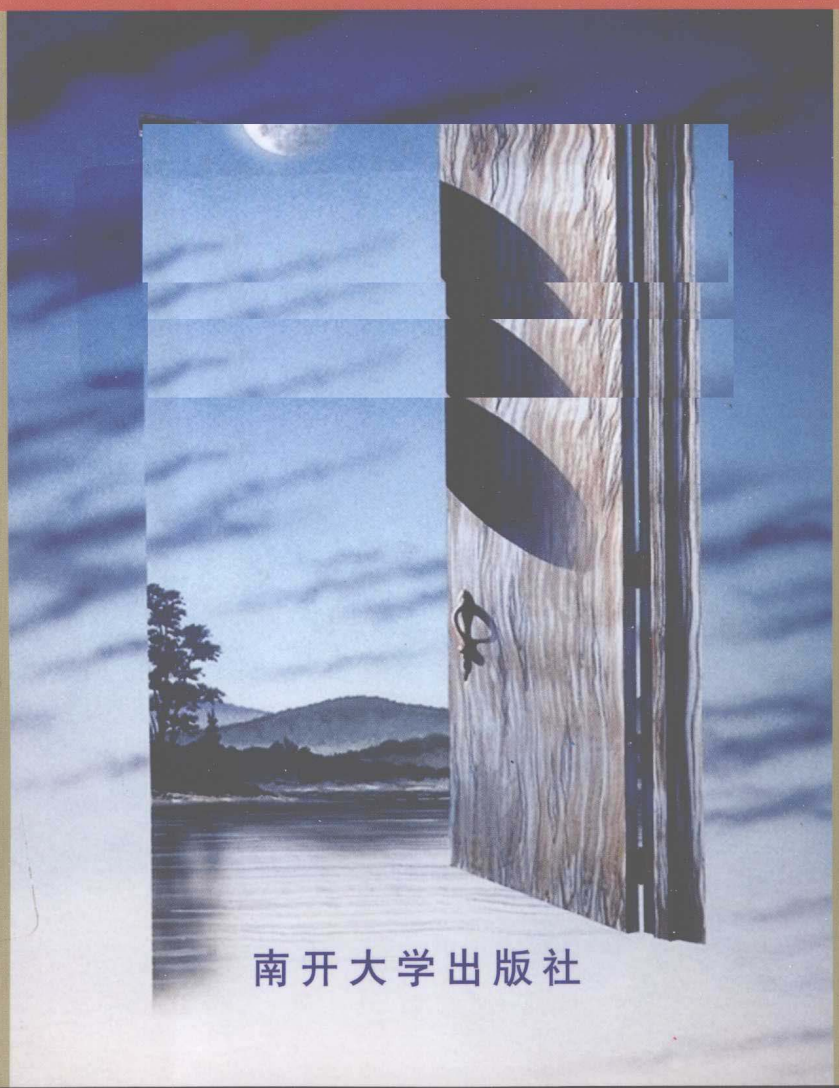
■ 高等院校基础课教材

ASP.NET实用技术

主　编　郭素芳　崔凤梅　孟冬梅

主　审　朱耀庭



南开大学出版社

ASP.NET 实用技术

主编　郭素芳　崔凤梅　孟冬梅

主审　朱耀庭

南开大学出版社

天　津

内容简介

本书秉持严谨、实用的原则，较为详细地介绍了使用 ASP.NET 及其 C#语言进行 Web 应用程序开发的方法。全书共分为 9 章，内容涉及 ASP.NET 概述、C#面向对象编程基础、ASP.NET 常用控件、ASP.NET 常用内置对象、数据库访问技术、在 ASP.NET 中应用 XML、主题与母版页、ASP.NET 的配置和优化，并在第九章详细剖析了使用 ASP.NET 3.5 进行 Web 开发的一个综合案例：个体工商户日常管理网站。本书的每个章节均配有大量丰富的实例，同时，主要章节均配有实验及习题，便于初学者对于所述内容的理解。

本书条理清楚，案例翔实，深入浅出，既可以作为高等学校 ASP.NET 技术的教材，又可以作为使用 ASP.NET 技术从事 Web 程序开发的程序员的技术资料。

图书在版编目(CIP)数据

ASP. NET 实用技术 / 郭素芳，崔凤梅，孟冬梅主编.
—天津：南开大学出版社，2010.5
ISBN 978-7-310-03405-5

Ⅰ.①A… Ⅱ.①郭…②崔…③孟… Ⅲ.①主页制作—程序设计 Ⅳ.①TP393.092

中国版本图书馆 CIP 数据核字(2010)第 060935 号

南开大学出版社出版发行
出版人：肖占鹏
地址：天津市南开区卫津路 94 号　　邮政编码：300071
营销部电话：(022)23508339　23500755
营销部传真：(022)23508542　　邮购部电话：(022)23502200
*
天津泰宇印务有限公司印刷
全国各地新华书店经销
*
2010 年 5 月第 1 版　　2010 年 5 月第 1 次印刷
787×1092 毫米　16 开本　20.125 印张　511 千字
定价：39.00 元(含光盘)
如遇图书印装质量问题，请与本社营销部联系调换，电话：(022)23507125

序

ASP.NET 是微软公司推出的网页设计的拳头产品。ASP.NET 技术的出现，革新了传统的 Web 设计模式，给 Web 开发带来新的技术和手段。

本书的写作基于 Visual Studio 2008+SQL Server 2005 环境。使用的开发语言主要是 C#，这也是目前流行的一种开发语言。

本书作者均为具有网页设计技术和多年 C#语言教学经验的教师，该书在讲解知识的同时又注重教学设计。本书的写作始终贯彻以应用为主的宗旨，在内容讲解中既保证了理论架构，又侧重于应用，而且重点介绍应用。书中的每一个知识点都有实例讲解，每一章都配有实验和习题，最后有案例；所有例题、实例和案例全部通过实际调试，并附在光盘中。本书非常适合以培养应用型人才为目标的学院作为教材使用。

南开大学信息学院计算机科学系教授/博士生导师
南开大学滨海学院计算机科学系主任
朱耀庭

目 录

第一章　ASP.NET 概述

【学习目标】

了解微软.NET 框架体系结构。
理解公共语言运行时（CLR）和.NET Framework 类库（FCL）。
了解.NET 框架涉及的几个概念。
理解 ASP.NET 的特点及工作原理。
掌握 IIS 的安装与配置。
掌握 Visual Studio 2008 的安装。
熟悉 Visual Studio 2008 集成开发环境。
掌握构建 ASP.NET 网站的基本步骤。
学会构建一个简单的 ASP.NET 网站。

ASP.NET 是微软公司推出的全新的 Web 动态页面编程环境。作为微软公司推出的全新的.NET 架构体系的一部分，ASP.NET 提供了一种基于组件的、可扩展且易于使用的方法，大大简化了 Web 编程。ASP.NET 不是对 ASP 的一个简单意义上的升级，而是一个基于 Web 开发的全新框架。ASP.NET 不再使用传统的脚本语言，而使用.NET 框架所支持的 VB.NET、C#.NET 等语言作为其开发语言，避免了原有脚本语言的诸多限制，因而可以为程序员在短时间内开发出高效、高可靠性且具有高可扩展性的 Web 程序提供强有力的支持和保障。

1.1　.NET 框架体系结构

微软.NET 平台，也即微软 XML Web services 平台。通过这个平台，可以非常便捷地创建各种 XML Web 服务，并将这些服务集成在一起。通过 XML Web 服务，允许应用程序通过 Internet 进行通讯和共享数据，而不管所采用的是哪种操作系统、设备或编程语言。.NET 框架（.NET Framework）正是整个.NET 平台中用于编写 Web 服务的开发工具的核心和基础。

1.1.1　.NET 框架体系结构概述

.NET 开发平台包括.NET 框架（.NET Framework）和 Visual Studio 工具集。其中，.NET 框架是开发平台的基础与核心。.NET 框架应用了许多全新的技术，提供了一个一致面向对象的高效安全的编程环境，极大地简化了 Internet 环境下 Web 应用程序的开发。

1．.NET 框架的组成

.NET 框架主要由两部分组成：公共语言运行时（Common Language Runtime，CLR）和.NET Framework 类库（Framework Class Library, FCL）（如图 1-1 所示）。

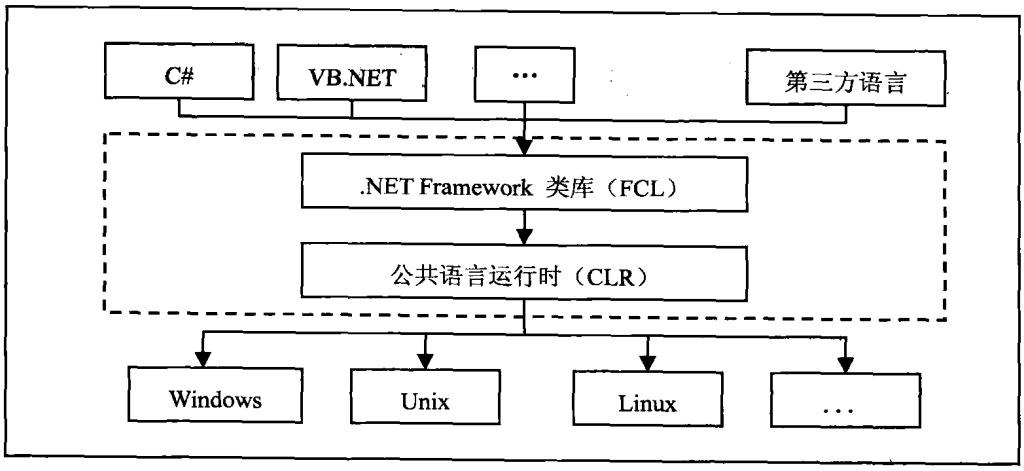


图 1-1 .NET 框架的组成（虚线部分）

（1）公共语言运行时（CLR）

CLR 可以理解为一个通用语言运行环境，它是.NET 框架的基础，负责管理和执行由.NET 编译器编译产生的中间语言代码。CLR 提供了程序运行时的内存管理、线程管理、远程管理、垃圾自动回收、代码执行、类型安全验证、安全检查、异常处理等服务。CLR 的工作原理为.NET 框架所提供的多语言执行环境提供了有力保障。

.NET 框架设置了中间语言，为跨平台服务提供了基础。.NET 框架产生的最终执行代码与具体编程语言无关，只和中间语言有关，这就使得各种服务程序的开发不再受任何编程语言的限制。目前，可以用来编写.NET 应用程序的编程语言很多，如微软的 C++、 VB.NET、微软专门随 .NET 推出的开发语言——C#，以及许多第三方语言，如 COBOL、Eiffel、Perl、Delphi 等。

用各种编程语言编写的程序经过编译后，并不会直接产生 CPU 可执行的代码，而是首先转变为一种不依赖于 CPU 的中间语言（IL，或称微软中间语言 MSIL），CLR 最终执行的是中间语言。当 CLR 执行这些中间语言指令时，CLR 内部的实时编译器（JIT）将它们转换成 CPU 可执行的本地代码，最终实现程序的运行，从而保证了.NET 框架的语言无关性以及平台无关性。因为 CLR 支持多种并允许嵌入第三方实时编译器，因此，同一段 MSIL 代码可以被不同的 JIT 编译成适合不同环境的本地代码（如图 1-2 所示）。

（2）.NET Framework 类库（FCL）

.NET Framework 类库是一个关于类和类型的程序库，它是.NET 框架中的另外一个重要组成部分（如图 1-3 所示）。它包含了数千个可重用的类，提供了对系统功能的访问，是构建 .NET Framework 应用程序、组件和控件的基础。各种不同的基于 CLR 的编程语言可以使用.NET Framework 类库开发出各种应用程序。

.NET 中的类库封装了对 Windows、网络、文件、多媒体的处理功能，是所有.NET 语言都必须使用的核心类库。下面是 Framework 类库中的几个类：

File 类：该类可以用于检查某个磁盘文件是否存在，创建新文件，删除指定文件，以及进行与文件有关的其他操作。

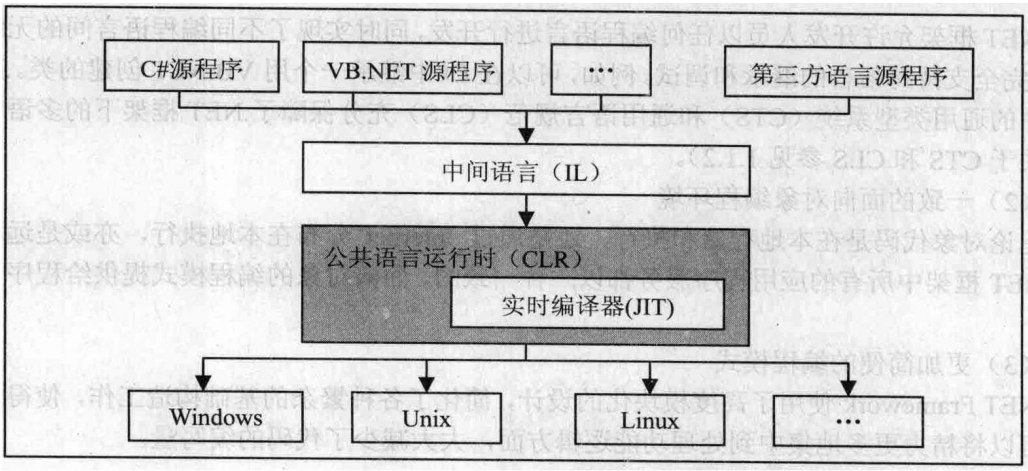


图 1-2　.NET 程序执行过程

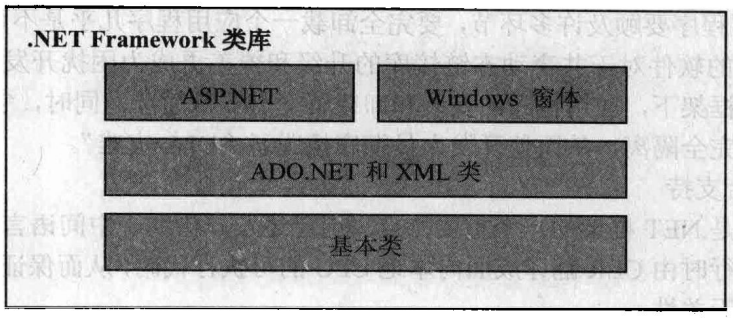


图 1-3　.NET Framework 类库

Graphics 类：使用该类可以处理各种不同格式的图像文件，如 GIF，JPG，BMP，PNG 等类型文件。也可以通过该类创建如矩形、圆、椭圆、圆弧等各种图形。

SmtpClient 类：使用该类可以发送包含附件的电子邮件等。

.NET Framework 类库中有上万个这样的类，为程序员开发各类应用程序提供了极大的方便。

.NET Framework 类库被组织成一个命名空间层次结构，该结构使用点语法命名方案，将相关类型分为不同的命名空间组，以便可以更容易地搜索和引用它们。System 是这个层次结构的根。例如，System 命名空间中就包含了 Object 基类型，所有其他类型都直接或间接由此基类型继承而得到。此外，System 命名空间中还包括了所有应用程序使用的基本数据类型的类，如 Byte、Char、Array、Int32、String 等以及其他许多实用类型。为了通晓.NET 开发平台的各种特性，程序员应当了解自己需要的类型包含在哪个命名空间中（命名空间的概念参见本书 2.6 节）。

2. .NET 框架的优点

.NET 框架具有强大的功能，与之前的任何微软开发平台所提供的技术相比，该框架有了质的飞跃。应当说，微软将现有技术中所有最好和最常用的功能都集中在这一个架构中。概括起来说，其优点主要体现在如下方面：

（1）跨越所有的编程语言

.NET 框架允许开发人员以任何编程语言进行开发，同时实现了不同编程语言间的无缝集成。它完全支持跨语言的继承和调试。例如，可以在 C#中继承一个用 VB.NET 创建的类。.NET 框架中的通用类型系统（CTS）和通用语言规范（CLS）充分保障了.NET 框架下的多语言编程（关于 CTS 和 CLS 参见 1.1.2）。

（2）一致的面向对象编程环境

无论对象代码是在本地存储和执行，还是通过 Internet 分布在本地执行，亦或是远程执行，.NET 框架中所有的应用程序服务都以一种一致的、面向对象的编程模式提供给程序开发人员。

（3）更加简便的编程模式

.NET Framework 使用了高度模块化的设计，简化了各种繁杂的基础构造工作，使得开发人员可以将精力更多地集中到处理功能逻辑方面，大大减少了代码的编写量。

（4）轻松的软件部署和版本冲突控制

构建在.NET 框架上的软件比常规的软件更容易部署和管理。通常情况下，安装一个传统的 Windows 应用程序要顾及许多环节，要完全卸载一个应用程序几乎是不可能的。而不同开发商的不同版本的软件对于共享动态链接库的升级和覆盖更成为困扰开发人员的“DLL 灾难”。而在.NET 框架下，应用程序的安装和卸载就变得非常简单，同时，它的新型版本机制将应用程序组件完全隔离，从而使开发人员彻底摆脱了“DLL 灾难”。

（5）多平台支持

多平台支持是.NET 框架的一个重要特点。前已述及，借助于中间语言，.NET 框架确保源程序代码在运行时由 CLR 翻译成面向本地 CPU 的可执行代码，从而保证了.NET 框架下的程序开发的平台无关性。

（6）安全的代码执行环境

.NET 框架提供了一个强健的安全系统，包括基于角色的安全性和代码访问安全性等，该系统可以确保程序代码在严格约束的、管理员定义的安全环境中运行。

（7）高性能的代码执行环境

.NET 框架引入了许多新技术，如新的内存、线程及进程管理技术，确保内存泄漏不再发生等。这些技术提高了应用程序运行的可靠性。

1.1.2 .NET 框架中的几个基本概念

如前所述，.NET 框架主要由 CLR 和 FCL 两部分组成。除了深刻理解 CLR 和 FCL 这两个重要概念之外，若要真正理解.NET 框架，还需要弄清如下几个概念。

1．IL/MSIL（Intermediate Language/Microsoft Intermediate Language，中间语言）

为了实现跨语言开发和跨平台的目标，应用程序源代码经过编译器生成的结果必须不依赖于操作系统和计算机硬件的机器指令，而应当是一种中间的、在所有操作系统和计算机硬件平台上都能执行的代码。为实现这一目标，.NET 编写的所有应用程序都不是被编译为本地代码，而是编译成微软中间语言代码 MSIL（Microsoft Intermediate Language），简称 IL。

IL 是与硬件无关的语言，其本质是一组伪机器指令的集合。IL 代码最终由 JIT（Just In Time）编译器转换成可直接由本地 CPU 执行的机器代码。例如，C#和 VB.NET 源程序代码通过它们各自的编译器编译成 MSIL，MSIL 再通过 JIT 编译器编译成相应的本地操作系统专用代码。

2．CTS（Common Type System，通用类型系统）

在传统的基于 Windows 的程序开发过程中，由于各种语言的类型的不一致导致的问题经常出现，不同语言之间互用是很难实现的（例如，分别用 VB 和 C++开发的程序很难集成在一起），因为只有基于同一类型系统的语言才有可能实现互用。

为此，微软专门在.NET 框架中制定了通用类型系统（CTS），用于解决不同语言的数据类型不同的问题。CTS 用来统一描述.NET 框架中的各种类型的定义和行为。CTS 是.NET 框架的一个重要特性，它定义了标准的面向对象的类型，所有的.NET 编程语言均支持这些类型。CTS 使得一种语言编写的代码和另一种语言编写的代码进行无缝集成成为可能，从而确保.NET 框架能够提供统一的编程模式，并能支持多种编程语言。

例如，C#中的整型是 int，而 VB.NET 中的整型是 Integer，通过各自的编译器把它们两个编译成 CTS 中的通用的类型 Int32，而所有的.NET 语言共享这一类型系统，在它们之间实现无缝互操作。

在实际学习中，不需要直接学习 CTS 规则，只需要从一个具体的编程语言（本书使用 C#语言）入手，掌握该语言所提供的语法和类型规则。虽然不同语言定义类型的语法可能不同，但是编译成中间语言代码后，类型将统一为 CTS 通用类型。

3．CLS（Common Language Specification，通用语言规范）

CLS 是 CLR 定义的一组语言特性集合，主要用来解决不同编程语言在.NET 框架中互操作的问题。

只有在互操作过程中涉及的数据类型和语言特性对所有的语言来说是公共的，才能确保被不同语言编译器所编译的对象能够进行互操作。CTS 为不同语言间的互用奠定了基础，但要想真正实现.NET 中的语言集成却并非易事。因为编程语言的区别不仅仅在于类型，语法或者说语言规范存在的区别同样会因影响不同编程语言在.NET 框架下的集成。

为此，公用语言运行时 CLR 标识了一组公共语言特征的集合，称为公用语言规范（CLS）。CLS 制定了一种以.NET 平台为目标的语言所必须支持的最小特征，以及该语言与其他.NET 语言之间实现互操作所需要的完备特征。凡是遵守这个标准的语言在.NET 框架下都可以实现互相调用。如果开发的组件在应用程序接口中仅使用 CLS 的特征语言，那么该组件能够被任何支持 CLS 的语言所编译的组件所访问。CLS 实际上是 CTS 的一个子集，它定义了所有面向.NET 的程序需要符合的最小规范集。

4．托管代码与非托管代码

所谓托管代码，是指由公共语言运行时 CLR 执行的代码，而不是由本机操作系统直接执行的代码。简言之，托管代码就是中间语言代码，也即 IL 代码。换言之，不由 CLR 接管，而是由操作系统直接执行的代码称为非托管代码。

托管代码由.NET Framework 的 CLR 接管，而不是由 Windows 系统直接管理，因而享有 CLR 提供的类型安全检查、内存管理和释放无效对象等服务，确保了托管代码可以避免很多程序的错误，同时也增强了程序代码的安全性。只需要为每种操作系统和特定的硬件平台提供一个.NET Framework 平台，就可以让同样一个托管代码不加修改地在使用不同的操作系统和硬件结构的计算机上运行。

非托管代码实际上就是本地代码，它不由 CLR 执行，非托管代码必须自己提供垃圾回收、类型检查、安全支持等服务。非托管程序的运行必须依赖于操作系统（如 Windows、Linux 等），而且编译器生成的程序文件包含的是特定 CPU 的机器指令。由于不同 CPU 的机器指令

不同，所以，生成的程序不能不加修改地在具有不同种类 CPU 的计算机上运行。

5．程序集

在微软的 MSDN 文档中对程序集做了如下描述："程序集是 .NET Framework 应用程序的构造块；程序集构成了部署、版本控制、重复使用、激活范围控制和安全权限的基本单元。程序集是为协同工作而生成的类型和资源的集合，这些类型和资源构成了一个逻辑功能单元。程序集向公共语言运行库提供了解类型实现所需要的信息。对于运行库，类型不存在于程序集上下文之外。"

对于初学者来说，这段文字比较抽象。通俗地讲，一个程序集可以理解为一个项目，或者一个完整的包。程序集是可以进行部署的最小单位，同时每个程序集都可以采用自己的版本控制策略和权限配置方案等。程序集中包含所定义的类型和相关的资源。

在编译各种语言源代码时，生成的 IL 代码存储在一个程序集中。程序集包括可执行的应用程序和应用程序使用的库，此外程序集还包含所需数据的信息（称为元数据）和所需的其他资源（如声音、图片文件等）。程序集包含程序的元数据，表示调用给定程序集中代码的应用程序或其他程序集不需要指定注册表或其他数据源，因此，只需把文件复制到远程计算机的目录下即可以完成部署应用程序，这与以前的 COM 有了很大的不同。

程序集可以存储在一个文件中，这样的程序集称为单文件程序集，也可以存储在多个文件中。这些文件中有且只有一个主模块文件，另外还需要模块文件和资源文件，这样的程序称为多文件程序集。

1.1.3 ASP.NET 概述

ASP.NET 是微软公司推出的新一代基于 B/S 的动态 Web 开发工具，完全不同于传统的 ASP 技术。它是 .NET Framework 的一个子集，这个子集提供了创建动态 Web 站点所需的一组类的集合，这些类包含了各种在 Web 服务器上执行操作的控件。ASP.NET 作为一种建立在 CLR 基础之上的程序开发构架，主要用于在服务器上开发功能强大的 Web 应用程序。

1．Web 基本工作原理

简单地讲，Web 基本工作原理采用 B/S（浏览器/服务器）模式，即用户通过浏览器获取服务器的某项服务。

（1）静态 Web 页

静态 Web 页不含代码，是直接利用 HTML 标记语言描述的 Web 页面。这种页面没有数据库的支持，如果想在网页中访问数据库，例如建立一个购物网站或者论坛，单纯靠这种 HTML 页面是无法实现的。静态 Web 页面文件的后缀名通常为.htm 或者.html（如图 1-4 所示）。

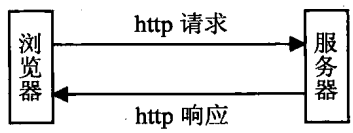


图 1-4 静态 Web 页工作原理

（2）动态 Web 页

动态 Web 页主要分为两大类：客户端的动态 Web 页和服务器端的动态 Web 页。

客户端的动态 Web 页，也叫 DHTML，主要是利用客户端脚本语言（如 JavaScript）实

现动态效果。它通常以脚本语言形式直接嵌在网页中，不需要与服务器交互。当这种页面文件从 WEB 服务器下载到客户端后，无须再经过服务器的处理，网页在客户端浏览器中直接响应用户的动作。从严格意义上讲，这种客户端的动态网页不是真正意义上的动态网页，也有另一种说法将这种页面界定为静态 Web 页。

服务器端的动态 Web 页，利用服务器端的语言实现动态的页面，是真正意义上的动态网页。这种页面程序在服务器端运行，通常都需要与数据库技术结合。服务器根据用户的请求，把从数据库中获得的数据组合到页面中，返回给客户端浏览器（如图 1-5 所示）。

ASP、PHP、JSP 以及本书述及的 ASP.NET 等均属于服务器端的动态网页开发技术。

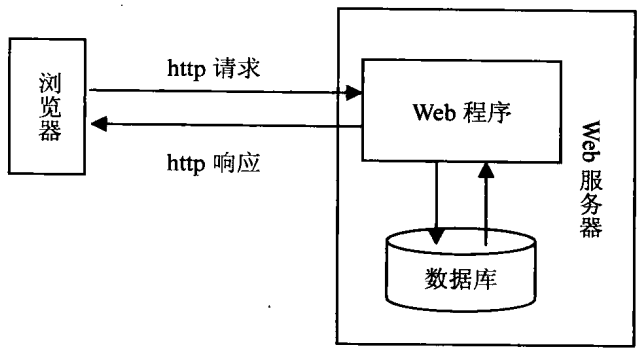


图 1-5　服务器端动态 Web 页工作原理

2．ASP 与 ASP.NET

ASP.NET 不能仅仅看做是 ASP 的升级，它在很多方面有质的突破。与 ASP 对比可以发现，ASP.NET 具有更大的优越性，可以大大简化编程工作。ASP.NET 与 ASP 的主要区别如下：

（1）开发语言不同

ASP 是一种服务器端脚本编写环境，在网页中仅限于使用脚本语言如 VBScript、JScript 等，缺乏事件驱动，增加了编程难度；ASP.NET 则允许使用 C#、VB.NET、C++.NET 等所有可运行于 CLR 之上的编程语言，这些语言功能完善，确保开发出强大的 Web 程序。

（2）代码的独立性不同

ASP 采用面向结构的编程方式，脚本语言编写的代码与 HTML 标记混合在一起，代码杂乱冗长，可读性差，使程序难于维护。ASP.NET 中，采用面向对象的编程方式，页面中包含各种可编程控件，以事件驱动方式编写代码。通过采用 code-behind 的方式，代码和界面设计实际上被置于两个不同的文件中，因而将界面设计工作和代码编写有效地分离，实现了代码的独立性，增强了程序的可维护性。

（3）运行机制不同

ASP 代码以解释方式运行，每次运行时都需要加载并逐行解释，程序执行效率比较低。ASP.NET 代码以编译方式运行。当服务器上的 ASP.NET 页面第一次被请求时，编译器先将 ASP.NET 代码编译为 IL 代码，并备一份到缓存中，再由即时编译器将其编译为本地 CPU 机器码。当页面第二次被同样请求时，直接由 JIT 编译这个备份，因此大大提高了运行效率。

（4）组件的部署