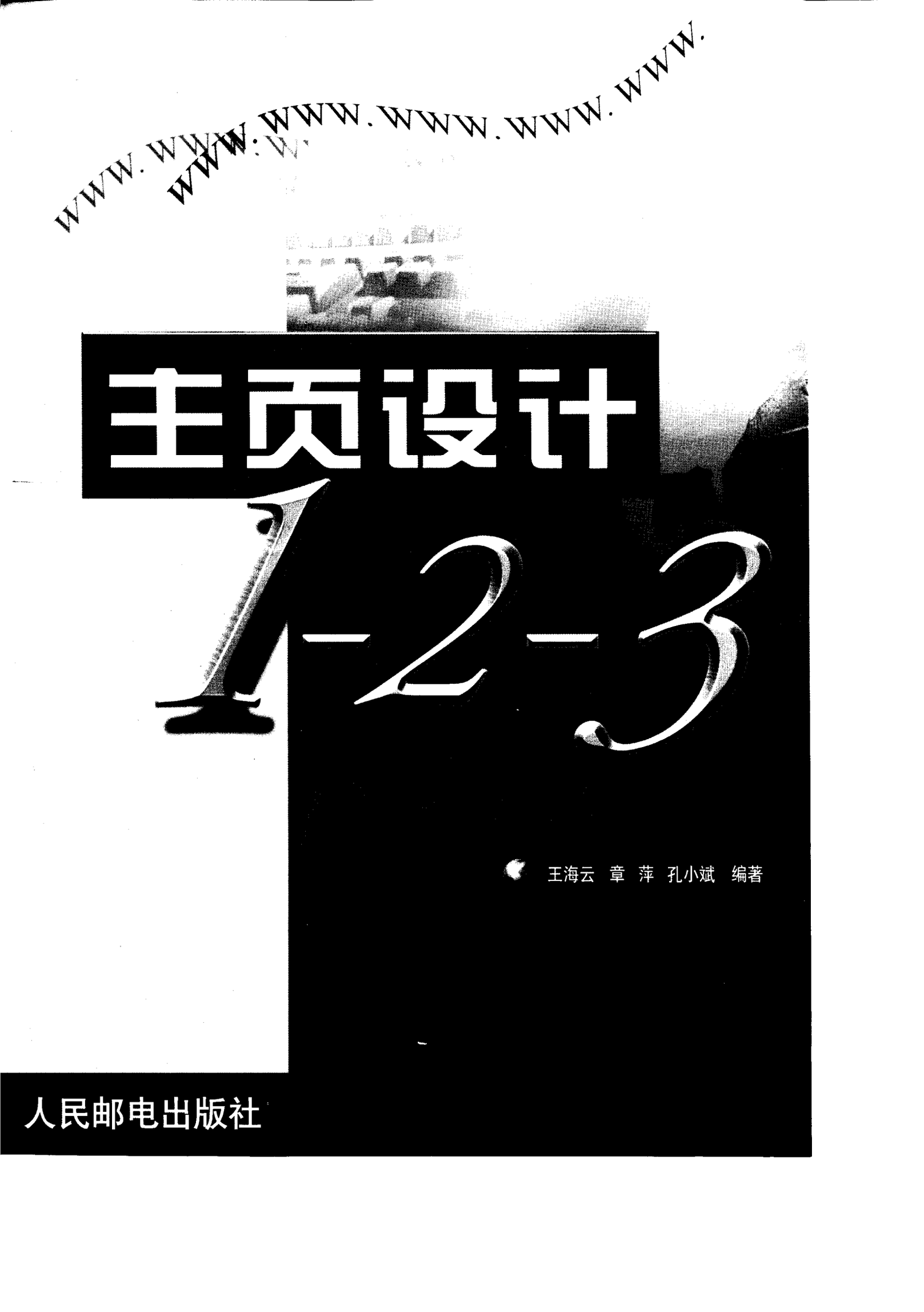
www.www.www.www.www.www.
主页设计
1-2-3
王海云 章 萍 孔小斌 编著
人民邮电出版社

主页设计 1-2-3

编 程 篇

王海云　章萍　孔小斌　编著

人 民 邮 电 出 版 社

内 容 提 要

本书是系列图书《主页设计 1-2-3》的第二本——《编程篇》，讲述了动态 Web 站点的设计原理和主要技术，结合实例，深入细致地阐述了动态站点的概念、基本特点以及 CGI、ISAPI、Java、JavaScript、VBScript、Active X、IDC、ASP（Active Server Page）等各种实现动态站点的技术，并且提供了大量的直接可以应用于站点设计的源代码，内容丰富，实用性强，适合设计动态站点的初学者和专业人员参考使用。

主页设计 1-2-3

编 程 篇

◆ 编 著 王海云 章 萍 孔小斌

责任编辑 陈万寿

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092 1/16

印张：14.5

字数：354 千字 1999 年 6 月第 1 版

印数：1-6 000 册 1999 年 6 月北京第 1 次印刷

ISBN 7-115-07679-0/TP·1031

定价：22.00 元

前　言

随着 WWW 应用领域的不断扩展，站点规模的日渐扩大，传统的静态 Web 站点已经越来越不能够满足人们对信息服务的需要。对于浏览者来说，他们已经不再满足于简单的浏览页面获取信息，他们更希望与 Web 站点交互，完成各种数据的处理工作。对于站点设计和维护者来说，由于站点规模的不断扩大，更多的信息要求上网，传统的手工设计静态页面已经无法满足这一需求。开发动态的、交互的、易于维护的站点将是今后站点设计的发展方向之一。

实现站点的动态性和交互性的技术非常多，包括 Java、Java Script、CGI、ISAPI、Active X VBScript、IDC、ASP(Active Server Page)等多种，各有所长，适合于不同场合使用。本书作者在参考国内外大量资料的基础上，结合自己的理解和经验编写了本书，对实现动态站点的各种技术进行了详细的介绍和比较，希望给那些将要或者正在设计动态站点的朋友们提供一些有益的参考。

本书是系列图书《主页设计 1-2-3》的第二本——《编程篇》，共分 10 章，内容涉及实现动态站点的各种流行技术。第一章主要介绍动态站点的主要概念、实现动态站点的主要技术、动态站点和数据库技术，第二章主要介绍使用 FrontPage 增加站点的动态特性和交互性，第三章至第九章分别介绍了 Java、JavaScript、CGI、ISAPI、IDC、ASP、Active X 和 VBScript、ASP(Active Server Page)等各种实现动态站点的技术和编程实例，最后一章介绍了兼容性站点的设计的主要技术。

本书第一、二、五、六、十章由王海云编写，第三、四章由章萍编写，第七、八、九章由孔小斌编写。另外参加本书编写工作的还有陈万寿、孔小凯、李庄、孔穗虹和孙衍、陆庆杭、邓玉龙等。

由于编者水平有限，有许多实用的动态站点技术书中都没有提及，错误之处也在所难免，恳请读者批评指正。有关编辑出版方面的问题，读者可上网咨询。

E-mail: commedit@public3.bta.net.cn　chenwanshou@263.net
http: //www.pptph.com.cn/

作　者
1999 年 4 月

目　　录

第一章　Web 站点的动态性和交互性

随着 Internet 的不断发展，传统的 Web 站点已经逐渐不能满足浏览者的需求。越来越多的 Web 站点正在由静态站点向动态和交互的站点转变。WWW 的动态性和交互能力随着一系列新技术的推出而不断增强。作为一个 Web 设计者，除了掌握必要的主页制作技术之外，更有必要了解动态和交互式站点的制作方法。在介绍 Web 站点的动态性和交互性之前，先让我们了解一下 WWW 的工作原理。

1.1　World Wide Web工作原理

World Wide Web 是使用 Internet 作为传输媒质的一个应用系统。WWW 上传输的基本单元是 Web 页面。一个 Web 页面包含文本、图像、声音、视频片断等。此外，Web 页面一个重要的特性就是它包含超链接。超链接改变了人们搜索、接收信息的线性模式，具有更好的交互性，更有利于信息的组织和表达。

Internet 不是单一的一个网络，它是一组全球网络的总体。WWW 信息就分布在这些网络的主机上并通过超链接联系起来。客户端的 Web 浏览器通过超文本传输协议（HTTP）从主机上运行的 Web 服务器软件（如 Netscape Enterprise Server 或 Internet Information Server 等）获取信息并与之通信。Web 服务器和浏览器之间的工作方式可以看作是一种客户/服务器模式。

下面我们看一下 Web 服务器和客户端浏览器之间的交互过程：

1. 用户在浏览器的地址栏中输入一个 WWW 地址，比如：http://www.microsoft.com。Web 浏览器根据用户输入的主机域名，通过域名服务器（DNS）解析出该主机的 IP 地址，然后向该 Web 服务器（如 www.microsoft.com）发起 Web 页面请求。

2. Web 服务器接收 Web 页面请求，找到相应的页面，并通过 HTTP 协议将这个页面传送给 Web 浏览器。

3. Web 浏览器接收页面并解释 HTML 语言，然后在窗口中显示页面内容。

首先，Web 浏览器发起一个页面请求。页面请求通常有两种方式：HTTP GET 请求或 HTTP POST 请求。GET 请求用于获取静态页面、图形或其他 Web 元素（如声音、视频、Java Applets 等）。但是，Web 浏览器也有可能请求 Web 服务器运行一个程序，来动态生成 Web 页面数据。比如，在 Yahoo 中输入一个关键字查询某一类站点，当用户按下查询按钮时浏览器就会把用户输入的关键字传送给 Web 服务器，并请求服务器上的搜索程序对这个数据进行处理（根据关键字查找），这些程序通常被称为公共网关接口程序（CGI 程序）。Web 浏览器

通过 HTTP POST 来请求 Web 服务器运行这些 CGI 程序。在 POST 请求中包含了传送给服务器程序的数据，在这个例子里传送的就是用户输入的关键字。

Web 页面请求通过 Internet 传送到相应的服务器。Internet 通过 Web 页面请求中的 URL（统一资源定位符）来判断请求应当发送到哪一台服务器上、哪一个页面应该被传输。

Web 服务器根据 URL 寻找相应的文档。如果找到了，就把文档传送给 Web 浏览器。为了帮助浏览器理解所传内容的类型，服务器还要传送一个内容类型头。这个头对于后面要介绍的 CGI 和 ISAPI 应用程序是非常重要的。因为使用这两种接口时，要自行提供这个头的内容。

现在 Web 浏览器接收文件，并一边接收一边对页面中的 HTML 代码进行解释，然后显示在浏览器窗口中。HTML 代码一方面告诉浏览器如何显示内容，另一方面告诉浏览器如何获取页面上的其他元素。比如，页面中包含的这一语句：<IMG SRC="\images\logo.gif"> 提示 Web 浏览器从当前 Web 目录下的 images 子目录中提取 logo.gif 图像。浏览器根据这些指示找到并下载全部页面元素，并显示在浏览器窗口中，这样就完成了一个页面的 接收和显示过程。

1.2　走向动态和交互的Web站点

早期的 Web 页面内容都是静态的。Web 设计者将设计好的页面放在 Web 服务器上，Web 浏览器获取这些页面并显示。浏览页面时，只能从页面上获取信息，而不能和页面进行交互。这就出现了一个问题：两端的计算能力没有得到充分的发挥，Internet 仅仅局限于信息的存放和展示而没有提供处理数据的能力。

这样的 Web 站点有两个问题。以一个某家公司的产品展示的 Web 站点为例，首先，对于这样一个站点，浏览者就只能通过网站了解一些产品的信息，但是却不能直接通过这个站点买到自己所需要的产品。这样的站点是无法让公司和它的客户满意的。

其次，如果这家公司提供的产品的种类和价格是在不断变化的（大部分公司的产品正是这样的），那么必须有专门的人员来完成随时更新所有相关的 HTML 页面的任务。这一任务工作量非常巨大而且容易出错。

为了解决这两个问题，人们提出并开发出了 Web 的交互性技术和动态页面技术。

Web 的动态性和交互性包含两层含义：

一是基于浏览器本身支持的动态和交互特性，由 Web 页面提供的动态性和交互性。这种动态性和交互性主要是发挥了客户端的计算能力，我们称之为客户端的动态性和交互性。

二是基于 Web 服务器的 Web 页面交互生成和动态发布技术。这种交互性和动态性扩展了 Web 服务器作为 HTTP 服务器传送 HTML 以及相关文件以外的功能，充分挖掘和发挥了服务器端的计算能力，我们称之为服务器端的动态性和交互性，或者称之为动态 Web 站点。

1.2.1　客户端的动态性和交互性

Web 浏览器本身支持许多客户端的动态和交互性技术，这些技术包括：

1. GIF89a动画

GIF89a 动画是增加浏览器显示页面时的动态性的最简单和费用最低的方法。如果浏览器支持 GIF89a 动画，它就会从下载的 GIF89a 动画文件中读取文件中包含的各个图象帧和播放参数，按照 GIF 动画设计时的效果循环显示 GIF 动画中的各个图像帧，形成一种最简单的动画效果。

在《主页设计 1-2-3》（入门篇）中我们已经详细介绍了 GIF 动画的使用方法和注意点，这里不再赘述。

2. Meta标签

前面我们已经介绍了使用 Meta 标签为页面指定语言文字属性以及供搜索站点的机器人使用的关键字。其实 Meta 标签还可以用来实现一些简单的动态效果：

(1) 强制刷新页面

如果要求某页面在每次下载时都重新从 Web 服务器读取，可以在主页 HTML 源文件开头<head>和</head>标签之间加入：

```
<meta HTTP-EQUIV="Pragma"  CONTENT="no-cache">
```

这样，浏览器将忽略本地页面缓存中的内容而强制读入用户的页面，相当于用户按了 Reload 命令。这种语句适合在 BBS 等经常需要更新的页面中使用。由于 BBS 的页面内容是在随时更新的，使用这一语句可以确保用户读到的页面版本始终是最新的。

(2) 控制页面缓冲时间

下面的语句：

```
<meta HTTP-EQUIV="expires"  CONTENT="TUE,23.NOV 1999 12:00 GMT">
```

用来设置网页到期的时间。如果在浏览器中设置浏览网页时首先查看本地缓冲里面的页面，那么浏览某一个网页时，浏览器会自动检查缓冲区中的页面是否存在，若存在则首先浏览本地缓冲区的页面，直到 meta 中设置的时间到期时浏览器才会去服务器上下载新的页面。

如果你设计的某些页面是定期更新的，那么你就可以将页面中的时间设置为下次你要更新页面的时间。通常我们可以自己编写一些小程序来自动为页面加入这一语句。

(3) 定时刷新页面

```
<meta HTTP-EQUIV="refresh" content="15;URL=http://你的 URL">
```

这是 meta 最常见的用法。当我们浏览一些页面时，常常会发现某些页面过一段时间之后会自动转到另外一个页面，这就是这一语句在起刷新作用。

利用这一语句，可以制作出简单的广告牌效果：先显示一个具有站点标志或其他会给

人留下深刻印象画面的简单页面，过一段时间之后自动转向一个显示主要内容的页面。这一语句还经常用来提示站点位置已经移动。比如说，你的个人主页搬家之后，为了让用户访问你的新主页，你可以在原来主页位置留下一个页面，在页面中加入一个简短的文字提示和指向你的新主页位置的链接，同时加入以上的 meta 语句，在 meta 语句的 URL 属性中指定新主页的位置。这样，当用户浏览你原来的主页位置时，他可以选择页面上的链接访问你的新主页，或者等一段时间之后让浏览器自动连接到新的页面上去。页面源码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta http-equiv="Refresh" content="4; URL=http://vcdynasty.yeah.net">
<title>主页已搬迁</title>
</head>
<body bgcolor="#060147" text="#FFFFFF" link="#FFFFFF" vlink="#008080" alink="#0080FF">
<p align="center"><br>
我的主页已经搬家，请点击下面这个链接或者等待浏览器自动把你带到新的主页位置。<br>
</p>
<div align="center"><center>
<table border="2" cellpadding="9" cellspacing="3" bordercolor="#808080"
bordercolorlight="#FFFFFF">
<tr><td align="center" rowspan="4">
<a href="http://vcdynasty.yeah.net">http://vcdynasty.yeah.net</a></td>
  </tr></table></center></div>
<p align="center"><br>
<br>
</p>
</body>
</html>
```

显然，利用这个定时刷新页面的 meta 语句，我们还可以制作出简单的幻灯片效果。语句类似上面，但是在页面 1 中让 URL 指向页面 2，而页面 2 指向页面 3……，最后一个页面指向页面 1，如此可以实现简单的循环显示页面的效果。

3. JavaScript技术

JavaScript 是一种内嵌在页面的 HTML 语言中并且由浏览器解释执行的脚本语言。现在许多页面都使用 JavaScript 来增强页面的动态性和交互性。

使用 JavaScript，可以在页面中动态显示当前时间、日期和滚动文字、最近更新日期；在状态栏上显示跑马灯效果的流动文字；对用户在链接上的鼠标移动做交互式的响应，比如在状态栏上显示有关链接的提示或者切换图像（也就是目前页面上非常流行的动态菜单）……JavaScript 还可以用来对用户由表单输入的数据进行有效性检查，防止浏览器将错

误的数据提交给 Web 服务器去处理，减少 Web 服务器不必要的负担。总之，利用 JavaScript 可以设计出各种精彩的动态和交互式页面效果。有关 JavaScript 我们将在后续章节中作详细介绍。

4. 基于表单和公共网关接口（Common Gateway Interface：CGI）的交互性

HTML 语言本身支持表单，表单包括单行文本框、多行文本框、检查框等多种类型的字段，用于接受用户的数据输入。通过表单，用户可以在页面中输入数据，比如在使用 Yahoo 搜索引擎时输入关键字，浏览器会自动将用户输入的数据发送到相应的 Web 服务器，Web 服务器会根据表单中指定的数据处理程序，通过一个接口（常见的是公共网关接口）将数据交由一个相应的应用程序去处理。应用程序处理完后将返回数据以 HTML 页面形式返回给 Web 服务器，由 Web 服务器传送给客户端浏览器去显示。

公共网关接口的提出为静态页面内容模型增加了一定的动态性和交互性。在这种接口下，用户通过表单向 Web 服务器发送基于某种应用的数据，比如调查表数据，Web 服务器获得该数据之后，通过 CGI 接口传送给存放在 Web 服务器上的应用程序。应用程序对这些数据进行处理并将处理结果以 HTML 页面形式通过 CGI 接口返回给 Web 服务器，Web 服务器再把动态生成的 HTML 页面发送给 Web 浏览器，工作模型如图 1.1 所示。

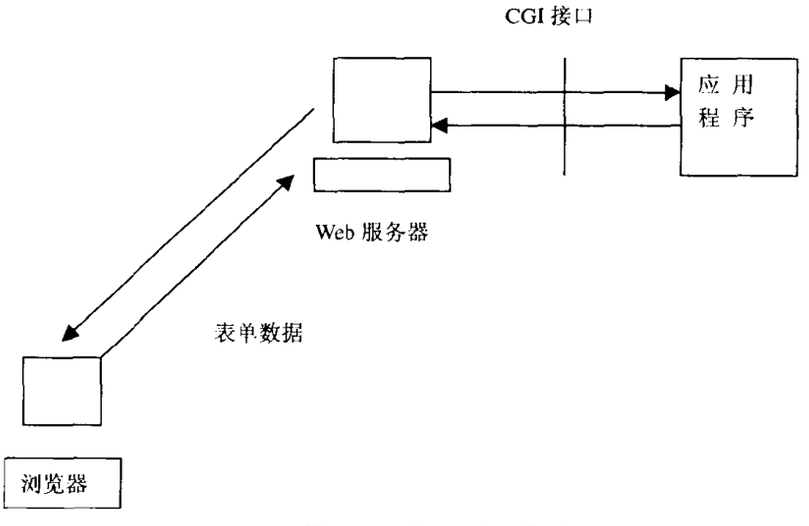

图 1.1　公共网关接口模型

在这种模型之下，服务器端的计算能力得到了有效的发挥。但是 Web 页面内容本身仍然是静态的，页面一旦载入浏览器之后，内容不会有任何改变，也不具备交互性。因此，客户端的计算能力仍然没有充分利用起来。为了解决这个问题，人们首先想到使用外部应用程序来增强 Web 内容的多样性。

5. 外部应用程序

在下载一个文档时，多数浏览器都可以直接调用与文档相对应的外部程序，与文档相关的应用程序在浏览器的选项中注册指定。比如，当我们下载一个 Word 文档时，浏览器会提示我们是否用本地的应用程序（如 Microsoft Word）来打开它或仅仅下载文件。播放 QuickTime 视频等也会启动外部应用程序。当浏览器下载一个未在浏览器中指定相应应用程

序的文档时，浏览器会提示用户选择一个外部应用程序。这样，如果 Web 设计者在页面中使用了一个 Word 文档，这个文档并不能直接在页面中显示，而必须调用相应的外部程序来显示，如 Microsoft Word。

这种方式在一定程度上增强了 Web 页面的动态特性。从客户/服务器角度来看，外部程序分担了一部分客户端的工作，增强了客户端的处理能力。但是，这种模式也有两个明显的缺陷：

- 用户必须知道使用什么程序来显示文档的内容，还要对浏览器进行配置，保证下载主页上的文档时能够启动相应的应用程序。
- 为了确保浏览器一端能够正确显示文档内容，Web 设计者需要分发相应的应用程序。

尽管安装外部程序已经相当简单，但是由于 WWW 上文档的多样性，用户还是不愿意仅仅为了显示某一个文档而安装一个应用程序。这样，就限制了外部应用程序这种模式的推广。从 Web 设计者角度来说，为什么要给一个使用者很少的应用程序设计文档（内容）？从用户角度来看，又何必仅仅为了显示一个文档而安装很少使用的一个应用程序？这样的结果是：外部应用程序用得越来越少。因此，这种方式仍然无法令人满意。

6. 插件

比外部应用程序更加方便和常用的技术是插件。插件是集成到浏览器中的辅助应用程序模块。下载插件所能够处理的内容时，浏览器通过标准的 API 接口直接调用插件模块，而不是启动应用程序，来处理这些内容。这使得浏览器具有更好的集成性，更易于使用。我们所熟悉的 Netscape Live Audio 实际上就是一个插件。当 Netscape 下载一个声音文件（如 Wave 文件或者 MIDI 文件）后，就会自动启动 Live Audio 插件去播放声音文件。我们在页面上见到的许多效果都是通过插件来实现的。常见的插件还有 Flash3、Real Audio、Real Video、Acrobat 插件等。

7. 客户端拉和服务器推技术

增加内容的动态性的方法还有客户端拉和服务器推技术。

客户端拉技术使 Web 设计者可以自动向用户提供一系列的页面。为了实现客户端拉技术，可以在页面中使用一个前面介绍的<META>标记。<META>标记告诉浏览器在一段时间之后从服务器重新拉当前页面或拉另一个页面。这样，就可以定时更新当前页面的内容或者制作页面自动切换、循环播放的幻灯显示效果。Netscape 和 Microsoft 都支持客户端拉特性。

服务器推技术是另一种增加页面动态性的方法。在这种技术中，服务器按用户订阅要求，主动地把不断更新的内容（如新闻、股市行情）等发送给客户端浏览器。这种模式下的页面动态性更好。这方面最著名的技术是 Microsoft 公司的 Active Channel 技术。但是无论是客户端拉还是服务器推技术，用户端浏览的页面本身仍然是静态的。

8. 动态可执行内容：Java和ActiveX技术

目前真正实现动态内容页面的技术有两种：Sun 公司的 Java 技术和 Microsoft ActiveX 技术。它们有两个共同之处：

- Web 浏览器使用客户端的计算能力使得页面动起来。比如，两者都可以制作在客户端执行的动画程序，从而在页面中显示动画效果。
- Web 浏览器知道如何调用合适的软件模块使页面动起来。

Sun 公司率先提出可执行内容的概念，也就是内容知道如何显示自己。

在静态模型之中，Web 浏览器、内容、显示内容的程序三者是分离开来的。如图 1.2 所示，每个对象都有自己明确的边界。在动态模型中，内容知道如何使自己动起来，也就是知道如何显示自己。Web 浏览器了解可执行内容的接口，也就是知道如何使内容动起来。这时候，浏览器、内容以及显示内容的程序三者是一个紧密的整体。

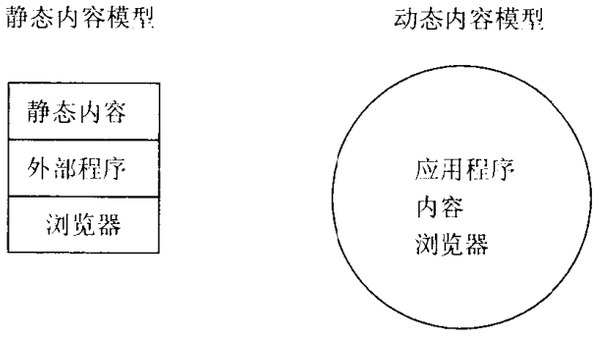

图 1.2 静态内容模型和动态内容模型

可执行内容解决了静态模型的两个缺陷。首先，由于下载的对象知道如何显示自己，因此浏览器也不必关心为这种数据提供什么样的应用程序。对于 Web 设计者来说，也不必担心用户端是否安装了显示相应内容的模块，因为在发布内容的同时已经把相应的应用程序发送给了用户。

HotJava 是第一个支持动态内容的浏览器。但是，由于浏览器的市场已经为 Netscape 所占领，它并没有取得商业上的成功。第一个真正使得动态内容成为实践的浏览器是 Netscape Navigator 2.0。Netscape Navigator 通过下面三个部件来支持动态内容，它们是：Java applets，Java Script 和插件（plug-ins）。

Microsoft 为了将其在桌面软件上的霸主地位延伸到 Internet，不甘落后，迅速推出了它的动态内容方案：Active X 技术＋Microsoft Internet Explorer 3.0。

Java Applets 是可下载的可执行内容。一个 Java Applets 不仅包含内容，也包含如何使自己的页面动起来的指令。比如，页面中的一个时钟 Java Applet 不仅包含时钟图像数据，也知道如何根据当前时间更新显示。Java Applets 可以响应用户的鼠标和键盘输入，也可以同浏览器交互，比如说，改变浏览器状态条的内容。

Active X 也提供了类似的可下载并执行的内容机制。这些可执行的内容被称作 ActiveX 控件，它是 OLE 2.0 对象在 Internet 上的延伸。与 Java Applets 不同的是，Active X 控件不仅可以在浏览器中使用，也可以在应用程序中使用。这使得用户桌面和 Internet 网络自然地融合为一个整体。

Web 设计者还可以在页面的 HTML 代码中直接使用 Java Script 脚本语言。浏览器下载嵌入到页面中的脚本语言，并对它进行解释执行。这使得 Web 设计者可以处理文本、列表、按钮等的输入，比如作一些判断、数据合法性检查等。Active X 提供类似的脚本 VBScript，

它是著名的编程语言 Visual Basic 的一个子集，它可以完成类似 Java Script 的功能。

1.2.2　动态 Web 站点

第一代 Web 站点可以称之为静态 Web 站点，它是由静态 Web 页面组成的。静态的含义是页面不是由服务器上的软件运行时产生的。

静态页面通常由页面制作者设计并上载到 Web 服务器上，直到 Web 设计者修改页面之前，页面内容是固定的。Web 设计这修改页面并重新上载。上载完之后的页面仍然是静态的。当服务器接收到来自 Web 浏览器的 HTTP 请求后，发送一个和它创建时完全相同的文件来响应用户。用户接收到的信息都是事先写好的以 HTML 格式存放的文件。传统的静态站点的服务器所做的工作只是将使用者要求的 HTML 文件传送到浏览器而已。

静态 Web 站点存在着两个问题：

一是只适合于产品广告、产品手册、学术资料、网上公告等简单文本信息的发布，不适合于海量数据库的发布。

二是服务器的计算能力没有得到充分发挥，没有交互性，因此无法实现网上购物、网上数据库查询等需要用户交互的服务。

新一代的 Web 站点是动态的，它是由动态 Web 页面构成的。

所谓动态的 Web 站点，指的是：

(1) 内容会变：网页会根据不同时间、不同使用者，而提供不同的内容。

(2) 自动更新：网页会自动更新，而不需要人工去直接修改页面并上传。

(3) 交互性：网页内容会依照用户输入的内容、选择而改变。

动态 Web 页面是指用户向服务器发出 HTTP 请求后，Web 服务器根据用户请求临时组织页面内容。

动态页面可以用多种方式生成：

(1) 使用一个静态 HTML 模板，用服务器上的软件在运行的时刻修改这个静态页面的内容，比如提供每天的日期信息。

(2) 根据数据库，由软件生成 HTML 页面。比如说，一个 Web 用户查询产品数据库中的某一产品的价格。Web 服务器上的应用程序将根据用户输入的关键字查询数据库，并生成动态查询结果返回给 Web 服务器，再由服务器返回给用户。

(3) 服务器上的软件可以自动检测 Web 用户的浏览器特性，并定制页面，使得页面在各种浏览器中都具有最佳效果。

(4) 用 Web 用户发布的信息来定制 HTML 输出。比如，一个用户选择了多种选项之后，Internet 服务器应用程序就可以生成相应的页面来满足用户的需求，典型的例子有 My Yahoo 和 My Netscape。

与静态网站相比，动态网站的优点是十分明显的：

(1) 站点维护更加容易：可以使用数据库管理整个网站，Web 服务器和应用程序可以自动访问数据库并提取信息生成页面。这样，只要更新数据库的内容，网站的内容就会自动更新。

比如，前面提到的公司的例子，如果使用了动态网站技术，就可以把产品信息存放到数据库里。当用户访问这个站点时，Web 服务器自动从数据库中取出产品信息并显示在页