

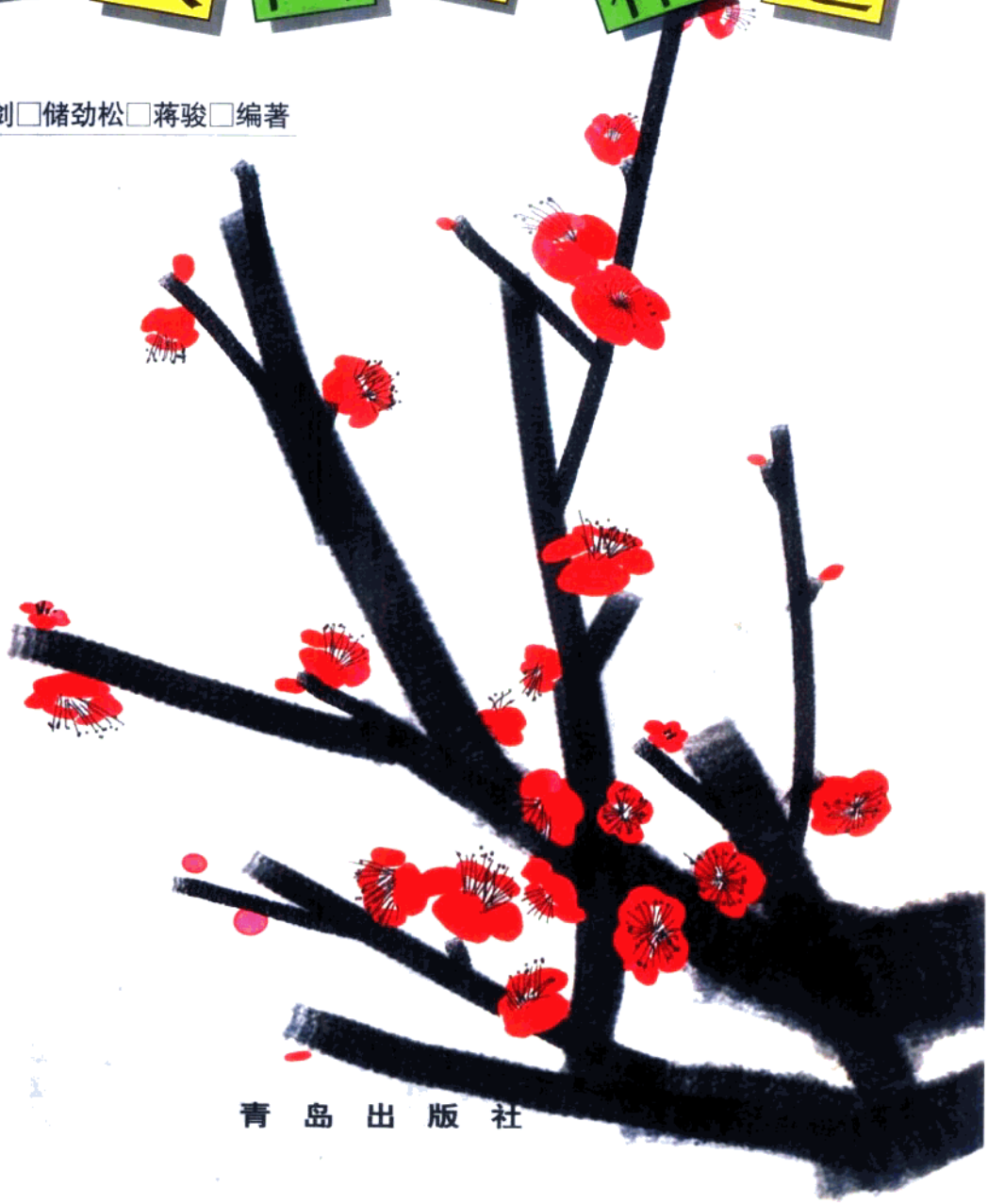


青松

DHTML

从入门到精通

王剑□储劲松□蒋骏□编著



青 岛 出 版 社

出版者的话

有史以来，没有哪一门科学能像电脑这样飞速发展！新技术层出不穷，新产品不断涌现，电脑工作者必须不断学习、更新知识，才能跟上形势，不被淘汰。然而人们的精力是有限的，面对良莠不齐、铺天盖地而来的各种电脑著述和技术资料，你不可能有很多的时间一一鉴别和阅读。这时就需要专家们根据自己的实践经验给以精选和引导。

为此，青岛出版社聘请了具有丰富教学经验和实践经验的专家，组成《青岛松岗电脑图书》编委会，向广大读者介绍适合我国国情的、最新最实用的电脑及网络技术。

《青岛松岗电脑图书》编委会对这套丛书的质量负责，并郑重承诺：编、校、印刷质量符合国家新闻出版署的质量要求——差错率低于万分之一。

《青岛松岗电脑图书》编委会由以下人员组成：

主 任：徐 诚 青岛出版社编审、社长兼总编辑

副主任：钟英明 台湾中兴大学教授

委 员：（按姓氏笔划排列）

叶 涛 西安交通大学副编审

庄文雄 青岛松岗信息技术有限公司总经理

孙其梅 青岛大学教授

吕凤翥 北京大学高级工程师

陈国良 中国科技大学教授

张德运 西安交通大学教授

陆 达 清华大学博士

樊建修 青岛出版社编审

目 录

第 1 章 DHTML: HTML 标准的沿革	(1)
1.1 HTML 的发展	(1)
1.1.1 关于 World Wide Web	(1)
1.1.2 HTTP 协议和服务器	(2)
1.1.3 Web 浏览器	(3)
1.1.4 SGML、HTML 和 CGI	(4)
1.2 静态 HTML 存在的问题	(5)
1.2.1 文本格式的限制	(6)
1.2.2 布局控制的局限	(6)
1.2.3 动态内容无法改变	(6)
1.3 DHTML 特性	(7)
1.3.1 文档对象模型	(7)
1.3.2 脚本语言支持	(7)
1.3.3 可改变文本的格式	(7)
1.3.4 可下载字体	(8)
1.3.5 元素定位与 Z 顺序	(10)
1.3.6 数据感知、捆绑和动态表单	(11)
1.4 样式单与脚本简介	(13)
1.4.1 样式单	(13)
1.4.2 脚本	(14)
本章小结	(14)
第 2 章 级联样式单	(19)
2.1 样式元素	(19)
2.1.1 字体	(19)
2.1.2 颜色和背景	(21)
2.1.3 文本	(22)
2.1.4 边框和边界	(24)
2.1.5 分类	(27)
2.1.6 CSS 单位	(29)
2.2 CSS 语法	(29)
2.2.1 利用在线式样式单指定样式	(30)
2.2.2 利用全局式样式单指定样式	(32)

2.2.3	利用链接式样式单指定样式	(33)
2.2.4	CSS 的继承和“级联”	(36)
2.2.5	CSS 的注释	(39)
2.3	CSS 的高级属性	(39)
2.3.1	定义 CSS 样式的类	(39)
2.3.2	定义单独指定的样式	(42)
2.3.3	伪类和元素	(43)
2.3.4	使用上下文选择准则	(44)
2.3.5	为单个元素指定样式	(46)
2.3.6	组合式样式单	(47)
2.4	CSS 的单位简介	(48)
2.5	CSS 的应用和提高	(48)
2.5.1	文本样式属性的应用	(48)
2.5.2	颜色和背景属性的应用	(52)
2.5.3	块元素的格式和操作	(54)
	本章小结	(56)
第 3 章	JavaScript 语言	(57)
3.1	介绍一段 JavaScript 小程序	(57)
3.2	数据类型和变量	(58)
3.2.1	数据类型	(58)
3.2.2	变量	(59)
3.2.3	常量	(61)
3.3	表达式和运算符	(61)
3.3.1	表达式和运算符	(61)
3.3.2	运算符的优先级	(63)
3.4	语句和注释	(64)
3.4.1	语句	(64)
3.4.2	语句块	(64)
3.4.3	注释	(64)
3.4.4	分号	(66)
3.5	函数	(66)
3.5.1	预定义函数	(67)
3.5.2	用户定义的函数	(68)
3.5.3	函数的返回值	(69)
3.5.4	函数的使用	(69)
3.5.5	变量的作用域	(69)
3.6	流程控制	(71)
3.6.1	if 结构	(71)
3.6.2	if...else 结构	(72)
3.6.3	for 循环结构	(72)
3.6.4	While 循环结构	(73)

3.6.5	break 和 continue 语句	(73)
3.7	对象	(74)
3.7.1	面向对象编程	(75)
3.7.2	Java Script 的对象模型	(75)
3.7.3	Java Script 内部对象	(75)
3.7.4	用户定义的对象	(76)
3.8	数组	(77)
3.8.1	创建一个空数组	(78)
3.8.2	填充数组	(78)
3.8.3	删除数组和数组元素	(79)
3.9	Java Script 编程举例	(79)
3.10	Java Script 的 Internet 资源	(82)
	本章小结	(82)
第 4 章	DHTML 对象模型	(83)
4.1	HTML 文档的对象处理方法	(83)
4.2	对象层次	(84)
4.3	集合和元素	(85)
4.3.1	访问集合中元素的方法	(86)
4.3.2	集合的属性	(88)
4.3.3	元素的方法	(89)
4.4	窗口对象	(89)
4.4.1	窗口对象的层次结构	(90)
4.4.2	窗口对象的属性和方法	(90)
4.4.3	位置(location)对象	(95)
4.4.4	历史(history)对象	(96)
4.4.5	文档(document)对象	(96)
4.4.6	框架(frames)对象	(96)
4.4.7	导航(navigator)对象	(100)
4.4.8	屏幕(screen)对象	(103)
4.4.9	事件(event)对象	(104)
4.5	文档对象	(104)
4.5.1	document.forms[]属性	(105)
4.5.2	document.title 属性	(106)
4.5.3	document.wrtie()方法	(106)
4.5.4	锚(anchors)对象	(108)
4.5.5	链接(links)对象	(109)
4.5.6	body 对象	(110)
4.5.7	层(layer)对象	(113)
4.5.8	表单(form)对象	(116)
	本章小结	(118)

第 5 章 DHTML 事件处理	(119)
5.1 事件和事件处理	(119)
5.1.1 事件	(119)
5.1.2 事件(event)对象	(119)
5.1.3 事件处理程序	(120)
5.1.4 事件类型	(120)
5.2 基于文档的事件	(121)
5.2.1 Load(加载)事件	(121)
5.2.2 Unload(卸载)事件	(122)
5.2.3 mouseOver(鼠标移至)和 mouseOut(鼠标移出)事件	(123)
5.2.4 keyPress(按键)、keyDown(落键)和 keyUp(起键)事件	(124)
5.3 基于表单的事件	(125)
5.3.1 click(单击)、mousedown(鼠标按下)和 mouseUp(鼠标弹起)事件	(126)
5.3.2 submit(提交)和 reset(复位)事件	(126)
5.3.3 focus(聚焦)和 blur(散焦)事件	(128)
5.3.4 change(改变)和 select(选择)事件	(129)
5.4 基于锚(anchor)的事件	(130)
5.4.1 click(单击)和 dblClick(双击)事件	(130)
5.4.2 mousedown(鼠标按下)和 mouseUp(鼠标弹起)事件	(133)
5.4.3 mouseOver(鼠标移至)和 mouseOut(鼠标移出)事件	(133)
5.5 基于元素的事件	(133)
5.5.1 load(加载)事件	(134)
5.5.2 error(加载错误)事件	(134)
5.5.3 abort(放弃加载)事件	(134)
5.6 基于窗口的事件	(134)
5.6.1 load(加载)和 unload(卸载)事件	(134)
5.6.2 focus(聚焦)和 blur(散焦)事件	(135)
5.6.3 move(移动)和 dragDrop(拖放)事件	(135)
5.6.4 resize(重设大小)事件	(135)
5.7 事件冒泡	(136)
5.7.1 包含层次	(136)
5.7.2 采用事件冒泡	(137)
5.7.3 取消事件冒泡	(137)
本章小节	(138)
第 6 章 布局和定位	(139)
6.1 JavaScript 样式单(JSSS)简介	(139)
6.1.1 JSSS 和 CSS 的主要区别	(139)
6.1.2 JSSS 的新对象	(140)
6.1.3 JSSS 的条件样式和使用上下文方法	(141)
6.2 使用样式单定位	(143)
6.2.1 使用 CSS 定位	(144)

6.2.2 使用层(<layer>)定位	(146)
6.3 绝对定位和相对定位	(147)
6.3.1 绝对定位(Absolute Positioning)	(147)
6.3.2 相对定位(Relative Positioning)	(149)
6.4 定位属性	(151)
6.4.1 position 属性	(151)
6.4.2 top 和 left 属性	(151)
6.4.3 pageX 和 pageY 属性	(153)
6.4.4 width 和 height 属性	(154)
6.4.5 overflow 属性	(154)
6.4.6 clip 属性	(155)
6.4.7 Z-index、above 和 below 属性	(156)
6.4.8 SRC 和 source-include 属性	(156)
6.4.9 visibility 属性	(157)
6.5 通过层的移动实现动画	(159)
本章小结	(162)
第7章 动态样式和内容	(163)
7.1 动态改变文本属性	(163)
7.1.1 改变文本的属性	(163)
7.1.2 改变文本的尺寸	(164)
7.1.3 改变文本的颜色	(165)
7.2 动态改变文本样式	(167)
7.3 动态改变文本的内容	(178)
本章小结	(186)
第8章 数据绑定	(187)
8.1 什么叫数据绑定	(187)
8.1.1 静态 HTML 的例子	(187)
8.1.2 动态 HTML 的例子	(189)
8.2 数据源对象	(190)
8.2.1 表格数据控件(TDC)	(190)
8.2.2 数据文件	(191)
8.3 绑定 HTML 标记元素与数据源对象	(192)
8.3.1 显示数据文件中的一条记录	(192)
8.3.2 支持数据绑定的 HTML 标记	(193)
8.3.3 设定数据的格式	(193)
8.3.4 将数据绑定到表格中	(194)
8.4 数据源对象的脚本编程	(196)
8.4.1 TDC 的属性和方法	(196)
8.4.2 数据排序	(196)
8.4.3 筛选数据	(198)
8.5 服务器动态网页技术简介	(200)

本章小结.....(201)

第 9 章 DHTML 与其他语言和对象插件.....(202)

9.1 使用 VBScript 替代 JavaScript.....(202)

9.2 使用 ActiveX.....(205)

9.2.1 使用 ActiveX 进行多媒体转换.....(206)

9.2.2 ActiveX 控件和多媒体滤波器.....(209)

9.3 使用 Java Applet.....(213)

9.4 在 DHTML 中嵌入 VRML.....(217)

本章小结.....(218)

附录 DHTML 对象、集合、方法及属性.....(219)

第 1 章 DHTML: HTML 标准的沿革

WWW(World Wide Web)是媒体领域的一场革命,人们可以利用 Internet 进行信息的交流,它所具有的图形图像技术、音频技术、有限的内部交互(CGI)和易于管理等优势是传统媒体无法企及的,正是这些特性受到了人们的极大青睐。

Web 的核心是超文本标记语言(HTML),这种基本的描述性语言可以创建在 Web 上传输和浏览的广告和信息页。在 Web 技术飞速发展的同时,HTML 语言却没有什麼变化。当今网络软件技术的开发商们投入了巨大的热情进行竞赛,追赶着人们登录 Internet 的狂热潮流。Web 站点的作者们正在寻找一种用最小的努力就能让他们的站点吸引人的工具(对那些处理文字、图形设计和做页面框架的人来说尤其显得迫切)。开发浏览器的软件公司急切地需要扩展 HTML 的功能,以满足那些想要在不同信息的显示上有更多控制权的网页制作者的需要,而正是这些新技术在同一水平上的合并引起了新标志的创立并最终导致了基于 HTML 的新 Web 技术——动态超文本标记语言(DHTML)的产生。

本章在介绍 WWW 及 HTML 发展的基础上,讲述导致 DHTML 发展的因素及其如何成为 Internet 上的发展重点。尽管微软公司(Microsoft)和网景公司(Netscape)对 DHTML 的理解存在着差异,但有一点是共同的: DHTML 为创建新的改革性的 Web 页提供更复杂和更引人注目的 Web 应用接口,使文件可以根据用户的动作控制内容本身、内容的位置和内容的外观。

1.1 HTML 的发展

1.1.1 关于 World Wide Web

1989 年 CERN 高能物理实验室的 Tim Burners-Lee 开始了 World Wide Web 项目的研究开发。当时,这一项目的目标是找出与分散在世界上的其他雇员和研究人员共享研究成果和思想的一条途径。在原来的计划中,Web 被称为“超文本项目”。超文本(Hypertext)是 Ted Nelson 在 60 年代杜撰的一个术语,它表示一种含有到其他文件的链接的文本,在这种文本中,读者可以单击一个或一组特定的词组来得到某一个相关题目的附加信息。而超媒体是一个内容比较丰富的术语,用于说明含有诸如声音和视频图像的多媒体格式信息的文件。

从技术的角度来说,World Wide Web 指的是一个抽象的信息场所。在这里,计算机控制着信息。而 Internet 一般指网络的物理场所,即由光纤、电缆和计算机组成的硬件。Internet 和 World Wide Web 的基础都是使用协议,也就是计算机通信使用的语言和规则。例如, TCP/IP(Transmission Control Protocol/Internet Protocol,即传输控制协议与互联协议)就是一套

联网协议,它能够使不同类型的计算机相互通信,同时它也是 Internet 的基础协议。World Wide Web 并不是一种协议,但是它是将若干协议例如 FTP(File Transfer Protocol 文件传输协议)、Telnet(远程登录,虚拟终端协议)、WAIS(Wide-Area Information Servers,广域信息服务)等连接在一起的载体。由于 World Wide Web 使用标准 Internet 协议来传输文件和文档,所以 WWW 经常被用做 Internet 的同义语,以表示计算机及信息主体的广域网络。

1.1.2 HTTP 协议和服务

WWW 上的信息交流是运作于客户——服务器模型之下的。一个 Web 服务器是指一台被设置为给其他计算机客户提供文件服务的计算机。由于该服务器仅在用户请求打开文件时才运作,所以这是共享文档的一种有效方法,因为它只占用一小部分服务器资源。

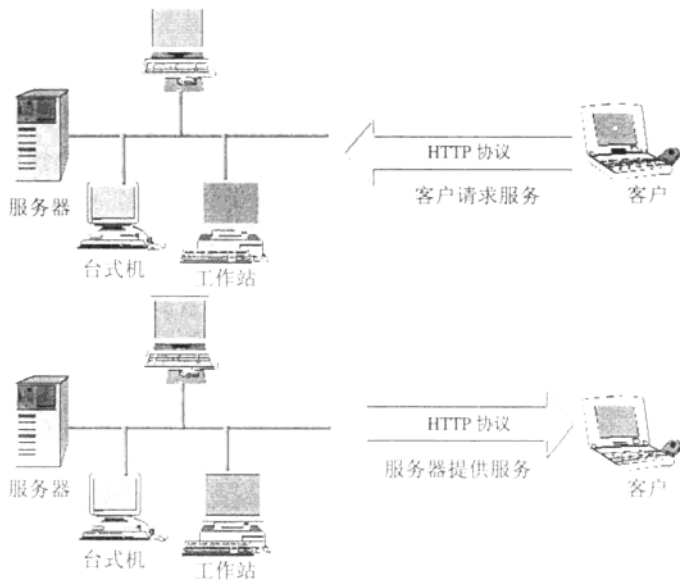


图 1.1 客户和服务连接

WWW 上文档的传输和浏览离不开 HTTP(Hypertext Transmission Protocol, 超文本传输协议),所有 Web 客户和服务都必须遵循 HTTP,以便发送和接收超媒体文件。可以说,WWW 的成功应归功于 HTTP 能够处理多应用程序协议的能力。多应用程序协议使得用户能够访问众多的 Internet 资源,如匿名 FTP、WAIS 资料服务等。HTTP 的多媒体能力包括支持文本的检索和显示,图形、动画、音频、视频的播放等。通常 Web 服务器也被称为 HTTP 服务器。

WWW 服务器主要运行在 UNIX 上,但是它们适应于许多平台和环境,其中包括 Windows NT、OS/2、Macintosh、Linux 等。由于图形接口的 32 位操作系统平台如 Windows NT、OS/2、Macintosh 的普及,以及 Linux 的异军突起,基于 UNIX 的 WWW 服务器在 Internet 的优势地位将逐渐被改变。

1.1.3 Web 浏览器

在运行一个 Web 客户(通常称为一个 Web 浏览器)时,该客户通过向某个计算机的 Web 服务器发送 Web 文档请求,连入由一个网络地址所指的那个计算机上,这个网络地址被称为 URL(Uniform Resource Locator, 标准资源定位器)。服务器通过把客户所请求的文本及文本中的一个超级链接所用的任何其他媒体(图像、音频等)发送给这个用户的 PC 机作为响应。由服务器送回的文档是 HTML(Hypertext Markup Language, 超文本标记语言)格式。HTML 能使读者通过单击一个单词、词组甚至是一个图像来存取文件或链接到其他 HTML 文件上。世界各地服务器通过这种文件和文档之间的超文本链接形成了一张紧密联系的巨大的网。

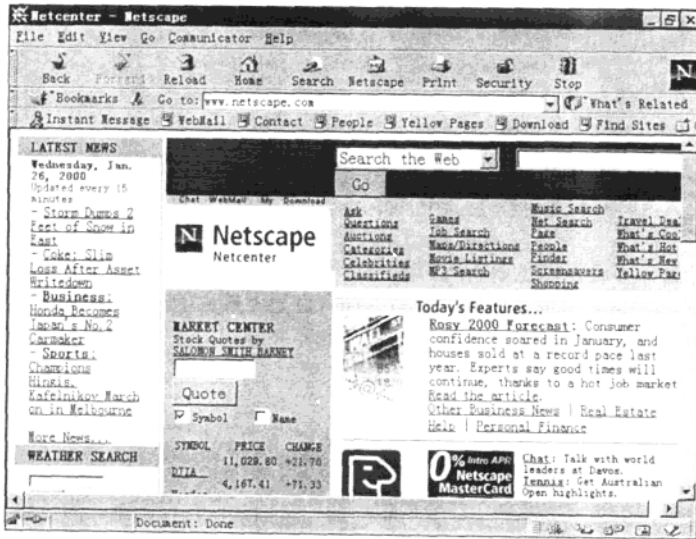


图 1.2 Netscape Navigator 及 Netscape 公司主页

Web 浏览器主要有两大阵营: Netscape Navigator 和 Microsoft Internet Explorer。因为同一个文档在不同浏览器中的样子互不相同,所以创建一个 WWW 上的统一标准是非常必要的。Web 上的两个主要标准是 HTTP 协议和 HTML 语言。这些基本的 Web 技术并不具备普遍基础。HTML 文件是为某种浏览器制定的,而不是为许多浏览器协同工作制定的。这就需要专门的 W3C(World Wide Web Consortium, 世界互联网协会)组织进行协调,以确保这一技术能在网上应用。但是由于互联网巨大的商业利益,浏览器公司都卷入了一场无休止的标准竞赛之中,甚至在 HTML 标准还未向大众公布之前,Microsoft、Netscape 等软件公司就在技术所能及的范围内不断推出他们的升级版本。所以我们的任务是,在了解众多版本差异的基础上找出共同点,解决兼容性问题,开发出跨平台的、设计精美的 HTML。

Netscape 公司是最早开发 WWW 浏览器的公司之一, Netscape Navigator 的出现至今仍然是 Internet 发展史上影响深远的一件事。尽管 Netscape 已经被 Sun 兼并,它仍然不断更新着 Netscape Navigator,并把最新的成果应用在其中。Netscape 对 DHTML 对象的使用可以追溯到 Navigator 2.0 版。



图 1.3 Microsoft Internet Explorer 及 Microsoft 公司主页

Microsoft 公司尽管在浏览器市场的名声并不太好，也有很多人看不惯 Microsoft 的“霸权”作风，但它作为世界上最大的软件公司并拥有世界上最大的浏览器市场却是不争的事实。而且 Microsoft 在一开始就免费发送浏览器软件和不断地对它进行技术升级的确是 WWW 迅速为世界所接受的重要推动力。

1.1.4 SGML、HTML 和 CGI

早期的 Internet 是一个多平台环境，该环境使文档互换存在困难。为了解决文档共享的问题，研究者发明了一种称为 SGML(Standard Generalized Markup Language, 标准通用标记语言)的专业语言。SGML 把重点放在一个文档中的组成部件上，所以信息的接收者就可以不受信息发送者专有选择的束缚。SGML 文档允许重新制定窗口的尺寸，便于浏览者最佳地利用自己的显示器；它允许打印这些文档，并且打印出的文档格式仍然保持原来的版式。

HTML 语言是从 SGML 语言演变而来的，但是它提供更加丰富和方便的布局信息。HTML 被称为标记语言，或简称标记。标记的描述被称为 DTD(Document Type Definition, 文档类型定义)。现在的 HTML DTD 支持基本超媒体文件的建立和版面布置。HTML DTD 有四个版本。HTML 1.0 主要是按照建立超级链接的技术规范而建立的。HTML 2.0 技术规范定义了允许用户显示直插式图像和使用交互式表格的特征。HTML 3.0 支持电子邮件 URL，允许各种列表的任意嵌套，把文本和图形建立成对象。现在 HTML 4.0 已经产生，它在版面控制和人机交互性上又更进了一步。可以预言，HTML 的发展将会朝着使用方便性、页面精美性、交互智能性更进一步的方向发展。

用 HTML 语言编写过调查表单的人们都知道，通过 HTML 调查表单收集信息后，有一个如何进一步处理这些信息的问题。事实上，只要取出调查表单中的输入信息，并将其发送到 Web 服务器上即可，而 CGI 正是实现该目的的标准方法。CGI(Common Gateway Interface, 通用网关接口)是一个用于内部网关程序的标准，所谓网关程序是指与 HTTP 服务器接口的程序。这是一种非常灵活的数据处理方法，它处理发源于 Web 浏览器并被传送到服务器的

资料。而且,这种程序在多个操作系统之间也是容易移植的。CGI 是基于 UNIX 操作系统设计的,因此该标准中的数据传输大多都依赖于“标准入”、“标准出”以及环境变量。

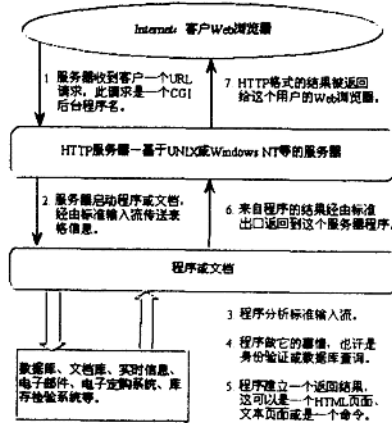


图 1.4 CGI 信息流程

一个表格 HTML 代码通常有如下语句:

```
<FORM ACTION = http://www.yourcompany.com/cgi-bin/handle.exe></FORM>
```

它表示将窗体送至某公司服务器上的 CGI 配置程序进行处理。

有一种常用的 CGI 处理方法,即 CGI 配置程序将信息的处理结果回送到 HTML 窗体,以使用户查看这些结果。通常,CGI 配置程序在回送结果之前,会重复使用用户的输入信息,以校验结果是否正确。为此,CGI 配置程序不但要把用户的信息送交处理,还要格式化其在 HTML 的输出。用户的 Web 浏览器软件接收到 HTML 格式化信息后,将动态地进行信息的显示。关于 CGI 代码的编写,我们在后面的有关 DHTML 与服务器端的数据处理中再作详细介绍。

图 1.4 给出了一个标准的 CGI 接口。

在对 Internet 及 HTML 发展做了简单的回顾之后,让我们看看 HTML 的局限性在哪儿,而 DHTML 则提供了那些改进的特性。

1.2 静态 HTML 存在的问题

静态 HTML 由标记集合组成,这些标记定义了一个页面在浏览器中怎样显示。这些标记以其结构和内容为基础,浏览器会自动对这些标记作最佳译码。人们因此可以利用它来控制行对齐、版面尺寸、图形图像的位置以及图像周围的文本流。但 HTML 仍然不是一门完善的排版语言。在其他用户看来,这种不受控制的 HTML 文档就像一个要提交打印机打印的文件,或者说,当它被加载到浏览器后,它只像一个木偶,不管怎样看它或者即使来回拖动鼠标,页面也不会变化。

当页面加载时,由浏览器来翻译创建页面的 HTML 标记,这些标记会指示浏览器使用何种标题、新的段落怎样放置、什么地址与超级链接相连、将图片放置在哪里等。当浏览器完成这些后,页面就会显示出来,处理过程也就停止,直到浏览器向 Web 服务器发出新的请求。

现在的 HTML 相对于早期有了较大的改进，最新也是最有用的进展叫样式单(Style Sheet)，样式单基本上是 HTML 的一个扩充，它进一步地控制浏览器对元素的解译，从而也就能进一步加强对 Web 页面文本和其他对象版面和位置的控制。然而 Web 页面仍然是静态的，当所有样式法则都被定义和解译后，浏览器就显示这个页面，不会再作其他变化。

静态 HTML 对 Web 上的交互式操作支持有限，仅限于较小且简单的点击接口。由于没有合并 Java Applet 或 ActiveX，所以 Web 接口并没有明显的优势。这样有时就会让一个用户去看一幅他并不想看的图形图像，许多低速链接的 Internet 用户却由于不能快速下载而无法播放图像。

一般说来，静态 HTML 主要在以下三方面存在局限性：文本格式、布局控制和动态改变文件的能力。

1.2.1 文本格式的限制

HTML 的文本格式一直是困扰页面设计者的问题。建立 Web 页面是为了更快更方便地向客户提供信息，而一个设计精美的 Web 页面是吸引广大客户的不可缺少的条件。但是，HTML 没有显式的布局和定位控制，而这些恰恰是设计精美的可视化产品必不可少的东西。

为了扩展文本格式，设计者不得不在进行 Web 设计时玩一些技巧。多数人用单像素控制技术对齐网页上的单元，利用传统的基于 DOS 文本模式的 ASCII 码控制符控制文本的显示；另一些人则用通过表格设计一些网格缩进的方法。这两种方法都很繁琐且收效甚微。

更令人难堪的是，设计者无法控制 WWW 的字体。通常情况下，能用的仅是粗体(B)和斜体(I)，而这些正是用户使用浏览器浏览 Web 页必不可少的东西。没有了这种控制，各种浏览器会对 Web 页的显示存在差别。如果设计者再使用类似表格、缩进的技巧创建布局，最终将会使页面格式变得混乱不堪。

1.2.2 布局控制的局限

在格式方面，静态 HTML 使设计者无法控制文本内容的定位和布局。虽然 HTML 可以很方便地组合文本和图形，但想让它具备报纸、杂志那样图像定位及文本环绕图像的布局几乎是不可能的。当然，设计者可以像设计房屋结构图那样，不厌其烦地利用测量坐标和若干 <TABLE> 标志使 Web 页面具有某些布局上的特征。显然，这一过程不适用于 Web 使用方便性的要求。因为，牺牲了许多布局单元，也就阻止了打印、复制和 Web 独有接口的创建，格式也就变得古板。尽管表格和帧的应用使用户有能力设计出更优美的布局，但由于应用程序的缺乏，人们无法选择合适的资料显示格式和框架。

1.2.3 动态内容无法改变

如前所述，静态 HTML 工作时，浏览器将查阅整个文档，并把它翻译和显示出来，整个工作过程就像用显示指令来填充一个文本文件。更有甚者，静态 HTML 的 Web 页的布局无法动态地随浏览器的改变而改变。例如，当用户在浏览时改变窗口的大小，文本会被刷新，但图像的大小并不会随之改变。

虽然静态 HTML 也存在一些动态内容，但它们都是被动的，没有交互性或者交互性有限。例如，动画 GIF 文件可以提供页上的动画，但必须事先在静态区指定对象的移动路径，

这必然使开发用户接口和布局控制更加复杂。一些窗体或图像映像也可以动态改变,但是它必须与服务器多次交互或操作,通过服务器端的 CGI 进行编排,这样网络的传输速度就会成为交互性性能的瓶颈。

通过这些较为枯燥的叙述,我们初步了解了静态 HTML 不能做什么。下面,我们通过实例,看看 DHTML 较静态 HTML 有哪些改进。

1.3 DHTML 特性

DHTML 在 W3C 有它自己的定义,但在实际应用中描绘成是 HTML、样式单和脚本的组合物。由于微软和网景这两家大公司对 DHTML 的理解存在差异,W3C 的仲裁角色说明脚本设计语言是用中立方法解决的,它不是某一个软件开发商的产品。

抛开以上这些问题,现在我们应该仔细介绍 DHTML 的一些基本特征了,这些特征是开发 DHTML 标准的不同公司所一致认可的。

1.3.1 文档对象模型

我们称 DHTML 查阅文档的方法为 DOM(Document Object Model,文档对象模型)。DOM 的工作原理是这样的,虽然页面仍然像文本那样,但浏览器处理的方法却不同。当浏览器遇到页面上的各个元素时,会保存其位置,并将相似的元素放在堆栈里。这样,它就能跟踪页面上从<TITLE>标题到窗体的每个元素。从本质上说,每当页面被加载时,浏览器就会建立起一个相关的数据库,每个标记就成为数据库里的一个记录。

1.3.2 脚本语言支持

DHTML 是用脚本语言(Script)控制文档中所有需要控制的元素的。脚本是一种类似于“宏”的特殊程序语言,它嵌在 Web 页内,用以控制页面上的不同元素,例如窗体元素、图形图像、帧等。HTML 使用的脚本语言基本有两大类——JavaScript 和 VBScript(Visual Basic Scripting Edition)。本书使用 JavaScript。

文档对象模型和脚本语言支持这两个特征非常重要,它使 Web 页面生动有趣,并且具有计算机程序所具有的交互性特征。

在进行 Web 页制作时,设计者必须完成一些基本任务,如:指明字体、定位图像、格式化文本格式。我们看看 DHTML 是怎样解决这些问题的。

1.3.3 可改变文本的格式

前面谈到,静态 HTML 难以扩展文本格式。在 DHTML 中,可以指定浏览器所用的字体,Windows 或 Macintosh 操作系统会自动装载这些字体,并用于设计者所设计的 Web 页。设计者可以指定一种最接近的字体,还可以方便地实现布局。图 1.5 就是 Netscape 公司关于 DHTML 的示例。

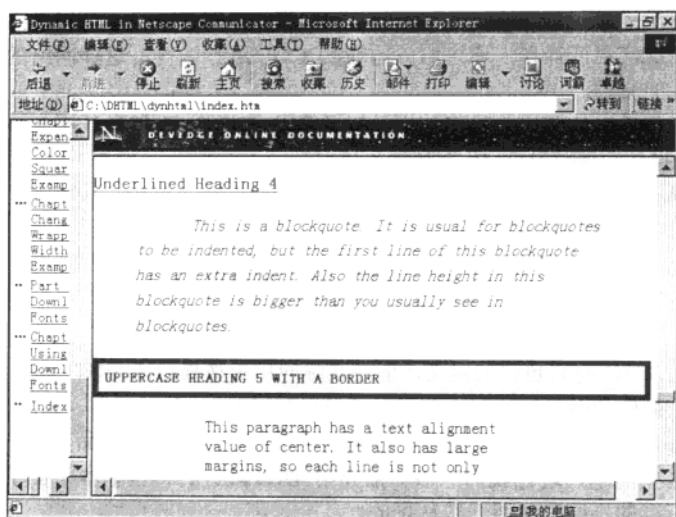


图 1.5 Netscape 的 DHTML 文本格式示例

从图中可以清楚地看出，我们可以设定以块为单元的元素边界并用以填充显示元素之间的距离；所有<P>元素都具有左右边界，而文字处于居中的位置；所有<H4>元素都是绿色且有下划线；所有<H5>元素都是大写且具有 4pt 厚度的 3D 浮雕风格的红色边界，其中文本的背景为黄色、前景为红色；所有的<BLOCKQUOTE>元素为蓝色斜体，行的高度比正常字体要大 50%。

DHTML 提供了一种独特的文本格式，可以在线改变文本。这对于创建嵌入式高亮度文本是很有用的。它可以在改变 DHTML 文档颜色、大小、字体一段时间后，或是当用户再次将鼠标移近文本时加载主页。图 1.6 是在线改变文本的简单示例。在用户鼠标未经过显示的文本时，显示的是蓝色的有底线字体，在用户鼠标经过显示字体后，显示的是红色的罗马字体。

使用此技术，我们就可以更改目前的链接页或创建新的 Web 接口，随着时间的推移页面可能会出现多次链接，每三秒显示一次新链接。通过提供用以替换 Web 页的内容，我们可以实现有许多文本观察器的 Web 上的交互式动态游戏。例如，在挖雷游戏中，我们可以根据用户挖雷的情况，动态地显示剩下雷数的多少。

动态内容或页面文本操作能力都可以在 HTML 内容加载后的页面中更改，所有这些都因 DHTML 的出现而产生革命性的效果。尽管 HTML 自从一开始就不断改动，但没有哪一次像“动态内容”产生如此大的影响。

1.3.4 可下载字体

Netscape 公司提出了可下载字体的概念，这种字体可以被其他浏览器所接受。用过静态 HTML 中的样式单和命令的用户可能已经感受过局限，因为你指定的字体在其他机器上不一定适用，从而造成页面开发者不知道自己的页面的最终显示结果。

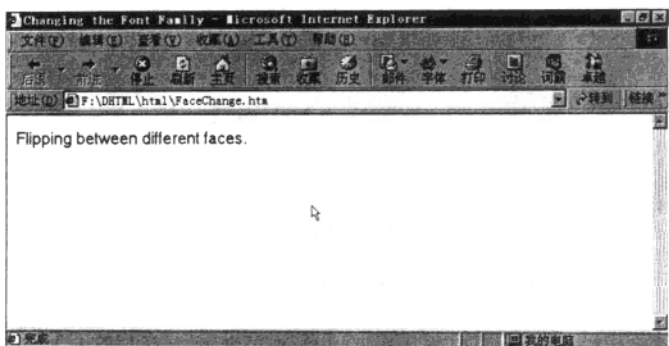


图 1.6 鼠标经过文本前的文本格式

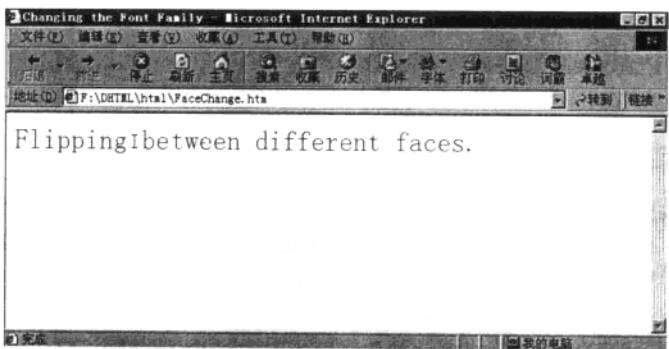


图 1.7 鼠标经过文本时显示的文本格式

通过提供一个有一定优先级的字体清单,再跟上诸如无底线字体(Sans-Serif)等通用型字体,就可以用样式单或标记解决这个问题,但这种方法并不是很好。Netscape 采用的方法是用由 BitStream 公司开发的 TrueDoc 技术,让字体信息以高度压缩的格式与页面一起下载,字体的详细信息以向量格式存放,利用保存在用户机器里的公用信息库可以快速地 将字体组装起来。但是,用户并不能将这种字体拷贝下来以供将来使用,因为这种字体仅仅 在当前页贮留在浏览器缓存内时才存在于用户终端。图 1.8 是一些 TrueDoc 字体格式的定义。关于 TrueDoc 字体格式的信息可以在如下网址找到:

<http://home.netscape.com/comprod/products/communicator/fonts/index.html>

<http://www.bitstream.com/world/>

所谓字符的向量格式就是指单个字符以图像格式显示。这个图像由一系列形状组成,其中 包括所有的直线、曲线以及其他一些形状特征。利用这些形状特征形成的字符所占的存取 空间要比普通的位图图像所占的空间小得多。