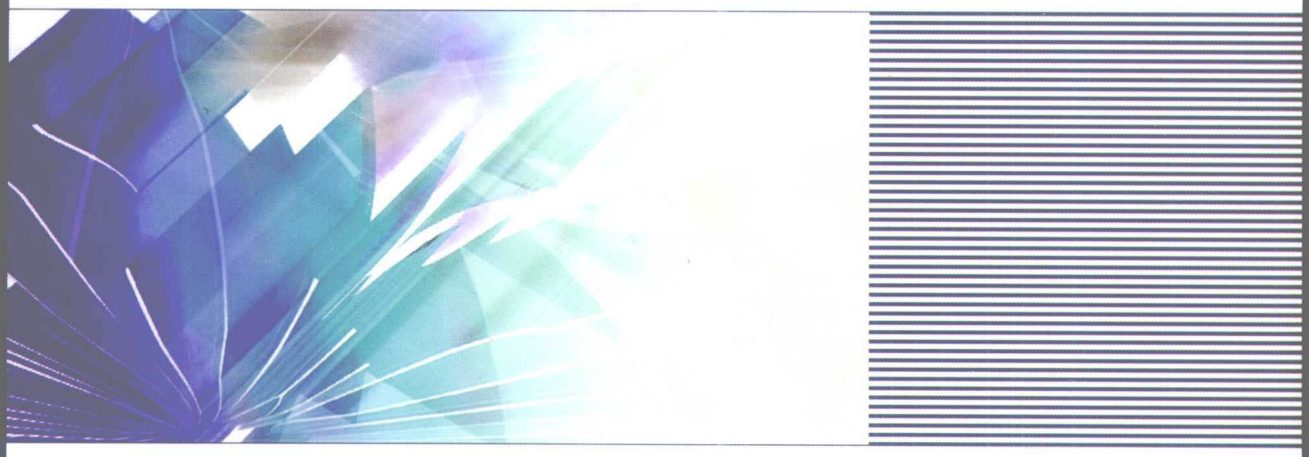
21世纪高等院校网络工程规划教材

21st Century University Planned Textbooks of Network Engineering



ASP.NET 4.0 Web 程序设计

ASP.NET 4.0 Web Applications

刘艳丽 张恒 编著

- 作者多年教学科研成果结晶总结
- 站在实用角度深入分析技术要点
- 精选大量例题并且增加汉字注释

人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等院校网络工程规划教材

21st Century University Planned Textbooks of Network Engineering

ASP.NET 4.0 Web 程序设计

ASP.NET 4.0 Web Applications

刘艳丽 张恒 编著

人 民 邮 电 出 版 社

北 京

图书在版编目（CIP）数据

ASP.NET 4.0 Web程序设计 / 刘艳丽，张恒编著. -- 2版. -- 北京 : 人民邮电出版社，2012.12
21世纪高等院校网络工程规划教材
ISBN 978-7-115-29834-8

Ⅰ. ①A… Ⅱ. ①刘… ②张… Ⅲ. ①网页制作工具－程序设计－高等学校－教材 Ⅳ. ①TP393.092

中国版本图书馆CIP数据核字(2012)第308237号

内 容 提 要

本书以通俗的语言、丰富的实例，详细介绍了ASP.NET 4.0网站开发技术。全书共分为17章，主要内容包括：Web程序设计概述、HTML和CSS、JavaScript编程基础、C#语言基础、ASP.NET Web开发基础、ASP.NET对象及状态管理、ASP.NET 4.0服务器控件、ADO.NET数据访问、数据绑定技术与绑定控件、ASP.NET网页布局与标准化、ASP.NET应用程序安全技术、LINQ与AJAX新技术等。此外，每章都有配套的实验，让读者寻找编程感觉，培养编程思想。

本书结构合理、条理清晰、实例丰富，图文对照，可以作为高等院校计算机科学与技术、网络工程、软件工程等相关专业ASP.NET课程的教材，也可供从事Web程序设计相关工作的技术人员自学参考。

本书的电子教案、示例源代码可以到人民邮电出版社教学资源与服务网上免费下载，网址为http://www.ptpedu.com.cn/。

21世纪高等院校网络工程规划教材

ASP.NET 4.0 Web程序设计

◆ 编　　著　刘艳丽　张　恒
责任编辑　刘　博
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷
◆ 开本：787×1092　1/16
印张：20.75　　2012年12月第1版
字数：541千字　　2012年12月北京第1次印刷

ISBN 978-7-115-29834-8

定价：42.00元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

ASP.NET 是 Microsoft 公司创建服务器端 Web 应用程序的新一代技术，它构建在 Microsoft.NET Framework 基础之上，.NET Framework 聚合了紧密相关的多种新技术，彻底改变了从数据库访问到分布式应用程序的一切。而 ASP.NET 是.NET Framework 中最重要的部件之一，通过它用户可以开发出高性能的 Web 应用程序。

ASP.NET 4.0 不仅在语言和技术上弥补了原有 ASP.NET 2.0 的不足，还提供了很多新的控件和特色以提升开发人员的工作效率。与之相应的 Visual Studio 2010 除了保持与 Visual Studio 旧版本相同的特点之外，也提供了大量新的特色帮助。本书全面介绍了 ASP.NET 4.0 技术的开发与使用，站在实用和实际的角度，深入浅出地分析该技术的各个要点。如果读者曾使用前一版本的 ASP.NET 编过程序，可以将重点放在学习本书 ASP.NET 的新特性上，如第 10 章的 LINQ、第 15 章的 AJAX 和第 16 章的 WCF 服务等。

本书是作者根据多年从事网络程序设计工作和讲授计算机专业相关课程的教学实践，在已编写多部讲义和教材的基础上编写而成的。本书精选大量例题，且增加了汉字注释，所有例题都在 Visual Studio 2010 上调试通过，读者可按照书中提示信息找到每章的源码（如 ch5_4 表示第 5 章第 4 小节的所有代码）。

本书分为 17 章。第 1 章～第 4 章为基础篇，主要内容包括 Web 程序设计概述、HTML 和 CSS、JavaScript 编程基础和 C#语言基础。本书将 JavaScript 单独讲解，主要是因为从实践经验中发现 ASP.NET 中使用 JavaScript 是很有必要的，没有 JavaScript 编程基础，有些简单的问题都很难理解，详细说明请见第 3 章与第 7 章的 7.6 节。第 5 章～第 11 章为核心篇，主要内容包括 ASP.NET Web 开发基础、APS.NET 对象及状态管理、ASP.NET 4.0 服务器控件、ADO.NET 数据访问、数据绑定技术与绑定控件、使用 LINQ 和 ASP.NET 网页布局与标准化。通过本篇的学习，读者能够开发出小型的 Web 应用程序，掌握数据库以及网站的页面设计。第 12 章～第 16 章为提高篇，主要内容包括 ASP.NET 应用程序安全技术、文件操作、ASP.NET 中使用 XML、ASP.NET 的 AJAX 扩展、Web 服务和 WCF 服务。通过本篇的学习，读者可以掌握 ASP.NET 安全技术、Web 服务在网站高级开发中的应用以及 ASP.NET 新技术。第 17 章为系统集成篇，主要内容包括网站发布、打包与安装。通过本篇的学习，读者可以将自己开发的程序部署到用户的服务器上。

本书第 1 章～第 4 章由张恒编写，第 5 章～第 11 章由刘艳丽编写，第 12 章～第 14 章由周洁编写，第 15 章～第 16 章由蔡体健编写，第 17 章由石红芹编写，全书由刘艳丽统稿。本书的代码由华东交通大学研究生协助编写与整理。

虽然我们尽可能保证文字和代码没有错误，但由于编者水平有限，加之编写时间仓促，书中难免存在错误和疏漏之处，希望广大读者批评指正。

编　者

2012 年 9 月

目　录

第 1 章　Web 程序设计概述

Web 应用和相关技术的飞速发展给人们的工作、学习和生活带来了重大变化，人们可以利用网络处理数据、获取信息，极大地提高了工作效率。本书以 ASP.NET 4.0 为框架讲解 Web 程序设计，在深入介绍之前，有必要了解 Web 程序设计的基本内容、概念和方法等基础知识。

1.1　Internet 与 WWW 概述

1.1.1　Internet 概述

Internet，中文正式译名为因特网，又叫做国际互联网。它是由那些使用公用语言互相通信的计算机连接而成的全球网络。一旦计算机连接到它的任何一个节点上，就意味着这台计算机已经连入 Internet 了。Internet 目前的用户已经遍及全球，有超过几亿人在使用 Internet，并且它的用户数还在呈几何倍数上升。

Internet 是一组全球信息资源的总汇。有一种粗略的说法，认为 Internet 是由许多小的网络（子网）互连而成的一个逻辑网，每个子网中连接着若干台计算机（主机）。Internet 以相互交流信息资源为目的，基于一些共同的协议，并通过许多路由器和公共互联网组成，它是一个信息资源和资源共享的集合。计算机网络只是传播信息的载体，而 Internet 的优越性和实用性则在于本身。与 Internet 相关的常用术语简单解释如下。

（1）因特网（Internet）：专指全球最大的、开放的、由众多网络相互连接而成的计算机网络。它由美国的 ARPAnet 发展而来，主要采用 TCP/IP。

（2）万维网（World Wide Web，WWW）：也称环球信息网，是基于超文本的、方便用户在 Internet 上搜索和浏览信息的信息服务系统。

（3）超文本（Hypertext）：是一种全局性的信息结构，它将文档中的不同部分通过关键字建立链接，使信息得以用交互方式搜索。它是超级文本的简称。

（4）超媒体（Hypermedia）：是超文本和多媒体在信息浏览环境下的结合，是超级媒体的简称。

（5）主页（HomePage）：是通过万维网进行信息查询时的起始信息页。

（6）浏览器（Brower）：这里专指 Web 浏览器，如 Microsoft 的 IE（Internet Explorer），以及可以跨平台的 Netscape Navigator、Opera 等。它可以向万维网服务器发送各种请求，并对服务器发来的、由 HTML 定义的超文本信息和各种多媒体数据格式进行解释、显示和播放。

（7）目录服务（Directory Service）：是 Internet 上根据用户的某些信息反查找另一些信息的一种公共查询服务。

（8）防火墙（Firewall）：是用于将 Internet 的子网和 Internet 的其他部分相隔离，以达到网络安全和信息安全效果的软件或硬件设施。

（9）Internet 服务商（Internet Service Provider，ISP）：是向用户提供 Internet 服务的公司或机构。其中，大公司在许多城市都设有访问站点，小公司则只提供本地或地区性的 Internet

服务。一些 ISP 在提供 Internet 的 TCP/IP 连接的同时，也提供他们自己各具特色的信息资源。

1.1.2 WWW 概述

WWW 是 World Wide Web（环球信息网）的缩写，也可以简称为 Web，中文名字为“万维网”。它起源于 1989 年 3 月，是由欧洲量子物理实验室（the European Laboratory for Particle Physics，CERN）所发展出来的主从结构分布式超媒体系统。通过万维网，人们只要使用简单的方法，就可以很迅速方便地取得丰富的信息资料。由于用户在通过 Web 浏览器访问信息资源的过程中，无须再关心一些技术性的细节，而且界面非常友好，因而 Web 在 Internet 上一推出就受到了热烈的欢迎，走红全球，并迅速得到了爆炸性的发展。

长期以来，人们只是通过传统的媒体（如电视、报纸、杂志和广播等）获得信息。但随着计算机网络的发展，人们已不再满足于传统媒体那种单方面传输和获取信息的方式，而希望有一种主观的选择性。现在，网络上提供各种类别的数据库系统，如文献期刊、产业信息、气象信息、论文检索等。由于计算机网络的发展，信息的获取变得非常及时、迅速和便捷。

到了 1993 年，WWW 的技术有了突破性的进展，它解决了远程信息服务中的文字显示、数据连接以及图像传输的问题，这使得 WWW 成为 Internet 上最为流行的信息传播方式。现在，Web 服务器成为 Internet 上最大的计算机群，Web 文档之多，链接的网络之广，令人难以想象。可以说，Web 为 Internet 的普及迈出了开创性的一步，是近年来 Internet 上取得的最激动人心的成就。

WWW 采用的是浏览器/服务器结构，其作用是整理和存储各种 WWW 资源，并响应客户端软件的请求，把客户所需的资源传送到 Windows XP、Windows 7、UNIX 或 Linux 等平台上。

1.2 Web 浏览器与 Web 服务器

1.2.1 Web 浏览器

浏览器是用于显示网页伺服器或档案系统内的 HTML 文件，并让用户与这些文件互动的一种软件。个人计算机上常见的网页浏览器包括 Microsoft 的 Internet Explorer、Mozilla 的 Firefox、Opera 和 Safari。浏览器是最经常使用的客户端程序。全球资讯网是全球最大的连接文件网络文库。

网页浏览器主要通过 HTTP 连接网页伺服器而取得网页，HTTP 容许网页浏览器送交资料到网页伺服器并且获取网页。目前最常用的 HTTP 是 HTTP/1.1，这个协议在 RFC2616 中被完整定义。HTTP/1.1 有自己一套 Internet Explorer 并不完全支持的标准，然而许多其他网页浏览器则完全支持这些标准。

网页的位置（网址）以 URL（统一资源定位符）指示；以 http:开头的网址表示通过 HTTP 登录。很多浏览器同时支持多种类型的 URL 及协议，如 ftp:是 FTP（文件传送协议），gopher:是 Gopher，https:是 HTTPS（以 SSL 加密的 HTTP）。

网页通常使用 HTML（超文本标记语言）文件格式，并在 HTTP 内以 MIME 内容形式来定义。大部分浏览器均支持许多 HTML 以外的文件格式，如 JPEG、PNG 和 GIF 图像格式，还可以利用外挂程式来支持更多文件类型。在 HTTP 内容类型和 URL 协议的结合下，网页设计者可以把图像、动画、视频、声音和流媒体包含在网页中，或让人们通过网页取得它们。

早期的网页浏览器只支持简易版本的 HTML。专属软件浏览器的迅速发展促使非标准

HTML 代码的产生。这导致了浏览器相容性的问题。现代的浏览器（Mozilla、Opera 和 Safari）支持标准的 HTML 和 XHTML（从 HTML 4.01 版本开始）。它们显示出来的网页效果都一样。Internet Explorer 仍未完全支持 HTML 4.01 及 XHTML 1.x。现在许多网站都是使用所见即所得的 HTML 编辑软件来建构的，这些软件包括 Adobe Dreamweaver 和 Microsoft Frontpage 等。它们通常预设产生非标准 HTML，这阻碍了 W3C 制定统一标准，尤其是 XHTML 和 CSS（层叠样式表，设计网页时用）。

有一些浏览器还载入了一些附加组件来 Usenet 新闻组、IRC（互联网中继聊天）和电子邮件。支持的协议包括 NNTP（网络新闻传输协议）、SMTP（简单邮件传输协议）、IMAP（交互邮件访问协议）和 POP（邮局协议）。

1.2.2　Web 服务器

Web 服务器也称为 WWW(World Wide Web)服务器，主要功能是提供网上信息浏览服务。WWW 是 Internet 的多媒体信息查询工具，是 Internet 近年才发展起来的服务，也是发展最快和目前应用最广泛的服务。正是因为有了 WWW 工具，才使得近年来 Internet 迅速发展，且用户数量飞速增长。Web 服务器是指驻留于 Internet 上的某种类型计算机的程序。当 Web 浏览器（客户端）连接到服务器上并请求文件时，服务器将处理该请求并将文件发送到该浏览器上，附带的信息会告诉浏览器如何查看该文件（即文件类型）。服务器使用 HTTP（超文本传输协议）进行信息交流，这就是人们常把它称为 HTTP 服务器的原因。

Web 服务器是可以向发出请求的浏览器提供文档的程序。

（1）服务器是一种被动程序：只有当 Internet 上运行在其他计算机中的浏览器发出请求时，服务器才会响应。

（2）最常用的 Web 服务器是 Apache 和 Microsoft 的 Internet 信息服务器（Internet Information Server，IIS）。

（3）Internet 上的服务器也称为 Web 服务器，是一台在 Internet 上具有独立 IP 地址的计算机，可以向 Internet 上的客户机提供 WWW、E-mail 和 FTP 等各种 Internet 服务。

Web 服务器和 Web 浏览器的关系如图 1-1 所示。

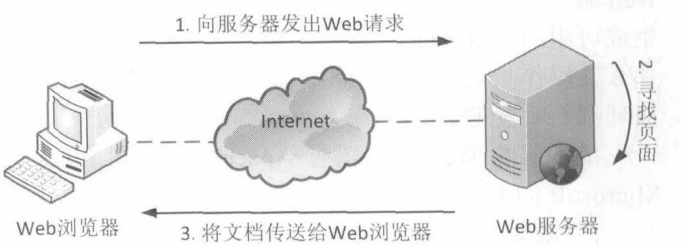


图 1-1　Web 服务器和 Web 浏览器的关系

Web 服务器不仅能够存储信息，还能在用户通过 Web 浏览器提供的信息的基础上运行脚本和程序。Web 服务器传送（serves）页面使浏览器可以浏览，然而应用程序服务器提供的是客户端应用程序可以调用（call）的方法（methods）。更确切的说：Web 服务器专门处理 HTTP 请求(request)，但是应用程序服务器是通过很多协议来为应用程序提供（serves）商业逻辑（business logic）的。

Web 服务器可以解析（handles）HTTP。当 Web 服务器接收到一个 HTTP 请求时，会返回一个 HTTP 响应（response），如送回一个 HTML 页面。为了处理一个请求，Web 服务器可以响应一个静态页面或图片，进行页面跳转(redirect)，或者把动态响应（dynamic response）的产生委托（delegate）给其他的程序，如 CGI 脚本、JSP（JavaServer Pages）脚本、servlets、ASP(Active Server Pages）脚本、服务器端（server-side）JavaScript，或者其他的服务器端（server-side）技术。无论它们（脚本）的目的如何，这些服务器端（server-side）的程序通常产生一个 HTML 的响应来让浏览器可以浏览。

Web 服务器的代理模型（delegation model）非常简单。当一个请求被送到 Web 服务器中时，它只单纯地把请求传递给可以很好地处理请求的程序（服务器端脚本）。Web 服务器仅仅提供一个可以执行服务器端程序和返回（程序所产生的）响应的环境，而不会超出职能范围。服务器端程序通常具有事务处理（transaction processing）、数据库连接（database connectivity）和消息发送（messaging）等功能。

1.3 Web 编程概述

Web 是一种典型的分布式应用框架。Web 应用中的每一次信息交换都要涉及客户端和服务端两个层面。因此，Web 编程技术大体上也可以分为客户端技术和服务端技术两大类。

1.3.1 Web 的工作原理

Web 的信息源保存在 Web 站点中，用户通过 Web 浏览器来访问。因此，Web 是一种基于客户机/服务器（Client/Server，C/S）的体系结构。用户使用浏览器从网上查阅 Web 信息，把需要的信息从网上下载到本机。信息分布点和用户需求信息的不同，表现在 Web 上是链接地址的不断变化。

浏览器的主要功能是解释并显示由 Web 服务器传送来的、由 HTML 写成的文档，包括嵌入 HTML 文档中的 GIF 和 JPEG 格式的图像。此外，浏览器还可以根据用户的需要配置某些辅助应用程序，用来处理嵌入 HTML 文档中的声音、视频等外部多媒体信息。通常将 Web 浏览器中显示的 HTML 文本称为 Web 页面（Page）。

Web 服务器是一个软件，用于管理 Web 页面，并使这些页面通过本地网络或 Internet 供客户机浏览器使用。在使用 Internet 的情况下，Web 服务器和浏览器通常位于两台不同的计算机上，也许它们之间相隔很远，甚至不在一个国家。但在本地情况下，也许是用一台计算机运行 Web 服务器软件，然后在同一台计算机上通过浏览器浏览它的 Web 页面。访问远程 Web 服务器与本地服务器之间没有什么差别，因为不论处于何种情况，Web 服务器的功能（即生成可用的 Web 页面）保持不变。如果用户是唯一在自己的计算机上访问 Web 服务器的人，那么其操作与用户在自己的计算机上运行 Web 服务器的操作相同。无论是何种情况，其工作原理都是不变的。

最常用的 Web 服务器有 Apache、IIS 和 iPlanet 的 Enterprise 服务器等，本书将只介绍 Microsoft 的 IIS。IIS 是能够运行 ASP.NET 的唯一服务器（将在 1.4 节重点介绍 IIS 的安装）。

1. 静态 Web 页面的工作原理

在 Internet 上浏览网页时，会发现许多 Web 页面的内容和外观总是保持不变，并且这些页面的文件后缀名都是.htm 或者.html，这就是静态 Web 页面。下面编写一个简单的名为“Welcome.htm”的静态页面并对其进行访问，步骤如下。

（1）新建一个文本文件，并输入如下代码。

```
<html>
<!-- 标题-->
<head>
    <title>我的页面标题</title>
</head>
<body>
<!-- 页面主体-->
Welcome! 这是最简单的网页。
```

```
</body>
</html>
```

（2）将此文件命名为“Welcome”，并修改扩展名“txt”为“htm”。

（3）将此文件保存到“C:\inetpub\wwwroot”目录下。将文件保存到该目录下是因为本书例程运行所用的Web服务器的主目录设为“C:\inetpub\wwwroot”。具体的内容将在本书后续章节介绍。

（4）启动IE，在地址栏中输入地址“http://localhost/Welcome.htm”，并按【Enter】键，运行结果如图1-2所示。其中“http://localhost/”代表本机的Web服务器。

图 1-2　静态页面

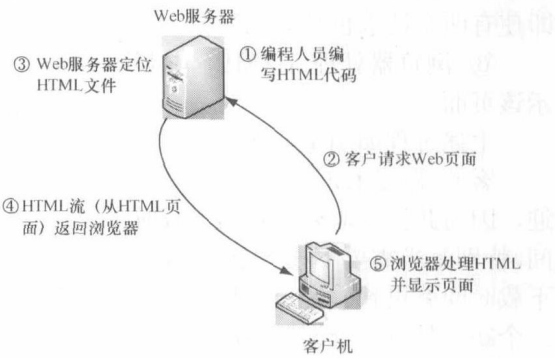


图 1-3　静态 Web 页面的工作原理

实际上，在用户访问这个页面之前，页面的内容已经确定，不管用户何时访问，以怎样的方式访问，页面的内容都不会再改变。静态页面如何显示在客户机浏览器中有以下 5 个步骤，过程如图 1-3 所示。

（1）编程人员编写由纯 HTML 代码组成的 Web 页面，并将其以.htm 文件格式保存到 Web 服务器上（在 Web 服务器上发布）。

（2）用户在其浏览器中输入页面请求（URL），该请求从浏览器传送到 Web 服务器。

（3）Web 服务器确定.htm 页面的位置，并将它转换成 HTML 流。

（4）Web 服务器将 HTML 流通过网络传回到浏览器。

（5）浏览器处理 HTML 并显示该页面。

类似 Welcome.htm 这样静态的、纯 HTML 文件能够呈现出完全可用的 Web 页面，甚至可以给这样的页面加入更多的 HTML 代码来修改字体和颜色，提高页面的显示效果和可用性。然而，编写纯 HTML 也只能做这些工作，因为页面的内容在用户请求页面之前已经完全确定，没有任何用户交互或动态响应功能。

2. 动态 Web 页面的工作原理

动态 Web 页面不能在用户请求页面之前通过将页面代码保存到文件这一方法来创建，而是在得到页面请求之后再生成 HTML 文件。主要有以下两种方法可以实现此功能。

（1）客户端动态 Web 页面。

在客户端模型中，附加到浏览器上的模块（即插件）完成创建动态页面的全部工作。通常包含一套指令的单独文件随 HTML 代码传送到浏览器，HTML 页面对该文件进行引用。但是，常见的另一种情况是这些指令与 HTML 代码混合在一起，当用户请求 Web 页面时，浏览器利用这些指令生成纯 HTML 页面。也就是说，页面根据用户请求动态生成。这样就生成了一个在浏览器中显示的 HTML 页面。

在客户端模型中，前面介绍生成 Web 页面的 5 个步骤变成了以下 6 个步骤。

① 编程人员编写一套用于创建 HTML 的指令，并将它保存到.htm 文件中。也可以用其他语言编写一套指令，这些指令可以包含在.htm 文件中，或放在单独的文件中。

② 用户在其浏览器中输入 Web 页面请求，该请求就从浏览器传送到 Web 服务器。

③ Web 服务器确定.htm 页面的位置，也许还需要确定包含指令的其他文件的位置。

④ Web 服务器将新创建的 HTML 流与指令通过网络传回浏览器。

⑤ 位于浏览器的模块会处理指令，并将.htm 页面的指令以 HTML 形式返回（只返回一个页面，即使有两个请求也是如此）。

⑥ 浏览器处理 HTML，并显示该页面。

上述过程如图 1-4 所示。

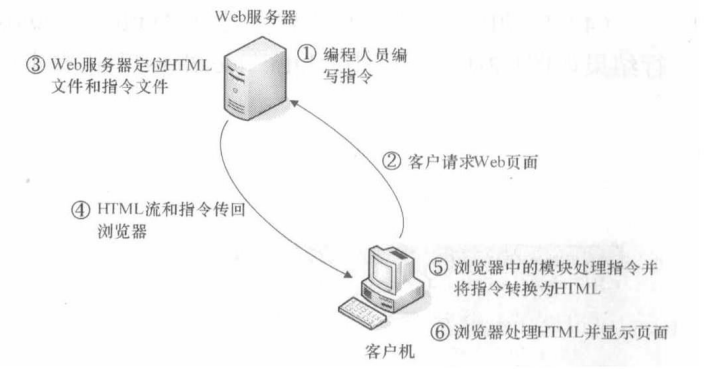


图 1-4 客户端动态 Web 页面的工作原理

客户端技术近来已不再受欢迎，因为此技术需要较长的下载时间，特别是当需要下载多个文件时，下载时间就更长。客户端技术的第二个缺点是每一个浏览器以不同的方式解释客户端脚本代码，因此无法保证所有的浏览器都能够理解它们。客户端技术的第三个缺点是当编写使用服务器端资源（如数据库）的客户端代码时会出现问题，因为代码是在客户端解释的，而客户端脚本代码是不安全的，很容易在浏览器中查看源代码。

（2）服务器端动态 Web 页面。

利用服务器端模型，HTML 源代码与混合在其中的一套指令被传回到 Web 服务器。当用户请求页面时，这套指令用于生成 HTML 页面，页面会根据请求动态生成。在服务器端模型中，前面介绍的 5 个步骤也变成了 6 个，但由于处理指令的位置不同，这 6 个步骤与客户端模型中的 6 个步骤略有不同。具体步骤如下。

① 编程人员编写一套创建 HTML 的指令，并将这些指令保存到文件中。

② 用户在其浏览器中输入 Web 页面请求，该请求就从浏览器传递到 Web 服务器。

③ Web 服务器确定指令文件的位置。

④ Web 服务器根据指令创建 HTML 流。

⑤ Web 服务器将新创建的 HTML 流通过网络传回浏览器。

⑥ 浏览器处理 HTML，并显示 Web 页面。

上述过程如图 1-5 所示。

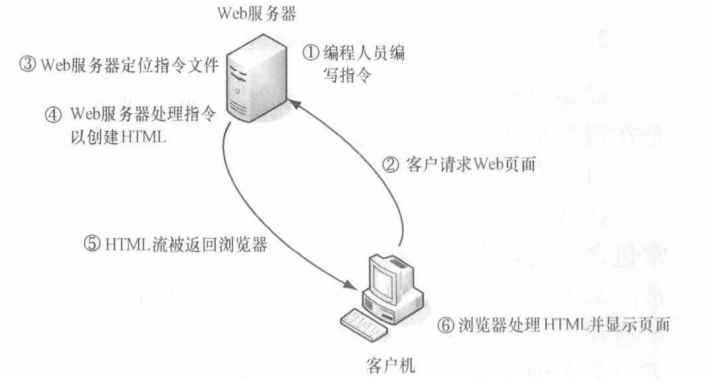


图 1-5 服务器端动态 Web 页面的工作原理

该方法与前面介绍方法的不同之处是在页面返回到浏览器之前，所有的处理过程都在服务器上完成。与客户端模型相比，此方法的主要优点之一是只有 HTML 代码传回浏览器，这意味着页面的初始代码隐藏在服务器中，而且可以保证大多数浏览器能够显示生成的 HTML 页面。ASP.NET 就属于服务器端模型。

1.3.2　动态 Web 开发技术概述

1. 提供动态内容的客户端技术

每一项提供动态内容的客户端技术都依赖于内置在浏览器中的模块（即插件）来处理指令。客户端技术是脚本语言、控件以及功能完善的编程语言的综合。

（1）JavaScript

JavaScript 是最早的浏览器脚本语言。JavaScript 语言的前身称为 Livescript，自从 Sun 公司推出著名的 Java 语言之后，Netscape 公司引进了 Sun 公司有关 Java 的程序概念，将原有的 Livescript 重新设计，并改名为 JavaScript。JavaScript 是一种基于对象和事件驱动并具有安全性能的脚本语言，JavaScript 可使网页变得更加生动。使用它的目的是与 HTML、Java 脚本语言一起实现在一个网页中链接多个对象，与网络客户交互作用，从而可以开发客户端的应用程序。它是通过嵌入或在标准的 HTML 中调入实现的。

JavaScript 编写容易，不需要有丰富的编程经验。现在，JavaScript 已经成为制作动态网页必不可少的元素，经常在网页上看到的动态按钮和滚动字幕等，大多数都是使用 JavaScript 技术制作的。

Microsoft 公司在 Internet Explorer 3.0 中推出了自己的 JavaScript 版本，其名称为 Jscript，且到目前为止，Internet Explorer 一直在支持它。虽然老版本的 Internet Explorer 和 Netscape 浏览器支持的语言有较大的差别，但现在这两个版本之间的差别很小。

（2）VBScript

在 Internet Explorer 3.0 中，Microsoft 公司也推出了自己的脚本语言——VBScript。VBScript 是基于 Visual Basic 的编程语言，直接与 JavaScript 竞争。就功能而言，VBScript 与 JavaScript 之间并没有太大的区别，具体使用哪一种语言由个人的爱好决定。但 VBScript 具有类似 VB 的简化功能，Visual Basic 开发人员有时会喜欢使用 VBScript，因为 VBScript 的大部分内容是 Visual Basic 语言（VB.NET 以前的版本）的子集。不过，对于初级编程者来说，VBScript 更具有吸引力的一点是不区分大小写（与 JavaScript 不一样），而且也不过分注重代码方面的细节。但正是由于这些“优点”，VBScript 运行速度慢，效率也较低。

VBScript 最大的缺点是没有一家 Microsoft 公司以外的浏览器支持由 VBScript 编写的客户端脚本。虽然曾经有一些用于 Netscape 的插件可提供对 VBScript 的支持，但一直没有流行起来，相比之下，JavaScript 有更为广泛的应用与支持。如果希望在客户端编写 Internet 上的 Web 页面，JavaScript 则是唯一可选择的语言。当编写内部网页面，且已知所有客户端都是 Windows 系统上的 IE 时，才考虑使用 VBScript。

JavaScript 和 VBScript 都依赖于称为脚本引擎的模块，该模块被内置到浏览器中，以动态方式处理指令，在这种情况下，这些指令也称为脚本。

（3）Java 小应用程序

Java 是用于开发应用程序的跨平台语言。当 Java 在 20 世纪 90 年代中期首次应用于 Web 时，曾引起了巨大的轰动。Java 代码以小应用程序的形式使用，这些小应用程序基本上是可以借助于<applet>标记方便地插入 Web 页面的 Java 组件。

Java 比脚本语言的功能更为强大且不牺牲安全性，它在诸如图形功能、文件处理方面提供了更好的支持，Java 也通过 JDBC 提供了强有力的数据库支持。

各种浏览器均通过 Java 虚拟机（JVM）得到内置的 Java 支持，而且有上千个标准的<object>标记和非标准的<applet>标记用于给 Web 页面添加 Java 小应用程序。这些标记告诉