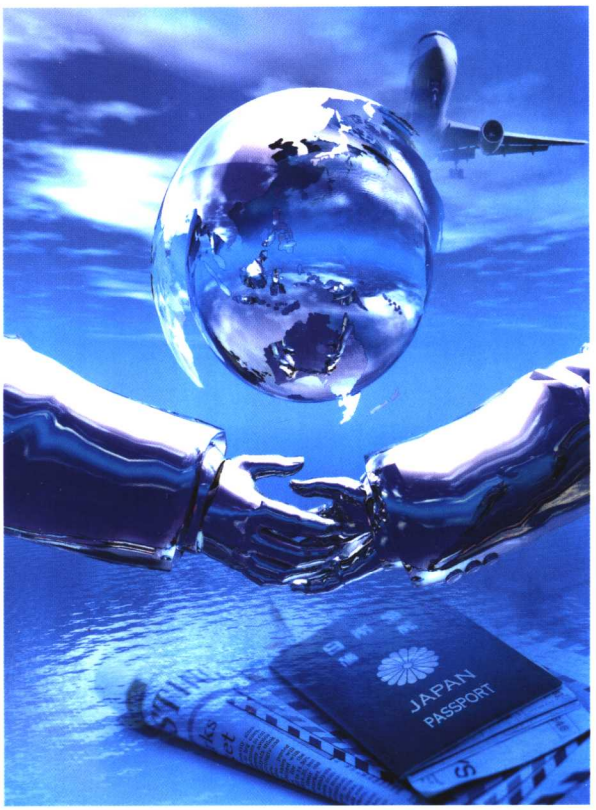
高等院校计算机应用技术系列教材

ASP.NET 简明教程

（C# 篇）

- 配置 ASP.NET 的运行环境
- C# 语法基础
- HTML 控件与 Web 服务器控件
- ASP.NET 常用内置对象
- ASP.NET 高级编程技术
- Web 增强控件与自定义控件
- ADO.NET 及其数据库访问技术
- 在 ASP.NET 中应用 XML
- ASP.NET 的配置与优化
- 在 ASP.NET 中实现安全和 Web 服务
- Web 网站构建课程设计指导



金雪云　编著

清华大学出版社

高等院校计算机应用技术系列教材

ASP.NET 简明教程(C#篇)

金雪云 编著

清华大学出版社

北 京

内 容 简 介

本书主要介绍了在 Windows 2000 操作系统上使用 ASP.NET 创建动态 Web 网页的相关技术，内容涵盖了 ASP.NET 程序的运行环境需求、C#语法基础、ASP.NET 扩展的 HTML 控件和 Web 内部控件的使用、ASP.NET 的常用内置对象、.NET 的命名空间、在 ASP.NET 中实现事件驱动、在 ASP.NET 程序中使用 Web 增强控件和自定义控件、ADO.NET 和使用 ADO.NET 进行数据库访问、在 ASP.NET 中应用 XML、对 ASP. NET 进行配置和优化，以及在 ASP.NET 中实现安全和 Web 服务的方法。本书附录部分还为学习网络程序设计课程的读者提供了一个课程设计题目——设计和实现一个完整网站与后台管理系统。

本书适合作为高等院校计算机与信息技术及相关专业的教材，也可供在.NET 框架下开发 Web 应用程序的程序员参考。本书所有代码可从 http://www.tupwk.com.cn/downpage 免费下载。

图书在版编目(CIP)数据

ASP.NET 简明教程(C#篇)/金雪云　编著. —北京：清华大学出版社，2006.1
(高等院校计算机应用技术系列教材)
ISBN 7-302-12191-5

Ⅰ. A…　Ⅱ. 金…　Ⅲ. ① 主页制作—程序设计—高等学校—教材　② C 语言—程序设计—高等学校—教材
Ⅳ. TP393.092

中国版本图书馆 CIP 数据核字(2005)第 143156 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
社 总 机：010-62770175　　**客户服务**：010-62776969
组稿编辑：王　定
文稿编辑：鲍　芳
封面设计：王　永
版式设计：康　博
印 刷 者：北京市清华园胶印厂
装 订 者：三河市李旗庄少明装订厂
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：21.75　**字数**：496 千字
版　　次：2006 年 1 月第 1 版　2006 年 1 月第1 次印刷
书　　号：ISBN 7-302-12191-5/TP·7856
印　　数：1～6000
定　　价：29.80 元

前　言

ASP.NET 作为.NET Framework 的一部分，具有.NET Framework 所拥有的一切优势。首先，在.NET Framework 中使用高级语言(例如 Visual Basic.NET，C#)编写程序时，并不是把这些语言所写的代码直接编译成机器代码，而是把程序编译成为中间语言(例如 MS 中间语言)。注意，中间语言并不是一种可以直接执行的机器代码，与高级语言编写的代码相比，它的可读性很差，但是进行了一系列的优化。为了执行中间语言，就需要一个执行环境 CLR。CLR 在.NET Framework 中的位置十分重要，可以说是.NET Framework 的基础。CLR 用 JIT(Just-In-Time)编译器把中间语言代码编译成可以执行的代码，并对程序进行最后的、与机器相匹配的优化，使得程序可以在宿主计算机上尽可能高效率地运行。这些经过优化的二进制代码会保存在缓存中，直到源代码改变为止。

从名字上看，ASP.NET 好像是 ASP 的一个新版本，但是从本质上讲，ASP.NET 革命性地改变了 Web 程序设计的设计方式。首先，ASP 使得网站的各种代码难于管理，因此面对大量的 HTML 代码和 VBScript，JavaScript 代码混合在一起的程序，程序员往往是一筹莫展，当需要改动程序时，宁愿写新的代码，也不愿意去改原来的程序，因为原来程序的模块化和可重用性都太低；而 ASP.NET 则可以使用 C#这样的面向对象的语言编写程序，并可以做到代码分离。第二，由于 ASP 只能使用 VBScript，JavaScript 这样的脚本语言，使很多功能都不可能轻松地实现；而 ASP.NET 则提供了很多功能强大的 Web 控件，使用事件驱动的方式进行程序设计。第三，ASP 程序是解释执行的，而 ASP.NET 程序是编译执行的，这导致 ASP 程序在执行效率上要大大低于实现同样功能的 ASP.NET 程序。为了改善网络应用程序的效率，随着.NET 的发布，学习如何在.NET 框架中使用 ASP.NET 成为很多 Web 程序设计人员的首选。

1. 本书内容介绍

第 1 章首先介绍了 HTTP 协议、静态网页和动态网页等 Web 基础知识，并简单介绍了什么是 ASP.NET，ASP.NET 的发展历史、ASP.NET 与 ASP 的比较，以及 ASP.NET 的工作原理；然后讲解了运行 ASP.NET 的配置需求；最后编写了本书的第一个 ASP.NET 程序，这是一个常见的 Hello World 程序，让读者对 ASP.NET 程序有一个直观的认识，了解如何加入 ASP.NET 代码到一个 HTML 网页中去。通过这些讲解，读者可以尽快地掌握一些关键的基础知识，对 ASP.NET 有一个概念性的了解，为后面进行 ASP.NET 程序设计打下基础。

第 2 章主要介绍了 C#的语法基础。在进行 ASP.NET 程序设计之前，首先要选择一种 ASP.NET 编程语言。C#是基于 C 和 C++创建的一种新的面向对象的语言，对面向对象程序设计提供全方位的支持。因此，使用 C#可以完全实现类的创建，并提供封装性、多态性

和继承性。本章介绍了 C#的初步知识，包括为什么选择 C#，常量、变量和表达式，分支和循环，以及过程和函数等基本语法。

第 3 章讲解了由 ASP.NET 扩展了的 HTML 控件和 Web 内部控件，并着重区分 Web 表单和一般表单。本章讲述的控件在 ASP.NET 程序设计过程中发挥了重要作用。针对早期的 ASP 版本，ASP.NET 所使用的控件都是在服务器端运行的，并且在实现上比 ASP 要方便得多。

第 4 章介绍了 ASP.NET 中常用的内置对象，包括 Response 对象、Request 对象、Application 对象、Session 对象、Server 对象，并讲解了环境变量的使用和 Cookie 的使用。所有的这些对象在 ASP 中都存在，ASP.NET 也保留了这些内置对象。Cookie 是在用户机器的硬盘中保存的小文本，用于存储一些需要保存的数据，这样可以节省服务器端的资源。本章中介绍了 Cookie 的基础知识、设置和检索 Cookie、检查用户的浏览器是否禁用了 Cookie。读者通过第 3 章和第 4 章的学习，可以开始进行基本的 ASP.NET 程序设计，并可以根据本书配备的例子来深入了解 ASP.NET 程序设计方法。

第 5 章介绍了 ASP.NET 的高级编程方法，重点讲解了命名空间的概念、类库以及.NET 中常用类库的使用方法；使用 Page 类，以及使用事件驱动的方式进行程序设计，同时讲解了 ASP.NET 中常用的 Page 事件。

第 6 章介绍了其他的 Web 控件，包括 Web 增强控件和自定义控件。增强的 Web 控件有日历控件、广告控件以及 6 个验证控件，这些控件是 ASP.NET 新增加的控件，为程序设计提供了极大的方便。另外，使用自定义控件可以获得最大的程序设计自由，并且可以把代码隐藏在 dll 文件中。

第 7 章介绍了使用 ADO.NET 进行数据库访问的方法。主要讲解了 ADO.NET 相对于 ADO 的优势、ADO.NET 的使用方法，以及在数据库应用中常用的 Web 控件(DataGrid，DataList，Repeater)。与 ADO 相比，它更容易实现数据共享，提高了标准化程度并使可编程性大大增强。同时，从效率上讲，使用 ADO.NET 将大大提高程序性能。

第 8 章介绍了如何在 ASP.NET 中使用 XML。XML 是 Extensible Markup Language(可扩展标记语言)的缩写，它提供了一种独立于应用程序的格式来保存数据，并可以通过这种格式很容易地在不同应用程序之间实现数据共享。现在，.NET 也把 XML 作为.NET 应用程序传递数据的一种主要的方法。本章主要讲解了什么是 XML，XML 的格式以及如何在 ASP.NET 中使用 XML 存储和管理数据。

第 9 章介绍了 ASP.NET 的配置和优化方法。ASP.NET 的所有配置文件都是 XML 格式的文件，因此采用这种方式进行 ASP.NET 配置有方便和灵活的优点。另外，所有的 ASP.NET 配置都是可以随时更改的。也就是说，在一个应用程序的运行期间，可以随时增加和删除 ASP.NET 配置文件中的项目，在修改成功后，可以立即激活使用，并不会影响服务器的效率。这与有些系统当配置发生变化时需要服务器重新启动才能使新配置生效相比较有非常大的优势。本章主要讲解了两个 config 文件(machine.config 和 web.config)的编写方法以及作用，并讲解了如何使用 global.asax。另外，为了能够让 ASP.NET 应用程序以更快更优的方式运行，除了使用一定的程序设计技巧，还可以采用一些选项来提高应用程

序的执行速度。本章重点讲解了使用缓存、跟踪和监视 ASP.NET 进程的方法。

第 10 章介绍了 ASP.NET 中如何实现安全性。安全性是对用户的身份进行验证，并对通过验证的用户按照对其授予的访问权限，来确定此用户是否可以访问某种资源的一个过程。本章重点讲解了实现身份验证和授权的几种方法，尤其是基于 Forms 的身份验证方法具有界面美观、实现方便的特点。除此之外，还简单介绍了如何使用 SSL 进行加密。

第 11 章介绍了如何在 ASP.NET 环境中实现 Web Service。其中包括创建 Web Service 的方法、使用 Web Service 的方法和在网上寻找 Web Service 的方法。

附录为使用本书作为教材的网络程序设计课程提供了一个完整的课程设计题目。此课程设计要求学生分组并分工合作完成一个完整的网站。题目中除了包含网站的各个栏目的设计实现之外，还包括后台管理功能的设计和实现。

2. 本书适用读者

本书适合作为高等院校计算机与信息技术及相关专业的教材，也适用于在.NET 框架下开发 Web 程序的设计人员参考使用。对于希望从基本概念开始学习的 Web 程序爱好者来说，本书也有详细的例子可以边学习边实践。

3. 使用本书的要求

运行本书中的所有例子需要建立下面的环境：

(1) Windows NT/2000/XP 操作系统。

(2) IIS 5.0 或者 5.0 以上的版本。

(3) Internet Explorer 6.0 或者更高版本。

(4) Microsoft SQL Server 2000 数据库。

(5) 在服务器中创建一个文件夹，原样建立本书示例程序。

(6) 设置 IIS 中默认 Web 站点的主目录属性页，将它的本地路径设置为刚刚建立的保存了所有例子程序的目录。

(7) 启动 Web 浏览器，在“地址”栏中输入 http://localhost/后面加入要运行的例子程序的路径和文件名。

本书由金雪云编写。在编写过程中受到来自各方面人士的帮助和支持，在这里表示衷心的感谢。由于时间仓促，本书不足之处敬请各位读者批评指正。

金雪云

目　录

第1章　ASP.NET概述

随着网络的不断普及，人们在学习、工作和生活中已经越来越离不开网络了。尤其是Internet，它作为一种跨平台的广域网，不需要用户在自己的机器上装载任何其他的软件，只要有一个浏览器，就可以浏览到各种各样的信息，享受各种各样的服务。随着网络接入技术的不断发展，越来越多的单位和个人都开始考虑创建Web应用程序，这其中最大的需求就是创建网站。除了美观之外，对于所有的Web程序设计人员来说，都希望自己制作的Web站点不仅美观，而且信息量丰富，功能强大。使用传统的HTML只能保证网页的美观，却不能引入更多更强大的功能，时间一长，用户就失去了兴趣。因此，采用动态网页设计成为现在网站设计的主流，ASP就是在这种情况下诞生的。

但是，随着时间的推移，程序设计人员发现，ASP一方面为网站的设计者带来了方便，一方面也使得网站的各种代码难于管理。对于程序员来说，当需要改动大量的HTML代码和VBScript、JavaScript代码混合在一起的程序时，往往需要花费很大精力甚至需要写大量的代码才能达到目的。为了解决这些问题，ASP.NET诞生了。

ASP.NET为Web应用程序的开发提供了许多新的特性，从而使Web应用程序的开发人员可以摆脱ASP环境中的种种束缚，使得在.NET平台上开发Web应用程序与在.NET环境中开发其他应用程序没有任何的区别。

【本章教学目标】

(1) 了解Web基础知识
(2) 了解ASP.NET的发展过程
(3) 了解ASP.NET的优势
(4) 建立ASP.NET的运行环境
(5) 了解编写ASP.NET程序的方法

1.1　Web基础知识

要了解如何利用ASP.NET环境建立Web应用程序，首先必须了解一些Web基础知识。

1.1.1　HTTP协议

HTTP协议即超文本传输协议(Hypertext Transfer Protocol)，是在Internet中进行信息传

送的协议。浏览器默认使用这个协议在 Web 上进行搜索。当用户在浏览器的地址栏中输入 URL 字符串 www.sina.com.cn 时，浏览器会自动使用 HTTP 协议来搜索 http://www.sina.com.cn 网站的首页。

从浏览器向 Web 服务器发出搜索某个 Web 网页的请求是 HTTP 请求。当 Web 服务器收到这个请求之后，就会按照请求的要求，去找相应的网页。如果找到这个网页，就把网页的 HTML 代码通过网络传回给发出 HTTP 请求的浏览器；如果没有找到这个网页，就发送一个错误信息给发出 HTTP 请求的浏览器。响应 HTTP 请求的所有这些操作称为 HTTP 响应。

HTTP 协议是无状态协议。也就是说，当使用这种协议时，所有的请求都是为搜索某一个特定的 Web 网页而发出的。它不知道现在的请求是第一次发出还是已经多次发出，也不知道这个请求的发送来源。当用户请求一个 Web 网页时，浏览器会与相关的 Web 服务器相连接，检索到需要的页面后，会立即把浏览器与服务器的连接断开。

从程序设计的角度来看，无状态的特点对于 HTTP 来说是一个缺点，这使得某些功能很难实现。但是由于网络本身的特点，HTTP 只能是无状态的。因为如果 HTTP 协议是有状态的协议，那么一个连接就会长时间地存在下去，这样才能判断一个用户到底使用了多长时间，在这段时间内都做了些什么事情。在 Internet 环境中，一个 Web 服务器要保存太多的连接(因为在 Internet 环境中，用户的数量是很难估计的)，最终导致服务器瘫痪。正因如此，对于所有的 HTTP 请求，Web 服务器都会以同样的方式来对待，并不会长时间地保存连接。

1.1.2 Web 服务器

Web 服务器并不是一台物理的机器，而是一种软件，可以管理各种 Web 文件，并为发出 HTTP 请求的浏览器提供 HTTP 响应。通常情况下，Web 服务器和浏览器分别处于不同的机器中，但是在某些特定情况下也可以让它们并存在同一机器上。例如要学习本书的例子就可以在一台机器上建立一个 Web 服务器，在它上面编写应用程序，然后通过同一台机器的浏览器察看这些应用程序的执行结果。

比较常见的 Web 服务器有 Apache 和 IIS(Internet Information Service)。由于 ASP.NET 只能在 IIS 上运行，所以本书把介绍重点放在 IIS 上。IIS 是 Microsoft 公司的 Windows 2000/XP 操作系统所提供的 Web 服务器。在 1.3.1 中会详细介绍 IIS。

1.1.3 静态网页

在动态网页出现之前，所有的网页都是静态的。静态网页是用纯 HTML 代码编写的网页，其代码是用一些编辑器输入的，或者是用一些网页设计工具生成的，保存为.html 或.htm 文件的形式。由于静态网页中没有任何与用户相关的部分，所以在设计完成之后，无论是哪个用户、在什么时候、以何种方式访问该网页，它的内容都不会有任何不同。

例 1.1　(01-01.htm)一个简单的静态网页，用于显示一个红色的"hello world."字符串，运行结果如图 1-1 所示。

```
1:    <html>
2:    <body>
3:    <font color=red>hello world.<br></font>
4:    </body>
5:    </html>
```

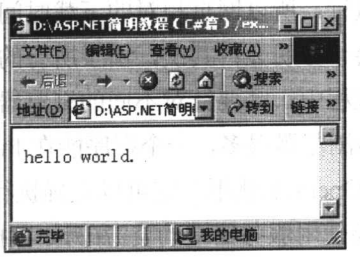


图 1-1　例 1.1 的运行结果

本书约定：

本书中所有例子都以上面的方式给出。代码前面的数码和冒号并不是代码的内容而是行号，使用它们的目的是为了更清楚地进行代码分析。

上面的例子是一个最简单的网页，目的是显示红色的"Hello World."字符串。这段代码是用 HTML 语言编写的。与 XML 类似，HTML 语言也是一种标记语言，而且是一种最简单的标记语言，因为在这种语言中，所有标记的含义都是确定的，不能根据用户的需求进行任何改变。例如在这段代码中就使用了 HTML 语言中最常见的一些标记，如<html>，</html>，<body>，</body>，，等。如果读者对 HTML 语言不熟悉的话，请参考相关的书籍，本书不做详细介绍。

过多地采用静态网页会导致很大的局限性。如果希望为用户显示一些个性化的信息，使用静态网页就无法达到目的。例如，用用户名和密码来识别用户身份；在特定的时间(如用户的生日)为用户发送 e-mail 祝贺等，这些功能在静态网页中是无法实现的。除此之外，静态网页还无法防止用户复制 HTML 代码，因为每个用户都可以采用浏览器的“查看源文件”命令看到网页的所有 HTML 代码。

1.1.4　动态网页

动态网页可以为不同的用户提供个性化的服务，这种动态性是通过程序设计实现的。随着技术的不断发展，在动态网页的实现过程中，有两种常用的程序设计方法——客户端程序设计和服务器端程序设计。

客户端程序设计是通过将程序下载到浏览器上来完成其动态特性的。通常情况是程序

员把客户端代码(即可以实现动态内容的程序)编写到 HTML 文件中，当用户发出对这个网页的请求时，客户端代码和 HTML 文件的代码一起以响应的方式返回给发出请求的浏览器。由于所有的代码(包括程序和 HTML 标记和客户端代码等)都被浏览器接收，所以这些程序的执行是由浏览器来实现的。常见的客户端编程技术有 JavaScript，VBScript 和 Java applet 等。

使用客户端程序设计的方法来实现网页的动态服务可以减少服务器的负担，充分利用客户端机器的资源。但是现在客户端程序设计技术已经越来越不受欢迎。首先，由于所有的代码都要下载到客户端来执行，所以网页相对的下载时间就会增加，尤其当程序的代码量很大时，下载时间的延长会十分明显；其次，由于所有的客户端代码都是由浏览器来执行的，所以，在程序编制的过程中，需要针对不同的浏览器进行编码和测试，以保证代码的正确执行，因为现在流行的浏览器很多，一个程序能在 Internet Explorer 上正确执行，但是在 Netscape Navigator 或者 Opera 上就不一定可以正确执行，这为程序的快速编制设置了很大的障碍；第三，如果需要使用服务器端的资源(例如数据库中的数据)，那么采用客户端程序设计无法实现；第四，采用客户端程序设计无法保证代码的安全，因为所有可以访问到这个网页的用户都可以采用浏览器的“查看源文件”命令来看到网页的所有代码(包括 HTML 代码和客户端程序)。

由于客户端程序设计有这么多的缺点，而现在的服务器的硬件速度又越来越快，相对可以使用的资源也就越来越多，使得客户端程序设计的节省服务器端资源的优势已经大大丧失，所以服务器端程序设计已经渐渐成为动态网页程序设计的主流。

服务器端程序设计的原理是：程序员编写的代码被保存在服务器上，当用户对某个动态网页发出 HTTP 请求时，这个请求所要访问的网页的代码都在服务器端执行完成，并把执行结果以 HTML 的形式传回浏览器。这样，由于浏览器接收到的只是程序执行的结果，所以上面提到的所有问题都可以迎刃而解。

1.2 ASP.NET 简介

1.2.1 基本概念

ASP(Active Server Pages)是一种功能强大而且易于学习的服务器端的脚本编程环境。它是 Microsoft 公司的产品，从 Windows NT Server 操作系统开始就附带这种脚本编程环境。并且，在 Windows NT Workstation、Windows 98 和 Windows 2000 中也都附带这个脚本编程环境。2001 年，Microsoft 在前面 3 个版本的 ASP 基础上，推出了全新的 ASP.NET，它使用 Visual Basic .NET 作为默认语言。使用这种环境，可以方便地创建动态、快速、交互性强的 Web 站点。

下面介绍一些在 ASP.NET 环境中工作需要了解的基本概念。

1. 解释和编译

机器能够执行的只是二进制代码，所以所有用助记符编写的程序都需要把程序语句翻译成机器可以执行的二进制代码才能真正执行。如果这个翻译的过程是在程序执行之前预先进行的，那么就是编译；如果这个翻译的过程是在程序的执行过程中进行的，那么就是解释。编译发生在程序执行之前，因此，对代码进行优化来保证编译的结果可以最好地利用机器硬件的各种性能。而解释是在程序的执行过程中进行的，所以没有办法对程序进行相关的优化。

早期的 ASP 是 IIS 的一种开放式的无需进行编译的应用程序环境。也就是说，ASP 程序是解释执行的。IIS 是服务器上安装的 Internet 信息服务器(Internet Information Server)，它是 Microsoft 公司开发的一个网络文件和应用程序服务器(即 Web 服务器)，包含在操作系统中。在 Windows 2000 中，它的版本是 5.0。IIS 支持 HTTP、FTP、和 Gopher 协议。由于 ASP 是服务器端的脚本编程环境，而所有的程序都是解释执行，这意味着在这个环境中的所有程序在每次被访问时，都需要 IIS 进行一次解释，客户端才会得到一个执行结果。

在 ASP.NET 中，所有的程序在执行前都是经过服务器编译的。在这一点上，ASP.NET 与早期的 ASP 版本有很大的不同，因此在程序执行的效率上也有很大的提高。具体的方法是在 ASP.NET 中，所有程序仍然保存在服务器端，当一个程序第一次被执行时进行编译，然后执行并返回执行结果；当这个程序被再次执行时，会直接在服务器上执行已编译好的可执行代码，然后把结果通过网络返回给客户端。因此，与 ASP 相比，ASP.NET 程序的执行速度会快很多。

2. HTTP 请求和 HTTP 响应

一个 HTTP 请求的处理过程如图 1-2 所示。

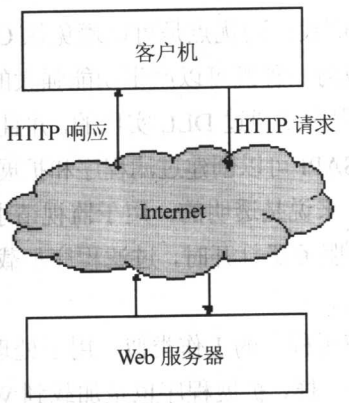


图 1-2　一个请求的处理过程

具体的处理过程是：在客户机中，有一个用于浏览网页的浏览器，用户在这个浏览器中发出HTTP请求(例如用户输入的URL或者单击一个超级链接)。HTTP请求通过Internet找到请求的 Web 服务器，并把请求传给此服务器的处理模块，由该模块负责找到相应的

ASP 程序或 ASP.NET 程序并执行(如果是 ASP 程序，通过 asp.dll 的 ISAPI DLL 进行解释执行；如果是 ASP.NET 程序，则通过 aspnet_isapi.dll 进行编译执行)。执行结果通过 Internet 返回给客户端，形成 HTTP 响应。

不论是 ASP 文件还是 ASP.NET 文件都是一个可以用任何文本编辑器编辑的纯文本文件，只要将这个文件的扩展名保存为 asp(ASP 程序)或者 aspx(ASP.NET 程序)就可以了。在进行 ASP.NET 程序开发时，可以选择使用记事本这样简单的纯文本的编辑器作为开发工具，也可以使用 Visual Studio.NET 进行开发。

3. 选择 ASP.NET 的原因

现在进行动态 Web 程序设计的技术有很多，常用的有 CGI，ISAPI，PHP，JSP，ASP 和 ASP.NET。

(1) CGI

CGI(Common Gateway Interface，公共网关接口)是在服务器上创建脚本的一种机制。几乎所有的 Web 服务器都支持 CGI。在 UNIX 系统上，大部分的 CGI 程序都是用 Perl 语言和 C 语言编写的。在 IIS 上，用 CGI 程序可以调用操作系统所提供的所有功能，并使用 ODBC 建立数据库连接来使用数据库功能。

但是，从开发人员的角度来看，这些应用程序在维护和调试时都十分困难。并且，从 Web 服务器的性能上来看，使用 CGI 还有一个更大的缺点：对于每一个客户的请求，CGI 都要产生一个进程来进行处理。这样，当服务器的访问量很大时，系统内部用于处理用户响应的进程就十分多，这么多的进程会让服务器的资源很快地消耗掉，从而导致服务器崩溃。

因此，使用 CGI 的优点是可以创建功能强大的应用程序；缺点是难以维护和调试，系统资源消耗大。

(2) ISAPI

使用 ISAPI 进行 Web 程序设计的优点是可以避免像 CGI 程序那样，对每一个客户请求都产生一个进程，所以它成为一种既可以产生功能强大的应用程序，又能减少性能损失的程序设计环境。ISAPI 应用程序是通过 DLL 实现的，并且可以加载到服务器的进程空间，以保证程序执行得更快。用 ISAPI 可以创建过滤程序和扩展程序两种不同的应用程序。

ISAPI 过滤程序对于客户来说是透明的，用于监视请求、自定义身份验证方案、随机数据转换以及更多的事情。在服务器打开时，过滤程序加载到 Web 的进程空间，并驻留在内存中，直到服务器关闭为止。

ISAPI 扩展程序与 CGI 应用程序的工作类似，用于处理表单、从数据库中检索数据、执行业务逻辑等。与过滤程序一样，扩展程序也是加载到 Web 服务器的进程空间，但是只在第一个用户从扩展程序请求服务时才发生，而不是在服务器启动时发生。扩展程序也可以用来进行操作系统级别的访问。

由于 ISAPI 的 DLL 是被加载到 Web 服务器的进程空间的，所以这种方法的最大缺点是，如果编写 ISAPI 应用程序时对性能和效率没有进行很好的考虑，就会导致服务器的崩溃。另一个缺点是它仍然要使用 C 语言来进行编写。这对于程序员来说，调试和维护仍然

不容易。

(3) PHP

PHP 起源于个人主页页面(Personal Home Pages)，现在指 PHP 超文本与处理程序(Hypertext Preprocessor)。当用户打开一个页面时，由服务器处理 PHP 指令，然后将执行结果返回给浏览器。PHP 是开放源码并且是跨平台的，可以在 Windows，UNIX 和 Apache Web 服务器上运行。但是，它需要单独进行下载并且经过复杂的安装才能够使用。PHP 语言类似于 C，所以不容易调试和维护仍然是它最大的缺点。

(4) ASP

ASP 是一种允许用户将 HTML 或 XML 标记与 VBScript 代码或者 JavaScript 代码相结合生成动态页面的技术，当一个页面被访问时，VBScript/JavaScript 代码首先被服务器处理，然后将处理后得到的 HTML 代码送还给浏览器。

使用 ASP 的优点是可以建立强大的应用程序，而且实现的效率相对很高；也很容易建立数据库连接，实现数据库访问；对于第三方开发人员，还可以开发自己的自定义控件来扩展它的功能。

ASP 只能建立在 Windows 的 IIS Web 服务器上。由于所有的代码都是解释执行的，所以相对速度较慢，并且无法有效地利用机器硬件的各种性能；VBScript/JavaScript 代码的结构性不好，会导致代码不容易理解。

(5) JSP

与 ASP 类似，JSP 也是一种允许用户将 HTML 或 XML 标记与 Java 代码相结合生成动态页面的技术，很多的 Web 服务器都支持 JSP，因此可以实现不同服务器中代码的兼容。由于 JSP 使用 Java 语法，所以 Java 程序员很容易掌握这种编程技术。

(6) ASP.NET

ASP.NET 与 ASP 相比效率更高，提供了很高的可重用性，并且对于实现同样的功能时比使用 ASP 的代码量要小得多。另外，ASP.NET 程序是编译执行的，可以更好地优化和利用服务器硬件的性能。所以说，ASP.NET 采用全新的编程环境，代表了技术发展的主流方向。

1.2.2　ASP.NET 的发展历史

要谈 ASP.NET 的发展历史，必须从它的前身 ASP 谈起。

1996 年 ASP 1.0 的诞生给 Web 应用程序开发带来了福音。早期的 Web 程序开发是十分繁琐的，以至于要制作一个简单的动态页面需要编写大量的 C 代码才能完成，这对于普通的程序员来说太难了。而 ASP 却允许使用 VBScript/JavaScript 这样的简单脚本语言编写嵌入在 HTML 网页中的代码，进行 Web 程序设计。在进行程序设计时，可以使用它的内部组件来实现一些高级功能(例如 Cookie)。ASP 的最大贡献在于它的 ADO(ActiveX Data Object)组件，使得程序对数据库的操作十分简单，从而使动态网页设计也变成一件轻松的事情。

到了 1998 年，Microsoft 发布了 ASP 2.0，是 Windows NT4 Option Pack 的一部分，作

为 IIS 4.0 的外接式附件。它与 ASP 1.0 的主要区别在于其外部组件是可以初始化的，这样，在 ASP 程序内部的所有组件都有了独立的内存空间，并可以进行事务处理。

到了 2000 年，随着 Windows 2000 的成功发布，IIS 5.0 所附带的 ASP 3.0 也开始被广泛使用。与 ASP 2.0 相比，ASP 3.0 的优势在于使用了 COM+，因此效率会比以前的版本更高，并且更稳定。

2001 年，ASP.NET 出现了。在研发时，它的名字是 ASP+。为了与微软的.NET 计划相匹配，并且要表明这个 ASP 版本并不是对 ASP 3.0 的补充，微软将其命名为 ASP.NET，版本号是 1.0。ASP.NET 在结构上与前面的版本大相径庭，几乎完全是基于组件和模块化的，Web 应用程序的开发人员使用这个开发环境可以实现更加模块化的、功能更强大的应用程序。现在，ASP.NET 2005 修正了以前版本中的一些 Bug，在移动应用开发，代码安全以及数据库 Oracle 和 ODBC 的支持等方面都做了很多的改进。

1.2.3 ASP.NET 与 ASP 的区别

从更深的层次来研究 ASP.NET，会发现 ASP.NET 与 ASP 之间的区别主要在以下几个方面。

(1) 效率

ASP 是一个脚本编程环境，只能用 VBScript 或者 JavaScript 这样的非模块化的脚本语言来编写，其程序在每次请求时解释执行。这就意味着它在使用其他语言编写大量组件时会遇到困难，并且无法实现对操作系统的底层操作。由于它是解释执行的，速度和效率要远远低于编译执行的程序。而 ASP.NET 则是建立在.NET Framework 之上的，它可以使用 Visual Basic、C#这样的模块化程序设计语言，并且在第一次执行时进行编译，之后不需要重新编译就可以直接运行，其速度和效率比 ASP 程序提高很多。

(2) 可重用性

编写 ASP 应用程序时，ASP 代码和 HTML 混合在一起。只要有需要，就可以在任意的一个位置插入一段代码来实现期望的功能。这种方法表面上看起来很方便，但在实际工作中会产生大量代码繁琐的页面，很难读懂，导致代码维护困难。虽然可以使用 include 指令尽量让程序模块化，但是仍然不是一个最终彻底的解决方案。而 ASP.NET 则可以实现代码和内容的完全分离，使前面提到的问题迎刃而解。

(3) 代码量

ASP 需要对所有要实现的功能通过编写代码来实现。例如，所有的 ASP 程序员都遇到过这样的情况：为了保证一个用户数据提交页面的友好性，当输入错误时会显示错误的位置，并尽量把用户原来的输入在控件中显示出来。这样的一个应用需要程序员编写大量的代码来实现。而在 ASP.NET 中只要程序员预先说明，就可以利用 ASP.NET 提供的验证控件自动实现。所以相对来说，要实现同样的功能，使用 ASP.NET 比使用 ASP 的代码量要小得多。

从上面的分析中可以得知，ASP.NET 相对于 ASP 的优势有以下几点：

- 代码更清晰。