21世纪 高等学校电子信息类专业规划教材

ASP.NET程序设计教程

田 原 沈成涛 李文波 编 著

清华大学出版社
http://www.tup.tsinghua.edu.cn

北京交通大学出版社
http://press.bjtu.edu.cn

21 世纪高等学校电子信息类专业规划教材

ASP. NET 程序设计教程

田　原　沈成涛　李文波　编著

清华大学出版社
北京交通大学出版社
· 北京 ·

内 容 简 介

本书介绍了使用 ASP. NET 创建动态 Web 网页的相关技术。主要内容有：ASP. NET 基础知识、运行 ASP. NET 程序的环境需求、HTML 语言基础、VB. NET 语言基础、利用 ASP. NET 建立 Web 页面、ASP. NET 常用内置对象、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 ASP. NET 的安全访问控制。最后介绍了一个网站的聊天室系统实例。本书在介绍各个知识点时，知识介绍浅显易懂，并且辅以大量的实例，突出了实践性，加强了读者动手能力。

本书结构清晰，内容丰富，可作为本科、大专层次院校的教材，也可作为工程师和程序设计爱好者的学习参考用书，并可作为各类培训班的培训用书。

图书在版编目(CIP)数据

ASP. NET 程序设计教程 / 田原，沈成涛，李文波编著. —北京：清华大学出版社；北京交通大学出版社，2006. 5
(21 世纪高等学校电子信息类专业规划教材)
ISBN 7 - 81082 - 759 - 6

Ⅰ. A… Ⅱ. ①田… ②沈… ③李… Ⅲ. 主页制作 - 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(2006)第 043472 号

责任编辑：杨祎　　特邀编辑：吴炳林
出 版 者：清 华 大 学 出 版 社　邮编：100084　电话：010 - 62776969
　　北京交通大学出版社　邮编：100044　电话：010 - 51686414
印 刷 者：北京鑫海金澳胶印有限公司
发 行 者：新华书店总店北京发行所
开　本：185 × 260　印张：19. 25　字数：465 千字
版　次：2006 年 6 月第 1 版　2006 年 6 月第 1 次印刷
书　号：ISBN 7 - 81082 - 759 - 6/TP · 277
印　张：1 ~ 4 000 册　定价：28. 00 元

前　言

ASP. NET 是 Microsoft 公司动态服务器页面(简称为 ASP)的最新版本,它是一种基于服务器的功能强大的技术,用于为万维网站点或企业的内部网创建动态的、交互式的 HTML 页。

ASP. NET 也构成了 Microsoft 的 . NET Framework 的核心元素,可以对功能强大的 . NET 开发环境提供基于 Web 的访问。它允许以一种非常灵活的新方式创建 Web 应用程序,把常用的代码封装到种种面向对象的控件中,这些控件可以由站点用户引发的事件来触发。

虽然从名字上看起来,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 程序设计的方式。与 ASP 相比,ASP. NET 有三个方面的显著优势:首先,使得网站的各种代码易于管理,提高了程序的模块化和可重用性;其次,使用高级语言(例如 VB. NET、C#)编写程序,使得很多功能可以轻松地实现;第三,ASP. NET 程序是编译执行的,执行效率高。为了改善网络应用程序的效率,随着微软 . 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。

目前市面上有不少的 ASP. NET 图书,但在教学时要找到一本很好的适合初学者入门的图书也不容易。有些图书起点太高,初学者难以明白基本概念,读者一开始学习就困难重重,从而产生厌倦心理而放弃学习;有些图书又简单得没有多少内容,读者学完后还是不会做实际的事情,不能达到一定的高度。

本书的主要内容:

在第 1 章中,简要介绍了 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,了解如何将 ASP. NET 代码加入到一个 HTML 网页中去。

在第 2 章中,简要介绍了超文本标记语言 HTML 的显示原理以及常用标记。HTML 是一种用于 Web 网页制作的排版语言,是 Web 网页最基本的构成元素。虽然现在网页制作的新技术很多,但是基本上还是架构在 HTML 之上。

在第 3 章中主要介绍了 VB. NET 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,ASP. NET 的默认语言就是 VB. NET。本章介绍了 VB . NET 的初步知识,包括为什么选择 VB. NET,常量、变量和表达式,如何实现分支和循环,以及过程和函数等基本语法。

在第 4 章中讲解如何利用 ASP. NET 建立 Web 页面。这章将介绍一些创建 ASP. NET Web 页面所需的基础知识。通过对这章的学习,使读者理解和掌握:ASP. NET 的工作原理、HTML 服务器控件及其使用、Web 服务器控件及其使用、验证控件及其使用、用户控件的创建和使用。

在第 5 章中介绍了 ASP. NET 中常用的内置对象,包括 Response、Request、Application、

Session 和 Server 对象，并讲解了环境变量的使用和 Cookie 的使用。

在第 6 章中介绍了使用 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。

在第 7 章中介绍了如何在 ASP. NET 中使用 XML。XML 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。这一章主要讲解了 XML 的基础知识以及如何在 ASP. NET 中使用 XML 存储和管理数据。

在第 8 章中介绍了 Web 服务。这一章首先学习如何把 Web 站点中的功能用作一个 Web 服务，接着讨论 Web 的其他用户如何发现这个功能，以及进行数据交换所使用的窗体。

在第 9 章中介绍了 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。

在第 10 章中介绍了 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。

在第 11 章中介绍了一个网站的聊天室系统，以此作为本书的一个大型综合性实例。

本书结构清晰，内容丰富，重点突出，语言流畅，既有理论性的阐述，也有具体的开发实例，可作为本科、大专层次院校的教材，也适用于使用 ASP. NET 进行软件开发的广大技术人员。

本书由田原、沈成涛、李文波编著。其中第 1、3、6 章由李文波编写，第 4、5 章由沈成涛编写，第 2 章、第 7 至 11 章由田原编写。田原负责审定全书内容。

由于作者水平有限，书中不妥或错误之处在所难免，敬请读者指正。

作　者

2006 年 4 月

目　录

第 1 章 ASP. NET 基础

Internet 已经成为人们生活、学习和工作中不可缺少的一部分。Internet 是跨平台的，它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，与 Internet 相连也变得越来越简单，越来越方便，以至于很多传统的行业现在也开始大大地依赖于这个神奇的网络，这在 Internet 普及之前是很难想象的。

在这种情况下，很多单位和个人都开始准备建立自己的网站。不论出于什么目的，所有人都希望自己制作的网站信息量丰富，功能尽可能强大。但是，如果仅使用 HTML 就只能保证网页的美观，却不能引入更多更强大的功能。我们看到过很多的小型网站，由于所有的页面都是静态网页，所以信息量和更新速度都没有办法得到提高，时间一长，用户就失去了兴趣。另外，由于对每一个页面都需要重新设计，所以发布新消息也变得很麻烦。既然计算机有强大的计算功能和存储功能，为什么只用它来存储一些静态网页，而不很好地使用这些功能呢？于是，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。

随着时间的推移，人们又发现，ASP 一方面为网站的设计者带来了简便，另一方面也使得网站的各种代码难于管理。对于程序员来说，面对的是大量的 HTML 代码和 VBScript、JavaScript 代码混合在一起的程序。当需要改动程序的时候，他们宁愿写新的代码，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。另外，由于 VBscript 这样的脚本语言的局限，使得很多功能都不能够轻松地实现，即使实现了，也需要写大量的代码。为了解决这些问题，ASP. NET 诞生了。

1.1 Web 基础知识

1.1.1 HTTP 协议

HTTP 协议即超文本传输协议（Hyper Text Transfer Protocol）。这个协议是在 Internet 中进行信息传送的协议，浏览器默认使用这个协议。例如当用户在浏览器的地址栏中输入 www. tom. com 的时候，浏览器会自动使用 HTTP 协议来搜索 http://www. tom. com 网站的首页。

从浏览器向 Web 服务器发出的搜索某个 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，寻找相应的网页。如果可以找到这个网页，那么就把网页的 HTML 代码通过网络传回浏览器；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。后面的这些操作称为 HTTP 响应。

HTTP 协议是无状态协议，也就是说，当使用这种协议的时候，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 网页的时候，浏览器会与相关

的 Web 服务器相连接，检索到这个页面之后，就会把这个连接断开。

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，因为这使得某些功能很难实现，但是由于网络本身的特点，这也是没有办法改变的。可以假设一下，如果 HTTP 协议是有状态的协议，那么就应该让一个连接长时间地存在下去，这样就可以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。这样在 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），会导致服务器瘫痪。正因为如此，对于所有的 HTTP 请求，Web 服务器都会以同样的方式来对待。

1.1.2　Web 服务器

当提到 Web 服务器的时候，很多人都会认为这是一台物理的机器。但实际上，Web 服务器是一种软件，可以管理各种 Web 文件，并为提出 HTTP 请求的浏览器提供 HTTP 响应。大多数情况下，Web 服务器和浏览器处于不同的机器，但是它们也可以并存在同一机器上。

比较常见的 Web 服务器有 Apache 和 IIS。由于 ASP. NET 只能在 IIS 上运行，所以本书把介绍的重点放在 IIS 上。IIS 是 Microsoft Windows 2000/XP 所提供的，在后面的章节会详细介绍 IIS。

1.1.3　静态网页

在动态网页产生之前，所有的网页都是静态的。静态网页就是用纯 HTML 代码编写的网页。这些网页的代码是用一些编辑器输入的，或者是用一些网页设计程序生成的，保存为 . html 或 . htm 文件的形式。由于这些网页中没有任何与用户相关的部分，所以设计完成之后，无论是哪个用户访问这个网页，在什么时候访问这个网页，以何种方式进入这个网页，它的样子都不会发生任何变化。

下面是一个静态网页的例子。

【例 1. 1】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。

图 1-1　运行结果

```
<html>
    <body>
        <font color=red> Hello, World! </font>
```

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

上面的例子是一个最简单的 HTML 静态网页，它的目的是显示红色的“Hello，World!”字符串。只要这个文件存在，不论什么用户要访问，在什么时候访问，以什么方式进入这个网页，都会显示同样的结果。

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，如果当前的时间是新年的开始，就可以在网页的最上面显示一个“新年好!”的信息；如果当前的时间是圣诞节，就可以在同样的位置显示一个“Merry Christmas!”的信息。除此之外，静态网页无法防止用户复制 HTML 代码，因为每个用户都可以来用浏览器的“查看源文件”命令来看到网页的 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，从前面关于静态网页的介绍中可以看出，使用这动态网页的优点是不言而喻的。而为了实现这种动态性，就需要进行程序设计。随着技术的不断发展，在动态网页的实现过程中，一般采用客户端编程和服务器端编程两种程序设计方法。

客户端编程是采用下载到浏览器上的程序来完成所有的有关动态服务的工作。通常的情况是程序员把客户端代码编写到 HTML 文件中，当用户提出对这个网页的请求时，这些客户端代码（即可以实现动态内容的程序）和 HTML 文件的代码一起以响应的方式返回给提出请求的浏览器。由于所有的代码（包括程序和 HTML 标记等）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。

在动态网页刚刚出现的时候，多数是使用客户端编程的方法来实现网页的动态服务，因为这样做可以减轻服务器的负担，充分利用客户端机器的资源。但是现在客户端编程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，所以相对而言下载的时间就会增加，尤其是程序的代码量很大的时候，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 IE 上正确执行，但是在 Netscape Navigator 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），那么采用客户端编程就无法实现。第四，采用客户端编程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）。

由于客户端编程有这么多的缺点，而现在的服务器的硬件速度又越来越快，相应可以使用的资源也就越来越多，使得客户端编程可以节省服务器端资源的优势已经大大丧失，所以服务器端编程已经渐渐成为动态网页编程的主流。服务器端编程的原理是：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提出 HTTP 请求的时候，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的问题都可以迎刃

而解。常见的服务器端编程技术有：CGI、PHP、ASP、JSP 和 ASP. NET。

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 NT Server 操作系统开始就附带这种脚本编程环境，并且，在 NT Workstation、Windows 98 和 Windows 2000/XP 中也都附带这个脚本编程环境。

微软于 2001 年在前面三个版本的 ASP 基础上，推出了全新的 ASP. NET，它开始抛弃前面三个版本都在使用的脚本语言而使用 Visual Basic. NET 作为它的默认语言。但是，无论如何变化，下面的优势依然存在，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。

1. 请求和响应

在讲解请求和响应有什么不同之前，先来了解程序的编译和解释有什么不同。

由于机器能够执行的只是二进制代码，所以所有用助记符来编写的程序都需要使用一个程序把程序语句翻译成机器能够执行的二进制代码。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执行的过程中进行的，那么就是解释。由于编译是在程序执行之前进行的，因此可以对代码进行优化，从而保证编译的结果可以最好地利用机器硬件的各种性能。而解释是在程序运行过程中进行的，所以没有办法对程序进行相关的优化。

早期的 ASP（ASP 1.0、ASP 2.0、ASP 3.0）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 Intemet 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），这个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的时候都需要 IIS 进行一次解释，从而客户端会得到一个执行结果。

在 ASP. NET 中，所有的程序执行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，所有的程序仍然是保存在服务器端的，当一个程序第一次被执行的时候进行编译，所以当这个程序被再次执行的时候会直接在服务器上执行它的已编译好的可执行二进制代码，然后把执行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。

ASP 请求的处理过程如图 1-2 所示。注意，ASP. NET 对 ASP 请求的处理过程与早期版本的 ASP 的处理有所不同，但在图中在服务器端并没有详细表示出请求的处理过程。

具体的处理过程是这样的：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输入 HTTP 请求。HTTP 请求通过 Internet 找到相应的 Web 服务器，并把这个请求传给这个服务器相应的处理模块。由这个处理模块负责找到相应的 ASP 程序或

ASP. NET程序，进行相应的执行（对于 ASP 程序来说，是通过一个名叫 asp. dll 的 ISAPI DLL 进行程序的解释；而对于 ASP. NET 程序来说，则是通过 aspnet_isapi. dll 进行处理），执行结果通过 Internet 返回给客户端，形成 HTTP 响应。

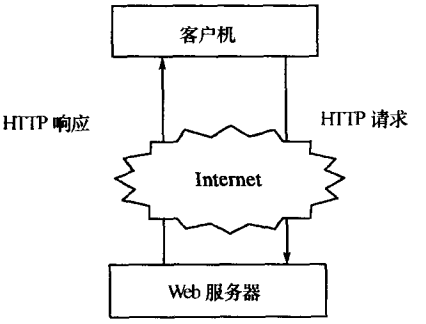

图 1-2　一个 ASP 请求的处理过程

无论是 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 aspx（ASP. NET 程序）就可以了。现在也有很多的开发工具可以进行 ASP. NET 程序的开发，比如说 Microsoft 的 Visual InterDev 等开发工具。不过也有很多程序员就选择使用记事本来开发 ASP 程序或 ASP . NET 程序。当然，对于 ASP. NET 来说，还可以使用最新的 Visual Studio. NET 进行各种控件的开发。

2. 选择 ASP. NET 的原因

现在有很多流行的技术进行动态网页的设计，下面比较一下常用的几种动态网页设计技术：CGI、ISAPI、IDC、ASP 和 ASP. NET。

（1）CGI。几乎所有的 Web 服务器都支持 CGI。在 UNIX 计算机上，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的时候都十分困难。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的时候，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，很容易导致机器的崩馈。

因此可以看出，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 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一个程序设计环境。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 可以创建两种不同的应用程序：过滤程序和扩展程序。

过滤程序对于客户来说是透明的，它用于监视请求、自定义身份验证方案、随机数据转换以及更多的事情。在服务器打开的时候，过滤程序加载到 Web 的进程空间，并一直驻留在内存中，直到服务器关闭为止。

扩展程序与 CGI 应用程序的工作类似，ISAPI 扩展程序用于处理表单、从数据库中检索数据、执行业务逻辑等。与过滤程序一样的是，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获得操作系统级别的访问。

由于 ISAAPI 的 DLL 是被加载到 Web 服务器的进程空间的，所以此种方法的最大缺点是，如果编写的 ISAPI 应用程序对性能的考虑不是很好，就会导致服务器的崩溃。它还有另外一个缺点，就是它仍然要使用 C 语言这样的十分复杂的语言来进行编写。这对于程序员来说，调试和维护仍然不容易。

(3) IDC。IDC 是 IIS 从第一个版本就有的功能，用它可以通过 ODBC 建立与数据库的连接，进行数据库的访问，并创建动态的页面把结果显示给用户。这样，使用 IDC 可以很容易地实现数据库的功能，不用创建 CGI 和 ISAPI 程序。

使用 IDC 要创建两个文件：一个查询文件和一个模板文件。查询文件用于保存对数据库访问的 SQL 语句，模板文件用于保存命令文件，它用 HTML 模板合并查询的结果。

用 IDC 进行数据库前端程序的设计很容易，但是，用它只能进行数据库编程，不能添加其他的逻辑功能。如果要进行服务器端的深入设计，就不能使用 IDC。

(4) ASP。可以说 ASP 结合了前面三种方法的所有优点：用它可以建立强大的应用程序，而且实现的效率相对很高，在这一点上它可以与 CGI 和 ISAPI 相媲美；用它也很容易建立数据库连接，实现数据库访问，在这一点上它可以与 IDC 相媲美；并且，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。

但是，在使用 ASP 进行程序设计的时候，由于 ASP 使用的是脚本语言，所有的代码都嵌入到 HTML 代码中，所以当编制功能复杂的网页时，会导致出现程序代码的可读性差的问题。另外，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。

(5) 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比使用 ASP 的代码量要小得多。另外，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 发展历史

从 1996 年 ASP 诞生到现在，ASP 发生了重大的变化，直到现在的 ASP. NET。

1996 年 ASP 1.0 诞生，它的诞生给 Web 开发界带来了福音。早期的 Web 程序开发是十分烦琐的，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有点太难了。而 ASP 却允许使用 VBscript 这种简单的脚本语言，编写嵌入在 HTML 网页中的代码。在进行程序设计的时候可以使用它的内部组件来实现一些高级功能（例如 Cookie）。它的最大的贡献在于它的 ADO（ActiveX Data Object），这个组件使得程序对数据库的操作十分简单，所以进行动态网页设计也变成一件轻松的事情。因此一夜之间，Web 程序设计不再是想像中的艰巨任务，仿佛很多人都可以一显身手。

到了 1998 年，微软发布了 ASP 2.0。它是 Windows NT 4 Option Pack 的一部分，作为 IIS 4.0 的外接式附件。它与 ASP 1.0 的主要区别在于它的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。