 标记索引	461

第 1 章

XHTML 简介

- 超文本标记语言——HTML
- HTML 文件的结构
- HTML 4.0 版本的新特性
- XML
- XHTML
- XHTML 1.0 的 DTD
- XHTML 1.1 的 DTD
- XHTML Basic 的 DTD
- 正确的 XHTML 格式
- XHTML 与 HTML 的差别
- XHTML 的相关软件
- 将 HTML 转换成 XHTML
- XHTML 基本元素详解
- 浏览器可辨识的颜色名称

XHTML 是 eXtensible HyperText Markup Language 的缩写, 直译为“可扩展的超文本标记语言”, XHTML 是一种用来描述网页内容所用的计算机语言。自从 Internet 开始流行以后, 浏览网页、下载文件、收发电子邮件或是玩网络游戏等, 已经成为许多人的固定休闲活动, 同时也为许多公司造就了无穷的商机。虽然现在市面上已有许多的网页编辑工具软件, 可以让您快速地编辑网页。但是有许多的功能, 是这些网页编辑工具软件所无法做到的。想要突破这些障碍, 就必须直接由描述网页内容所用的计算机语言着手。

当您在浏览网页时, 显示在网页上的文字或者图片数据, 从您浏览的网站到您的计算机之间, 是以“超文本标记语言”的类型传输。所谓超文本标记语言(HTML)是将网页上的数据以一种特定的语言来描述。当您计算机上的浏览器, 如 Microsoft Internet Explorer 或 Netscape Navigator, 接收到对方网站传过来的超文本标记语言数据时, 浏览器会解释其内容, 并且将解释的结果显示在您的计算机屏幕上。

1.1 超文本标记语言——HTML

1.1.1 起源

在 60 年代, IBM 开发了一种称为 GML (Generalized Markup Language, 广义标记语言) 的计算机语言来描述与排版其内部文件。之后, IBM 又改进了 GML, 成为 SGML(Standard Generalized Markup Language), 同时 SGML 也广泛地被工业界采用, 成为计算机文件的标准格式化语言。ISO 也在 1986 年将 SGML 列为国际标准规范。

1989 年, 两位 CERN (European Laboratory for Particle Physics, 欧洲粒子物理实验室) 的研究人员 Tim Berners-Lee 和 Anders Berghund 将 SGML 简化成一种以标记文字(Tag-based) 为基础, 可以在互联网上使用的计算机语言, 称为 HTML(HyperText Markup Language)。

HTML 一开始只是用来描述静态的文字或图像, 但是经过了许多人的努力, 今天的 HTML 已能做出许多动态的效果。例如网上聊天, 播放网络电视和音乐等。

1.1.2 规范机构及版本

所有的 HTML 规格都是由国际机构 W3C(World Wide Web Consortium)统一制定的, W3C 的网址是: <http://www.w3.org>。

HTML 的版本如下表所示:

版本	制定日期	规范所在网址
HTML 1.0	1993 年 1 月	
HTML 2.0	1995 年 2 月	http://www.w3.org/MarkUp/html-spec
HTML 3.0	1995 年 3 月	http://www.w3.org/MarkUp/html3/
HTML 3.2	1997 年 1 月	http://www.w3.org/TR/REC-html32

续表

版本	制定日期	规范所在网址
HTML 4.0	1998 年 4 月	http://www.w3.org/TR/1998/REC-html40-19980424
HTML 4.01	1999 年 12 月	http://www.w3.org/TR/html4
XHTML 1.0	2000 年 1 月	http://www.w3.org/TR/xhtml1
XHTML 1.1	2000 年 1 月	http://www.w3.org/TR/xhtml11
XHTML Basic	2000 年 2 月	http://www.w3.org/TR/xhtml1-basic

HTML 在 4.01 版之后,即改称为 XHTML。原因在于 HTML 是一种以标记(Tag)为基础的计算机语言,所有的标记都是由 W3C 统一制定,程序人员不能修改或自己制定新的标记。

XHTML 则无此限制,因为 XHTML 是将 HTML 与一种称为 XML(eXtensible Markup Language)的新技术结合所产生的新计算机语言。利用 XML 的功能,程序人员可以根据自己的需要制定新的标记。同时 XHTML 也修改了 HTML 杂乱无章的文件格式,XHTML 严格要求网页文件必须符合其规范要求。

XHTML 现行的版本是 1.0 版,但是另有 1.1 版的试用版。

1.2 HTML 文件的结构

既然 XHTML 是由 HTML 改良而来,所以两者的语言特性并无多大差别。我们先看一个 HTML 的例子。

图 1.1 是一个简单的网页范例。在 Microsoft Internet Explorer 内执行菜单命令【文件】|【打开】,载入书中所附光盘的\Ch1\ex1-1.html 文件,即显示如图 1.1 所示的网页。

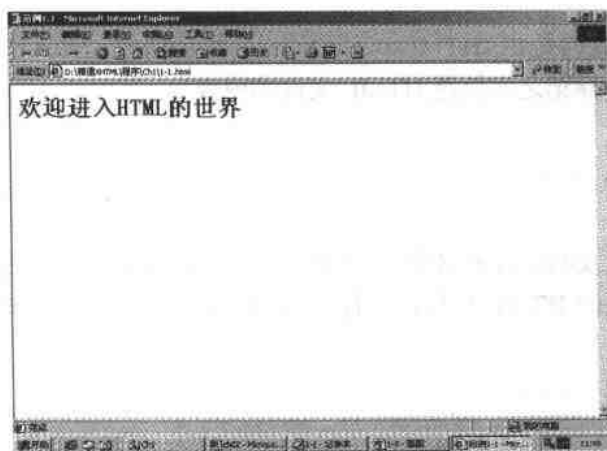


图 1.1 一个简单的HTML网页

接着执行 Internet Explorer 菜单命令【查看】|【源文件】,即会出现 Windows 的记事本,并且显示一段文字,如图 1.2 所示。

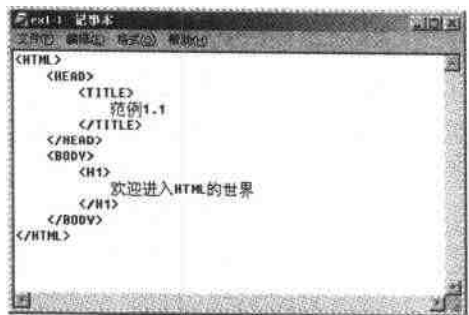


图1.2 图1.1的网页内容

在图 1.2 中, 可以看到许多由符号包含的文字, 例如<HTML>、<HEAD>、<BODY>等, 这些称为标记(Tag)。HTML 文件就是由这些标记来组成其文件的结构。例如图 1.2 中的文字, 就是用来显示图 1.1 的网页。HTML 文件的标记主要由下列 3 个标记组成:

- ◆ 文件开始标记<HTML>与结束标记</HTML>
- ◆ 表头开始标记<HEAD>与表头结束标记</HEAD>
- ◆ 主体开始标记<BODY>与主体结束标记</BODY>

```
<HTML>    → 文件开始标记
<HEAD>    → 表头开始标记
           表头区
</HEAD>   → 表头结束标记
<BODY>    → 主体开始标记
           主体区
</BODY>   → 主体结束标记
</HTML>   → 文件结束标记
```

1.2.1 文件开始与结束标记

HTML 文件的结构是由<HTML>标记开始, 并且通常是由</HTML>标记结束。在<HTML>与</HTML>标记之间就是 HTML 文件的内容。

```
<HTML>
.....
</HTML>
```

但是由于 HTML 文件的结构松散的缘故, 许多浏览器对于一个没有开始标记<HTML>或结束标记</HTML>的 HTML 文件, 一样可以显示相同的结果。例如:

```
<HTML>
.....
```

XHTML 文件则必须含有开始标记<HTML>与结束标记</HTML>。

1.2.2 表头开始与结束标记

表头开始标记<HEAD>与表头结束标记</HEAD>所包含的表头区, 含有此 HTML 的

许多文件信息。例如文件的标题、作者、日期、关键字等。这些信息除了标题会显示在浏览器的窗口标题栏外，其余的信息文字都不会被浏览器显示出来，但是可以作为参考使用。例如搜索引擎在搜索网页时，就是搜索 HTML 文件表头区的标题与关键字栏，来与您输入的搜索文字做比较。

表头区内的每一项数据都由标记包含起来。您能够使用的标记都已经事先定义好了，HTML 文件不允许用户自定义新的标记。但是 XHTML 文件则可以让用户通过 XML 的技术，来自定义新的标记。

在图 1.2 中，我们在表头区加入了标题文字“范例 1.1”，这个标题文字使用<TITLE>与</TITLE>标记包含起来。标题文字会出现在浏览器的窗口标题栏（如图 1.1 所示）。<TITLE>与</TITLE>标记是由 HTML 规格所制定的标记，用来设置文件标题。

```
<HTML>
  <HEAD>
    <TITLE>
      范例 1.1
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      欢迎进入 HTML 的世界
    </H1>
  </BODY>
</HTML>
```

表头标记与表头区是可以省略的。由于 HTML 文件的结构松散的特性，如果您没有写上表头开始标记<HEAD>与结束标记</HEAD>，或是只使用开始标记<HEAD>，而直接使用<TITLE>与</TITLE>标记的话，如下例：

```
<HTML>
  <TITLE>
    范例 1.1
  </TITLE>
  <BODY>
    <H1>
      欢迎进入 HTML 的世界
    </H1>
  </BODY>
</HTML>
```

许多浏览器一样可以显示相同的结果。

1.2.3 主体开始与结束标记

主体开始标记<BODY>与主体结束标记</BODY>所包含的主体区，含有此 HTML 文件实际显示在浏览器窗口上的信息。每一个网页的显示内容，包含的文字、静态图片、动态图像、声音等数据都是放在主体区内。

主体区内的每一项数据也是由标记包含起来。这些标记都是由 HTML 规则所事先制定的，用户不能在 HTML 文件中自定义标记，但是如果是 XHTML 文件则允许用户自定义标记。

以图 1.2 为例,我们在网页上显示一行文字“欢迎进入 HTML 的世界”,就是使用<H1>与</H1>标记包含起来。<H1>标记内的文字会以粗体字显示。

```
<HTML>
  <HEAD>
    <TITLE>
      范例 1.1
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      欢迎进入 HTML 的世界
    </H1>
  </BODY>
</HTML>
```

由于 HTML 文件结构松散的特性,<BODY></BODY>,或是<H1></H1>不必成对出现。例如:

```
<HTML>
  <HEAD>
    <TITLE>
      范例 1.1
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      欢迎进入 HTML 的世界
    </H1>
  </HTML>
```

或是

```
<HTML>
  <HEAD>
    <TITLE>
      范例 1.1
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      欢迎进入 HTML 的世界
    </BODY>
  </HTML>
```

都可以显示相同的结果。

1.2.4 标记的属性

如前所述,HTML 的信息是包含在标记内。可以设置这些标记的属性,让 HTML 文件显示特殊的效果。要在标记内设置属性,使用下列格式:

<标记名称 属性名称=数值 属性名称=数值.....>

一个标记可以有多个属性,属性名称之间以空白字符隔开。例如前例的<H1>标记有

一个 ALIGN 属性，用来设置文字的对齐方式。

```
<HTML>
  <HEAD>
    <TITLE>
      范例 1.1
    </TITLE>
  </HEAD>
  <BODY>
    <H1 ALIGN="CENTER">
      欢迎进入 HTML 的世界
    </H1>
  </BODY>
</HTML>
```

如图 1.3 所示，“欢迎进入 HTML 的世界”的文字被排列在浏览器窗口的中间。

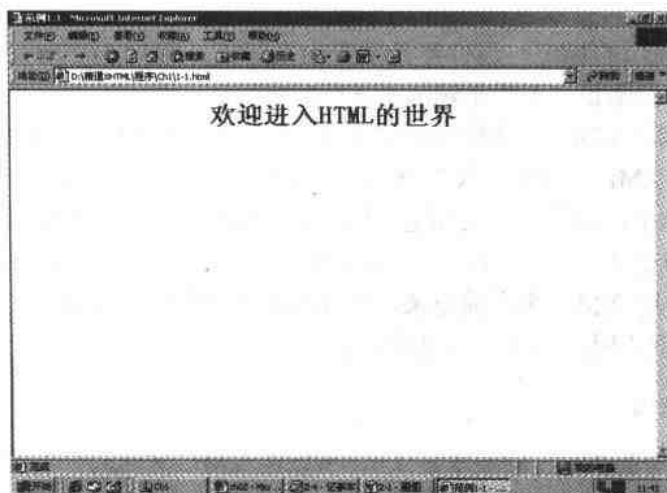


图1.3 使用标记的属性范例

1.3 HTML 4.0 版本的新特性

HTML 4.0 之前的版本，都是将一个网页所要显示的信息全部放在同一个 HTML 文件中，就像是大杂烩一样，毫无结构性可言。自从 4.0 版开始，HTML 定义了新的文件结构，将文件的结构码与实际显示码分开来。

什么是文件的结构码？例如文字的字体与颜色、背景图片文件、段落的对齐方式等，这些结构码决定显示信息所表现的类型。例如标记包含文字的字体与颜色，<CENTER>标记包含段落的对齐方式，<BGCOLOR>属性决定网页的背景颜色等。

从 HTML 4.0 开始，建议不要使用这些文件的结构码。取而代之的是将 HTML 的文件结构写在另外的文件类型定义文件（DTD，扩展文件名*.dtd）与排版样本文件（CSS，扩展文件名*.css）中。而原来的网页则只保存要显示在显示器上的文字与图片。不过大多数的浏览器并没有将建议不使用的标记或属性直接丢弃，而是同时支持 HTML 4.0 之前的