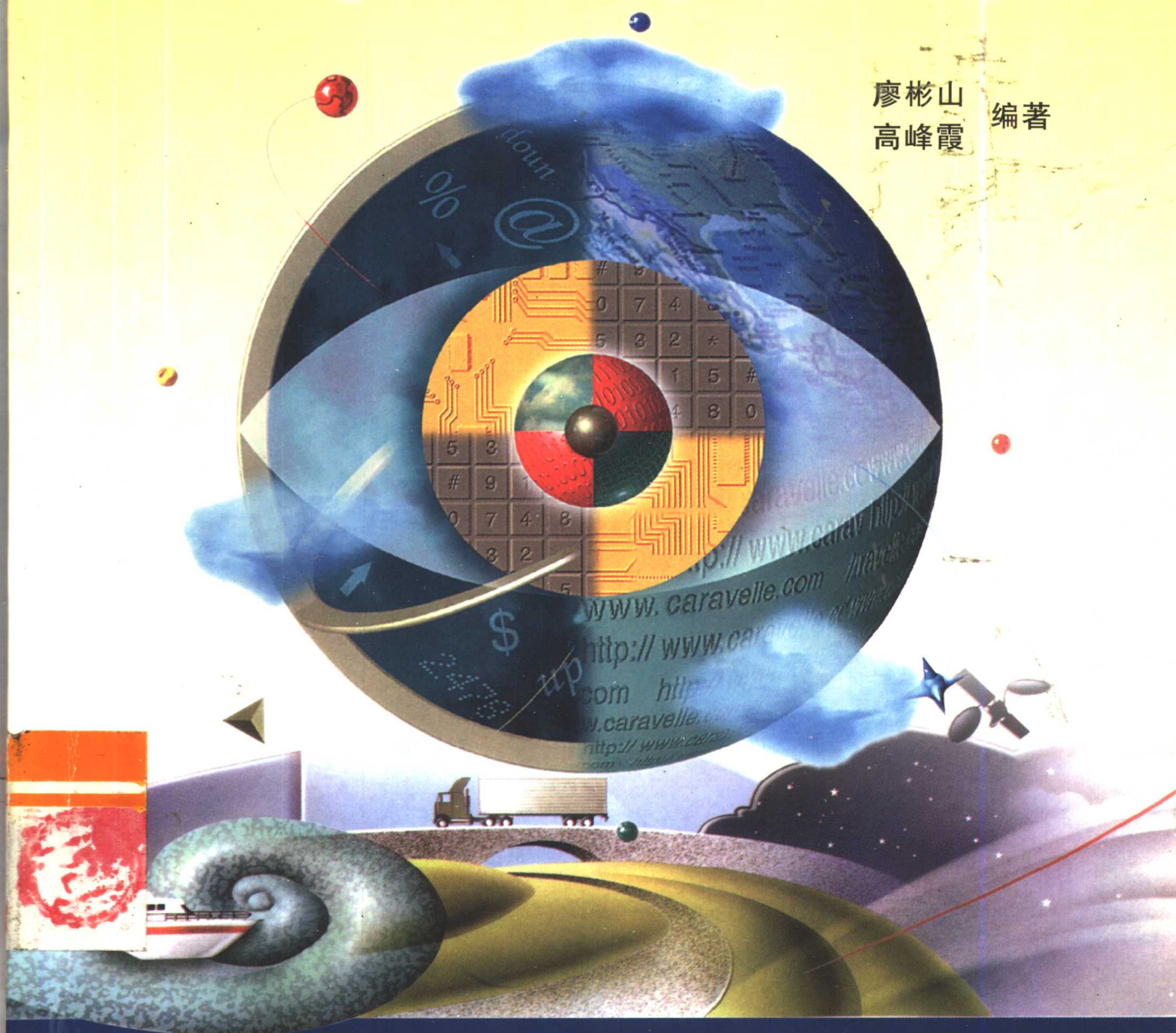


ASP

动态网站开发教程

廖彬山 编著
高峰霞



清华大学出版社

<http://www.tup.tsinghua.edu.cn>



ASP 动态网站开发教程

廖彬山 高峰霞 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

ASP 是基于 IIS 的开放式的服务器端脚本环境,它很好地将 HTML、脚本命令和组件融合在一起,集简单性、高效性和易扩展性于一体。与 CGI 相比,使用 ASP,可以更容易地创建动态、交互且高效的 Web 服务器应用程序。

本书从实用角度出发,对怎样用 ASP 进行 Web 动态网站的开发做了全面系统的介绍。全书共 9 章,从 ASP 基础开始,然后逐渐深入 ASP 的学习,同时用大量篇幅对读者十分感兴趣的 Web 数据库开发做了详细介绍。本书对 ASP 中的每个知识点、概念和难点都力求以较精炼的语言进行讲解并配以必要的实例,读者只要照书中的实例上机操作,便可全面掌握 ASP 编程的思路和开发技巧与体系,并在此基础上举一反三,根据自己的需要开发出功能强大的动态网站。本书对于广大网站开发者是一本不可多得的实用参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

书 名: ASP 动态网站开发教程

编 著 者: 廖彬山 高峰霞 编著

出 版 者: 清华大学出版社(北京 清华大学学研大厦,邮政编码:100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 宋 方 闫红梅

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京科技发行所

开 本: 787×1092 1/16 印张:19.75 字数:468 千字

版 次: 2000 年 11 月第 1 版 2000 年 11 月第 1 次印刷

书 号: ISBN 7-302-01217-2/TP·458

印 数: 0001~7000

定 价: 29.00 元

前言

ASP 是 Microsoft 开发的服务器端的脚本环境,是目前非常流行的开放式的 Web 服务器应用程序开发技术。ASP 既不是一种语言,也不是一种开发工具,而是一种技术框架,其主要功能是为生成动态、交互且高效的 Web 服务器应用程序提供一种功能强大的方法或技术。ASP 的主要特性是能够把脚本、HTML、组件和强大的 Web 数据库访问功能结合在一起,形成一个能在服务器上运行的应用程序,并把按用户的要求专门制作的 HTML 页面送给客户端浏览器。ASP 属于 ActiveX 技术中的服务器端技术,与通常在客户端实现动态页面的技术(如 Java Applet、ActiveX 控件和 VBScript 等)不同,ASP 中的命令和脚本都是在服务器端解释执行,执行后的结果产生 HTML 页面并送给浏览器,由于脚本在服务器端执行,因而开发者不必担心浏览器是否能执行脚本。同时,由于只是 HTML 页面送到浏览器,在浏览器上看不到 ASP 程序,因此可防止程序被窃取,保护了开发者的利益。另外,ASP 还具有许多显著的优点。例如,ASP 运行在 Web 服务器的同一进程中,能更快、更有效地处理客户请求;ASP 提供了更简单、更方便的数据库访问方法,使开发基于数据库驱动的 Web 应用程序更容易;ASP 支持 VBScript 和 JScript,并能以插件形式支持其他脚本语言,如 REXX 和 Perl 等。

本书按易学、易懂、易掌握的原则,结合 ASP 内部知识体系,由浅入深、循序渐进地介绍了怎样用 ASP 进行 Web 动态网站的开发。全书共 9 章和 2 个附录。第 1 章对 ASP 程序的特点、运行环境、工作流程及创建方法做了介绍;第 2 章对流行的两种脚本语言 VBScript 和 JScript 进行了讲解;第 3 章阐述了怎样用 Request 和 Response 对象来实现客户端和服务端之间的动态交互;第 4 章介绍了怎样用 Application、Session 和 Server 对象来管理应用程序和会话;第 5 章介绍了常用 ASP 服务器组件的使用,包括 Ad Rotator、Content Linker、Content Rotator、File Access 和 CDONTS 等;第 6 章以大量篇幅对读者十分感兴趣的 Web 数据库开发做了详细介绍;第 7 章介绍了 ASP 的相关知识,包括脚本调试和自定义错误信息等;第 8 章对流行的网站开发环境 Visual InterDev 做了介绍;第 9 章通过两个综合实例(聊天室和网上书店)对前面所学的知识进行了贯穿。附录部分对 IIS 5.0 和 SSI 命令做了介绍。

本书的最大特点在于对 ASP 中的每个知识点都精心编排了必要的实例。通过对实例的学习,读者会对所学知识有更深入的理解,并能更全面地掌握用 ASP 开发 Web 动态网站的思路、技巧和体系。

本书由廖彬山和高峰霞负责编写。另外,参与本书部分编写及录排工作的还有高越明、王世航、叶伟东、王志军、孙智伟、林利、章大洋和刘珊。

本书在编写过程中得到了清华大学出版社和清华松岗电脑信息有限公司的大力支持、鼓励和帮助,在此表示衷心的感谢。由于笔者学识所限,缺点和错误在所难免,竭诚欢迎广大读者批评指正。

作者

E-mail: liaobs@sohu.com

目 录

第 1 章 ASP 基础	(1)
1.1 Web 开发历史	(1)
1.2 ASP 的运行环境	(2)
1.3 ASP 的工作流程	(3)
1.4 创建 ASP 程序	(3)
1.4.1 使用 ASP 命令	(5)
1.4.2 删除白空格	(6)
1.4.3 多脚本语言混合编程	(6)
1.5 ASP 内建对象	(7)
1.6 包含文件	(8)
1.7 思考与练习	(10)
第 2 章 VBScript 和 JScript	(11)
2.1 服务器端脚本和客户端脚本	(11)
2.2 VBScript 脚本语言	(14)
2.2.1 将单行语句分成多行	(14)
2.2.2 在代码中加注释	(14)
2.2.3 使用不同进制的数字	(15)
2.2.4 数据类型	(15)
2.2.5 变量	(16)
2.2.6 常量	(19)
2.2.7 数组	(19)
2.2.8 运算	(20)
2.2.9 控制结构	(22)
2.2.10 With 语句	(26)
2.2.11 过程	(26)
2.2.12 VBScript 函数摘要	(30)
2.3 JScript 脚本语言	(33)
2.3.1 语句和语句块	(34)
2.3.2 变量	(34)
2.3.3 数据类型	(35)
2.3.4 运算符	(36)
2.3.5 数组	(38)
2.3.6 控制结构	(39)
2.3.7 函数	(42)
2.3.8 复制、传递和比较数据	(42)
2.3.9 特殊字符	(43)
2.3.10 固有对象	(43)
2.4 思考与练习	(45)

第3章 与客户端动态交互	(46)
3.1 Request 对象	(46)
3.1.1 QueryString 集合	(46)
3.1.2 Form 集合	(49)
3.1.3 ServerVariables 集合	(51)
3.2 Response 对象	(54)
3.2.1 Response 对象的方法	(54)
3.2.2 Response 对象的属性	(58)
3.3 使用 Cookie	(61)
3.3.1 什么是 Cookie	(61)
3.3.2 Cookie 的属性	(61)
3.3.3 设置 Cookie	(61)
3.3.4 获取 Cookies	(62)
3.4 思考与练习	(63)
第4章 管理会话和应用程序	(64)
4.1 Application 对象	(64)
4.2 Session 对象	(66)
4.2.1 在 Session 对象中存储数据	(66)
4.2.2 启动和结束会话	(67)
4.2.3 SessionID Cookie	(67)
4.2.4 Session 对象的事件	(68)
4.3 Global.asa 文件	(69)
4.4 Server 对象	(70)
4.4.1 设置脚本超时	(70)
4.4.2 Server 对象的方法	(71)
4.5 思考与练习	(74)
第5章 使用 ASP 服务器组件	(75)
5.1 创建组件实例	(75)
5.1.1 用 Server.CreateObject 创建对象实例	(75)
5.1.2 用<OBJECT>标记创建对象实例	(75)
5.1.3 从 Java 类中创建对象	(76)
5.1.4 设置对象的作用域	(76)
5.1.5 类型库声明	(78)
5.2 使用 Ad Rotator 组件	(79)
5.3 使用 Content Linker 组件	(81)
5.4 使用 Content Rotator 组件	(85)
5.5 使用 Browser Capabilities 组件	(86)
5.6 使用 File Access 组件	(88)
5.6.1 用 FSO 编程	(89)
5.6.2 获得驱动器的信息	(90)

5.6.3 处理文件夹	(91)
5.6.4 处理文件	(93)
5.7 收发电子邮件	(99)
5.7.1 发送电子邮件	(99)
5.7.2 接收电子邮件	(101)
5.8 其他基本组件	(102)
5.8.1 使用 Dictionary 对象	(102)
5.8.2 Permission Checker 组件	(103)
5.8.3 Page Counter 组件	(104)
5.8.4 Counters 组件	(105)
5.8.5 MyInfo 组件	(105)
5.8.6 Tools 组件	(105)
5.8.7 IIS Log 组件	(106)
5.9 思考与练习	(106)
第 6 章 用 ADO 访问数据库	(107)
6.1 数据库基础	(107)
6.1.1 概念及术语	(107)
6.1.2 SQL Server	(110)
6.1.3 结构化查询语言	(112)
6.1.4 ODBC	(116)
6.1.5 OLE DB	(120)
6.1.6 ADO 对象模型	(120)
6.2 Connection 对象	(122)
6.2.1 创建 Connection 对象	(123)
6.2.2 连接数据源	(124)
6.2.3 执行 SQL 查询	(125)
6.3 Recordset 对象	(127)
6.3.1 游标类型	(132)
6.3.2 锁定类型	(134)
6.3.3 添加记录	(135)
6.3.4 更新记录	(137)
6.3.5 删除记录	(139)
6.3.6 Supports 方法	(139)
6.3.7 Status 属性	(140)
6.3.8 分页显示	(141)
6.4 Fields 集合和 Field 对象	(143)
6.4.1 Fields 集合的属性和方法	(143)
6.4.2 Field 对象的属性	(144)
6.4.3 Field 对象的方法	(145)
6.4.4 实例——图片上传和显示	(146)
6.5 Command 对象	(149)
6.5.1 Command 对象基本用法	(150)
6.5.2 参数查询	(151)

6.5.3 Parameters 集合和 Parameter 对象	(156)
6.5.4 存储过程	(157)
6.6 事务处理	(161)
6.7 Errors 集合和 Error 对象	(162)
6.8 Properties 集合和 Property 对象	(162)
6.9 思考与练习	(163)
第 7 章 相关内容	(164)
7.1 调试 ASP 脚本	(164)
7.1.1 脚本错误	(164)
7.1.2 Microsoft 脚本调试器	(166)
7.1.3 断点调试	(167)
7.2 自定义错误消息	(170)
7.2.1 添加自定义错误消息	(171)
7.2.2 自定义 ASP 错误处理	(173)
7.3 事务	(176)
7.3.1 声明事务脚本	(176)
7.3.2 提交或中止事务	(177)
7.4 用 Visual Basic 开发 IIS 应用程序	(177)
7.4.1 Visual Basic 开发环境简介	(177)
7.4.2 IIS 应用程序的结构	(180)
7.4.3 系统要求	(180)
7.4.4 WebClass 简介	(181)
7.4.5 开发过程	(182)
7.4.6 添加模板 WebItem 到 WebClass 中	(183)
7.4.7 添加自定义 WebItem 到 WebClass 中	(184)
7.4.8 WebClass 事件	(185)
7.4.9 发送 HTML 到浏览器	(188)
7.4.10 从 HTML 表单中获取信息	(190)
7.4.11 执行文本替换	(191)
7.5 思考与练习	(193)
第 8 章 Visual InterDev	(194)
8.1 使用基础	(194)
8.1.1 主要特点	(194)
8.1.2 开发环境和工作模式	(195)
8.1.3 用户界面概览	(196)
8.2 建立 Web 工程	(199)
8.3 建立 Web 页	(202)
8.3.1 添加新页面	(202)
8.3.2 设置页面属性	(203)
8.3.3 文本与列表	(203)
8.3.4 滚动文本	(205)

8.3.5 超级链接	(205)
8.3.6 书签	(206)
8.3.7 插入图像	(206)
8.3.8 表格	(208)
8.3.9 表单	(209)
8.3.10 文档大纲	(211)
8.4 客户端脚本	(211)
8.4.1 设置脚本语言	(212)
8.4.2 事件处理	(212)
8.4.3 脚本大纲	(213)
8.4.4 事件的冒泡处理	(214)
8.4.5 使元素可编程	(216)
8.4.6 样式属性	(217)
8.4.7 文本的替换处理	(217)
8.4.8 DHTML 对象模型	(218)
8.5 样式表	(224)
8.5.1 CSS 结构和规则	(224)
8.5.2 将样式表应用到页面中	(226)
8.5.3 插入 SPAN 和 DIV	(227)
8.5.4 CSS 编辑器	(229)
8.6 滤镜和过渡	(232)
8.6.1 滤镜	(232)
8.6.2 过渡	(234)
8.7 服务器端脚本	(237)
8.8 数据环境	(237)
8.8.1 创建一个数据连接	(237)
8.8.2 创建数据命令	(239)
8.8.3 查询设计器	(241)
8.8.4 获取和显示数据库信息	(242)
8.9 思考与练习	(246)
第 9 章 综合实例	(248)
9.1 聊天室	(248)
9.2 网上书店	(269)
9.2.1 创建一个网上书目	(269)
9.2.2 添加图书到购物袋中	(272)
9.2.3 显示购物袋	(273)
附录 A IIS 5.0	(277)
A.1 IIS 管理单元	(277)
A.2 Web 站点管理	(277)
A.2.1 站点属性和属性继承	(277)
A.2.2 添加站点	(279)

A.2.3	停止、暂停或启动站点	(281)
A.2.4	主目录	(281)
A.2.5	虚拟目录	(282)
A.2.6	默认文档	(283)
A.2.7	重定向请求	(284)
A.2.8	设置 Web 站点内容的截止日期	(285)
A.2.9	为 Web 页添加页脚	(285)
A.2.10	为 Web 站点指定操作员	(286)
A.3	配置应用程序	(286)
A.3.1	创建应用程序	(286)
A.3.2	应用程序保护	(287)
A.3.3	设置应用程序映射	(287)
A.3.4	配置 ASP 应用程序	(288)
A.3.5	启用 ASP 调试	(289)
A.4	验证	(290)
A.4.1	启用验证	(292)
A.4.2	配置匿名验证	(293)
A.4.3	配置基本验证	(293)
A.5	访问控制	(293)
A.5.1	IP 地址访问限制	(294)
A.5.2	Web 服务器权限	(295)
A.5.3	设置目录和文件的 NTFS 权限	(296)
A.6	WebDAV	(297)
A.7	ASP 的新增和改进功能	(299)
附录 B	服务器端包含	(301)
B.1	# config 命令	(301)
B.2	# echo 命令	(302)
B.3	# exec 命令	(304)
B.4	# flastmod 命令	(304)
B.5	# fsize 命令	(304)

第 1 章 ASP 基础

ASP(Active Server Pages)内含于 IIS(Internet Information Server)中,是 Microsoft 开发的服务器端的脚本环境。通过 ASP,可以结合 HTML(HyperText Markup Language)、脚本命令和组件来创建动态、交互且高效的 Web 服务器应用程序。

1.1 Web 开发历史

Web(World Wide Web 或 WWW,中文称万维网)是在超文本基础上形成的庞大信息网。Web 由许多站点构成。每个 Web 站点是一组资源的集合,这些资源位于 Internet/Intranet 的一台或多台服务器上。Web 站点提供信息的基础是网页。

Web 开发是基于客户/服务器的。在这种情况下,客户是浏览器,而服务器就是 Web 服务器。浏览器和 Web 服务器通过 HTTP(HyperText Transfer Protocol)交换信息。Web 上的大多数交互均可看成请求和响应,即浏览器向 Web 服务器发出请求(一般是要求显示一个用户想看的网页),而 Web 服务器则向浏览器返回响应(通常是一个 HTML 页面或图像等)。

传统的 Web 站点大都由一个或多个静态 HTML 页面组成。所谓“静态”是指页面内容“固定不变”,当浏览器通过 HTTP 向服务器请求提供网页时,服务器只是将原先已设计好的页面传送给浏览器。静态页面仅仅使用标准的 HTML 代码,有的带有 GIF 动画。静态页面使得用户只能被动地浏览制作者提供的信息,而且不易维护。为了不断更新内容,制作者必须不断地重复制作页面。随着站点内容和信息量的日益扩增,工作量将大得超出想象。

由于静态页面缺少交互性,因此动态内容的开发就成了越来越多的 Web 站点追求的目标。所谓“动态”是指用户按自己的需要通过浏览器发出请求,服务器运行一个脚本或程序来返回包含最新信息的页面供用户浏览。

1. CGI 和 ISAPI

动态内容的实现技术有许多,较早使用的是 CGI(Common Gateway Interface)。CGI 的原理是:当用户通过浏览器向 Web 服务器发出带条件的请求时,Web 服务器将指定的条件转给 CGI 程序处理,CGI 处理完后把结果转成网页形式还给 Web 服务器,Web 服务器再把网页传送给用户。CGI 能很好地实现动态内容,但有两个致命缺点:①对每个请求,CGI 都会产生一个新的进程,同一时刻发出的请求越多,服务器产生的进程就越多,耗费掉的系统资源也越多,这样在用户访问的高峰期,网站就会表现出响应时间延长、处理缓慢的情况,严重的会导致整个网站崩溃;②创建和修改 CGI 程序相当困难,这是因为 CGI 程序没有被集成到 HTML 中,而是需要专门的语言(如 Perl 和 C)来编制。对于大多数网页开发人员来讲,要掌握和精通这些语言要花很长的时间。

ISAPI(Internet Server Application Programming Interface)能实现 CGI 的全部功能并做

了许多扩展。ISAPI 的原理大体上与 CGI 相同,都是通过交互式网页来取得客户的输入信息,然后交服务器后台处理。但 ISAPI 建立的程序是动态链接库(DLL: Dynamic Link Library),它不仅能像 CGI 程序那样被客户请求激活,而且能预先激活来监视客户输入。由于 DLL 可以和服务器装于同一地址空间中,因此执行效率比 CGI 高。

2. 脚本

用脚本实现动态内容的方法是把可执行脚本直接嵌入到 HTML 页面中去。浏览器不再需要服务器执行查询,而只要在加载 HTML 页面时执行脚本就可以了。这些脚本既可在客户端执行,也可在服务器端执行。客户端编写脚本最常用的语言是 VBScript 和 JScript,服务器端采用的是 A。ASP 既不是一种语言,也不是一种开发工具,而是 Microsoft 开发的服务器端的脚本环境。ASP 不仅提供了 CGI 的所有功能,而且具有许多显著优点:

- ASP 运行在 Web 服务器的同一个进程中,能更快、更有效地处理客户请求;
- ASP 提供了简单的数据访问方法,使开发基于数据驱动的 Web 应用程序更容易;
- ASP 带有 VBScript 和 JScript 脚本引擎,可以直接用这两种语言来编写脚本,只要用 ASP 特殊的标记〈 % 和 % 〉分隔符将脚本括起来,ASP 就能解释这些脚本。当然,如果想使用自己熟悉的其他脚本语言(如 REXX 和 Perl 等),则必须获取并安装相应的脚本引擎(脚本引擎是用于执行脚本代码的程序);
- 不必担心浏览器能否处理脚本,这是因为脚本是在服务器上而不是在客户端运行的,传送到浏览器上的 HTML 页面是在服务器上生成的;
- ASP 程序执行完后,服务器仅将运行结果返回到浏览器,这样既减轻了客户端的负担,也大大提高了交互速度,而且服务器端脚本不易被剽窃,提高了安全性;
- 编写 ASP 程序既可用任何文本编辑器(如“记事本”),也可用专用的 ASP 开发工具(如 Visual InterDev)。

1.2 ASP 的运行环境

运行 ASP 所需的环境如表 1.1 所示。

表 1.1 ASP 的运行环境

操 作 系 统	Web 服务器
Windows NT Server 4.0 或更高版本	Internet Information Server 3.0 或更高版本
Windows NT Workstation 4.0 或更高版本	Peer Web Services 3.0 或更高版本
Windows 95 或更高版本	Personal Web Server 3.0 或更高版本

注意:借助于第三方开发商提供的服务器扩展程序,也可以在 Unix、Linux 和 Apache 上执行 ASP 程序。任何一个 Web 服务器,只要内嵌 ASP 解释程序,就可以支持 ASP 编写的动态网页。一般情况下,都是在基于 Windows NT 服务器的 IIS 上运行 ASP 程序。本书的所有例子就是在 Windows 2000 的 IIS 5.0 上调试通过的。关于 IIS 5.0 的详细内容,请读者参见附录 A。

1.3 ASP 的工作流程

ASP 程序是以 .asp 为扩展名的文本文件,其控制部分是用 VBScript 和 JScript 等脚本语言来编写的。ASP 的工作流程大致是:①当浏览器从 Web 服务器上请求 .asp 文件时,服务器调用 ASP;②ASP 全面读取请求的文件,执行所有的服务器端脚本,并将脚本输出与静态 HTML 代码进行合并;③最终的 HTML 页面将在 HTTP 响应中传送给浏览器。

图 1.1 对 ASP 的工作流程做了简要说明。

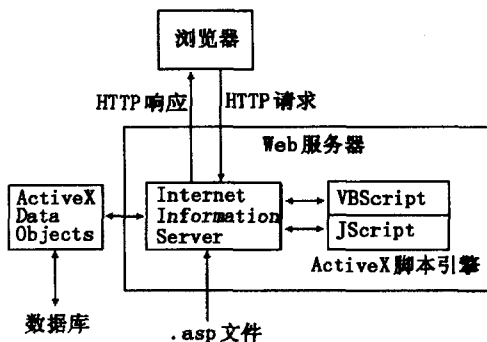


图 1.1 ASP 的工作流程

1.4 创建 ASP 程序

ASP 程序是由文本、HTML 标记和脚本组合而成的。在 ASP 程序中,脚本通过分隔符与文本和 HTML 标记区分开来。ASP 用分隔符 `<%` 和 `%>` 来包括脚本命令。由分隔符括起的命令称为主脚本命令,这些命令由主脚本语言进行处理。默认主脚本语言是 VBScript。例如,下面是一个简单的 ASP 程序:

```
<HTML><BODY>
当前时间是<% = Time() %>
</BODY></HTML>
```

用“记事本”编好以上代码后,请将其保存到 Web 站点的主目录或某个虚拟目录上,这样就可以在浏览器的地址栏中通过输入文件的 URL(Uniform Resource Locator)地址来进行访问。例如,假定 Web 站点上有个虚拟目录,该目录的别名是 aspsamples,物理位置是 c:\Scripts\AspNet,那么将上面的 ASP 程序保存到 c:\Scripts\AspNet 文件夹的 sample.asp 文件中后,在浏览器的地址栏中输入 `http://localhost/aspsamples/sample.asp` 并按回车键,便可执行该程序了。执行时,Web 服务器将 `<% = Time() %>` 替换为当前时间,然后将结果返回到浏览器(图 1.2)

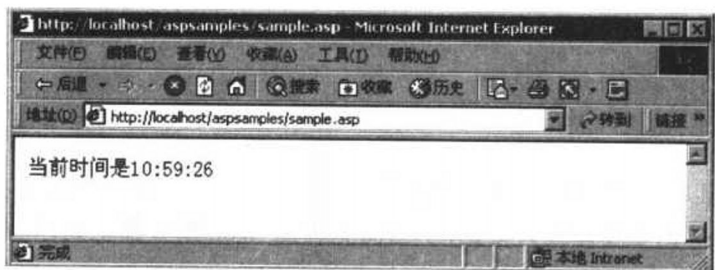


图 1.2 显示当前时间

注意:主目录和虚拟目录的有关内容,请参见附录 A。另外,访问 .asp 文件时,不能使用实际的物理路径,如不能在浏览器的地址栏中通过输入 c:\Scripts\Asp\sample.asp 来执行上面的 ASP 程序。

在 ASP 分隔符 `<%` 和 `%>` 内,可以包括主脚本语言允许的任何语句、表达式和操作符等。例如,下面给出的条件语句 `If...Then...Else` 便是常用的 VBScript 语句:

```
<HTML><BODY>
<%
If Time < #12:00:00# And Time >= #00:00:00# Then
    strGreeting = "早上好!"
ElseIf Time < #19:00:00# And Time >= #12:00:00# Then
    strGreeting = "下午好!"
Else
    strGreeting = "晚上好!"
End If
%>
<% = strGreeting %>
</BODY></HTML>
```

以上代码先根据时间段将问候语放在变量 `strGreeting` 中,再用 `<% = strGreeting %>` 命令将 `strGreeting` 变量的值发送到浏览器(图 1.3)。这样,在正午 12 点(Web 服务器所在的时区)前浏览该程序时,将看到“早上好!”;下午 7 点前浏览时,将看到“下午好!”;而晚上 7 点至 12 点浏览时,将看到“晚上好!”。

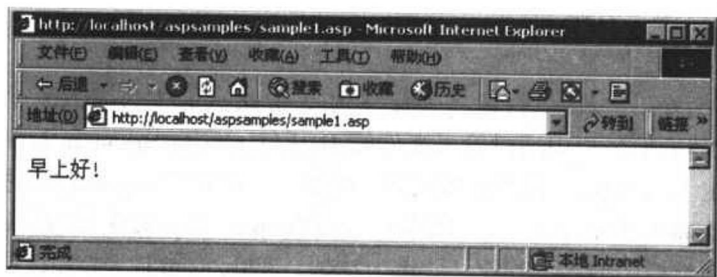


图 1.3 根据时间段显示不同的问候语

在语句的不同部分之间也可直接加入 HTML 文本,如下面的脚本在 `If...Then...Else` 语句中加入了 HTML 文本,结果与上面的脚本一样:

```
<HTML><BODY>
< % If Time < # 12:00:00 # And Time > = # 00:00:00 # Then % >
早上好!
< % ElseIf Time < # 19:00:00 # And Time > = # 12:00:00 # Then % >
下午好!
< % Else % >
晚上好!
< % End If % >
</BODY></HTML>
```

如果不想将 HTML 文本置于脚本命令之间,则可用 Response 对象的 Write 方法将文本发送到浏览器。例如:

```
<HTML><BODY>
< %
If Time < # 12:00:00 # And Time > = # 00:00:00 # Then
    Response.Write "早上好!"
ElseIf Time < # 19:00:00 # And Time > = # 12:00:00 # Then
    Response.Write "下午好!"
Else
    Response.Write "晚上好!"
End If
% >
</BODY></HTML>
```

1.4.1 使用 ASP 命令

ASP 提供了脚本语言以外的命令:输出命令和处理命令。输出命令指 < % = expression % > 显示的表达式值。例如,前面的 < % = strGreeting % > 就是用于将问候语传到浏览器的输出命令。输出命令等同于用 Response 对象的 Write 方法显示信息。

处理命令 < % @ keyword % > 为 ASP 提供处理 .asp 文件所需要的信息。例如,以下命令将 JScript 设为主脚本语言:

```
< % @ LANGUAGE = JScript % >
```

处理命令必须出现在 .asp 文件的第一行,而且 @ 和关键字之间必须加入一个空格。处理命令的关键字如表 1.2 所示。

在单个处理命令 5 中可以包含多个关键字,关键字/值对之间用空格分开,而且不能在等号左右加入空格。例如,以下命令同时设置了主脚本语言和代码页:

```
< % @ LANGUAGE = JScript CODEPAGE = 936 %
```

表 1.2 处理命令的关键字

关 键 字	说 明
LANGUAGE	设置主脚本语言。主脚本语言是用来处理分隔符〈 % 和 % 〉括起的命令的语言。默认时,主脚本语言是 VBScript。可以将任何一种具有脚本引擎的脚本语言设为主脚本语言。
CODEPAGE	设置代码页(字符编码)。
LCID	设置文件的位置表示符。
TRANSACTION	指定 .asp 文件将在事务处理环境下运行。
ENABLESESSIONSTATE	指定 .asp 文件是否使用会话语句。

1.4.2 删除白空格

对于 VBScript 和 JScript,可在起始分隔符和结束分隔符之间使用白空格(如空格、制表符、回车符和行缩进等),以使命令简单易读,但 ASP 将从命令中删除白空格。例如,以下语句均为有效且等价:

```
〈 % strGreeting = "早上好!" % 〉  
〈 % strGreeting="早上好!" % 〉  
〈 %  
strGreeting = "早上好!"  
% 〉
```

如果需要保留两个语句之间的白空格(如显示语句中变量的值时),请使用 HTML 不间断空格符()。例如:

```
〈 %  
strCur = "当前时间是"  
dtmTime = Time()  
% 〉  
〈 P〉〈 % = strCur % 〉&nbsp;〈 % = dtmTime % 〉〈 /P〉
```

注意:对于所有其他脚本语言,ASP 将保留白空格,以便这些对书写位置或缩进有所要求的语言能被正确解释。

1.4.3 多脚本语言混合编程

在同一个 .asp 文件中,除了使用主脚本语言外,还可以在页面的局部采用其他脚本语言来实现特定的功能。为了实现多脚本语言混合编程,要用到 HTML 的〈SCRIPT〉〈/SCRIPT〉标记,该标记的格式是:

```
〈SCRIPT LANGUAGE= languagename RUNAT= Server〉  
    功能实现部分  
〈/SCRIPT〉
```


其中, language 是脚本语言名称, 可以是任何脚本语言, 但 Web 服务器上必须安装有该语言的脚本引擎; RUNAT = Server 表示该段脚本在服务器端执行。若省略, 则该段脚本将发送到客户端执行。

例如:

```
< % @ LANGUAGE = VBScript % >
< %
If Time < # 12:00:00 # And Time > = # 00:00:00 # Then
    bgc = "red"      '将背景色设为红色
ElseIf Time < # 19:00:00 # And Time > = # 12:00:00 # Then
    bgc = "green"    '将背景色设为绿色
Else
    bgc = "blue"     '将背景色设为蓝色
End If
% >
< HTML > < BODY bgColor = "< % = bgc % ">
< %
printDate()        '调用 printDate()函数显示当前日期和时间
% >
< /BODY > < /HTML >
< SCRIPT LANGUAGE = JScript RUNAT = Server >
function printDate()
{
    var x
    x = new Date()
    Response.Write(x.toString())
}
< /SCRIPT >
```

以上代码执行时先根据不同的时间段为页面设置不同的背景色, 然后调用 JScript 编写的 printDate() 函数来显示当前日期和时间。

注意: 用 < SCRIPT > < /SCRIPT > 标记编写服务器端脚本时, 应注意: ① 脚本语句必须作为某个函数或过程的一部分出现, 而不能单独出现在过程或函数体之外, 因为过程或函数体之外的命令没有固定的执行顺序, 可能会导致无法预知的结果; ② 不能使用输出命令, 即不能使用 < % = 变量名 % > 向生成的 HTML 页面输出, 但可用 Response.Write 方法将内容发送到浏览器; ③ 编写时不可再用注释标记 < ! -- 和 -- > 来屏蔽, 因为服务器必须能解释脚本, 否则会引起错误; ④ VBScript 调用 JScript 函数不区分大小写。

1.5 ASP 内建对象

ASP 提供有 Request、Response、Session、Application、Server、ObjectContext 和 ASPError 等内建对象(表 1.3), 这些对象在 ASP 编程中扮演着十分重要的角色。