



动态网站编程基础

何 枫 主编

西南交通大学出版社

动态网站编程基础

何 枫 主编

西南交通大学出版社

· 成 都 ·

内 容 简 介

本书较为系统地介绍了 ASP 程序设计的基本原理、方法、安全及调试技术。重点阐述了 ASP 程序设计中的对象、组件、数据库等实用技术,并通过大量的例子来说明各种技术的具体使用方法。

本书可以作为经济信息管理、电子商务以及计算机等相关专业专科生或本科生的 ASP 动态网页程序设计课程的教材,也可以作为从事实际网络程序设计的开发人员和工作人员的学习参考书。

图书在版编目 (C I P) 数据

动态网站编程基础 / 何枫主编. — 成都: 西南交通大学出版社, 2003.5
ISBN 7-81057-703-4

I. 动... II. 何... III. 网站—程序设计
IV. TP393.092

中国版本图书馆 CIP 数据核字 (2003) 第 006738 号

动态网站编程基础

何 枫 主编

*

责任编辑 秦 薇

封面设计 肖 勤

西南交通大学出版社出版发行

(成都市二环路北一段 111 号 邮政编码: 610031 发行部电话: 87600564)

<http://press.swjtu.edu.cn>

E-mail: cbsxx@swjtu.edu.cn

四川森林印务有限责任公司印刷

*

开本: 787mm×1092mm 1/16 印张: 13.75

字数: 317 千字 印数: 1—3000 册

2003 年 5 月第 1 版 2003 年 5 月第 1 次印刷

ISBN 7-81057-703-4/TP·303

定价: 19.00 元

前 言

网络技术的迅速发展带来了信息技术的巨大变革。交互式网页编程技术也成为当今计算机程序设计技术的一个重要分支,越来越受到人们的关注。目前比较流行的交互式网页编程技术有 **Java** 程序设计、**ASP** 程序设计和 **PHP** 程序设计等,本书结合电子商务的专业特点,系统地介绍了 **ASP** 编程技术及其在电子商务中的应用。

本书为读者全面地介绍了 **ASP** 程序设计的知识。第 1 章介绍了动态网页设计的基本知识及 **ASP** 程序的基本原理及运行环境;第 2 章介绍了 **HTML** 语言的基本理论知识及应用;第 3 章介绍了 **VBScript** 在 **HTML** 中的应用及其工作原理;第 4 章介绍了 **VBScript** 与 **ASP** 的基本编程方法;第 5 章介绍了 **ASP** 编程的基本对象;第 6 章介绍了在 **ASP** 中的组件的定义和使用方法;第 7 章介绍了在 **ASP** 中如何使用数据库进行数据的查询、修改、删除等操作;第 8 章介绍了 **ASP** 程序的管理、调试、安全等操作;第 9 章介绍了 **XML** 语言与 **ASP** 的编程;第 10 章介绍了 **ASP** 在电子商务中的应用实例。

本书第 1、2、3、4、5、9、10 章由何枫编写,6、7、8 章由凯定吉编写,全书由何枫主编,尹治本教授主审。

限于作者的水平及时间,本书尚有许多不足之处,恳请读者指正。

作 者

2002 年 11 月于成都

目 录

1	1 动态网页与 ASP 概述
1	1.1 静态与动态网页概述
1	1.1.1 HTML 的起源
2	1.1.2 动态页面的起源
4	1.2 ASP 概述
4	1.2.1 什么是 ASP
5	1.2.2 ASP 的功能
5	1.2.3 使用范围
6	1.3 ASP 的环境与开发
6	1.3.1 环境的建立
7	1.3.2 开发工具
7	1.3.3 ASP 程序编写简述
9	习 题
10	2 超文本标记语言
10	2.1 概述
10	2.1.1 HTML 语言与 HTML 文件
10	2.1.2 HTML 的基本语法
11	2.2 创建网页
11	2.2.1 创建基本的网页
12	2.2.2 基本的 HTML 网页格式
14	2.2.3 HTML 的基本文本格式
17	2.3 HTML 页的链接
17	2.3.1 Internet 地址
18	2.3.2 网站间的链接
18	2.3.3 网站内部链接
19	2.3.4 网页内部的链接
20	2.4 加入图像
21	2.5 HTML 的窗体
21	2.5.1 窗体中的 HTML 标识符
23	2.5.2 文本框
23	2.5.3 窗体按钮

24	2.5.4 密码框
24	2.5.5 复选框
25	2.5.6 单选按钮
25	2.5.7 建立下拉式列表框
26	2.5.8 建立滚动列表框
27	2.5.9 建立文本区
27	习 题
28	3 VBScript 语言编程基础
28	3.1 VBScript 语言简介
28	3.1.1 VBScript 及其主要功能
28	3.1.2 在 HTML 页面中使用 VBScript
30	3.2 VBScript 数据类型
31	3.3 VBScript 变量
31	3.3.1 声明变量
33	3.3.2 变量命名规则
33	3.3.3 标量变量和数组变量
34	3.4 VBScript 常量
34	3.4.1 声明自定义常量
35	3.4.2 固有常量
35	3.5 VBScript 的表达式与运算符
35	3.5.1 算术运算符
36	3.5.2 关系运算符
36	3.5.3 逻辑运算符
37	3.5.4 连接运算符
37	3.5.5 运算符优先级
38	3.6 VBScript 语句中的基本语句
38	3.6.1 分支语句
41	3.6.2 循环语句
45	3.7 VBScript 语句中的过程
46	3.7.1 Sub 过程
47	3.7.2 Function 过程
48	3.7.3 使用参数
50	3.8 VBScript 中的函数
50	3.8.1 VBScript 函数
51	3.8.2 数据类型转换函数
52	3.8.3 数学运算函数
53	3.8.4 字符串处理函数

54	3.8.5 日期时间处理函数
54	习 题
55	4 VBScript 与 ASP
55	4.1 VBScript 与 ASP 关系
55	4.2 ASP 的特点与环境
55	4.2.1 ASP 的特点
56	4.2.2 ASP 的环境要求
56	4.2.3 ASP 的运行机制
57	4.3 PWS 的安装与设置
57	4.3.1 在 Windows 98 上安装 PWS
60	4.3.2 PWS 的设置
62	4.4 IIS 的安装与设置
62	4.4.1 在 Windows 2000 Server 上安装 IIS
63	4.4.2 IIS 的设置
73	4.5 在 ASP 中使用 VBScript
73	4.5.1 设置主脚本语言
73	4.5.2 在客户端使用 VBScript
73	4.5.3 在服务器端使用 VBScript
77	4.5.4 脚本性能问题
78	习 题
79	5 ASP 的内嵌对象
79	5.1 ASP 的内嵌对象概述
79	5.2 Response 对象
79	5.2.1 Response 对象使用
80	5.2.2 Response 对象的方法
81	5.2.3 Response 对象属性
84	5.3 Request 对象
89	5.4 Application 对象
89	5.4.1 Application 对象特性
90	5.4.2 集合
91	5.4.3 方法
92	5.4.4 事件
93	5.4.5 Application 对象应用举例
94	5.5 Session 对象
94	5.5.1 Session 对象的特性
97	5.5.2 集合

98	5.5.3 属性
99	5.5.4 方法
100	5.5.5 事件
101	5.6 Server 对象
102	5.6.1 属性
102	5.6.2 方法
104	习 题
106	6 ASP 中的常用组件
106	6.1 使用内置文件组件
106	6.1.1 使用文件组件对文件操作
106	6.1.2 对文本文件的写操作
107	6.1.3 在文本文件中读取数据
108	6.1.4 文件的删除、移动和拷贝
109	6.1.5 检测文件是否存在
110	6.1.6 检查文件属性
111	6.1.7 对文件夹和驱动器进行操作
113	6.2 站点导航
116	6.3 Ad Rotator 组件
116	6.3.1 使用 Ad Rotator 组件
118	6.3.2 Ad Rotator 对象的属性
119	6.4 计数器组件
120	6.5 Browser Capabilities 组件
121	6.6 其他组件的使用
121	6.6.1 LyfDNS 组件
122	6.6.2 LyfImage 组件
123	6.6.3 LyfUpload 组件
128	习 题
130	7 ASP 中访问数据库技术
131	7.1 创建 ODBC DSN 文件
132	7.1.1 配置 Microsoft Access 数据库的文件 DSN
133	7.1.2 配置 SQL Server 数据库文件 DSN
134	7.2 使用 Connection 对象
134	7.2.1 用 Access 2000 创建一个数据库
136	7.2.2 使用 Connection 对象打开和关闭数据库
137	7.2.3 用 Connection 对象执行查询
138	7.3 使用 RecordSet 对象

138	7.3.1	RecordSet 对象的一些重要属性和方法
139	7.3.2	RecordSet 对象对数据库的操作
143	7.4	使用 SQL 语句来操作数据库
144	7.5	使用 Command 对象
145	7.6	数据库的连接管理
147	7.7	ASP 中 ADO 的优化
147	7.7.1	Connection
148	7.7.2	Command
148	7.7.3	RecordSet
149	7.8	在 ASP 中使用数据库连接实例——考试系统的设计
153		习 题
155	8	ASP 程序的管理、调试及安全技术
155	8.1	ASP 应用程序的管理
155	8.1.1	管理应用程序
156	8.1.2	ASP 中的管理会话
161	8.2	维护 ASP 应用程序的安全
161	8.2.1	NTFS 权限
161	8.2.2	维护 Global.asa 的安全
161	8.2.3	Web 服务器权限
162	8.2.4	脚本映射文件
162	8.2.5	Cookie 安全性
162	8.2.6	使用身份验证机制保护被限制的 ASP 内容
163	8.2.7	保护元数据库
163	8.2.8	使用 SSL 维护应用程序的安全
164	8.3	调试 ASP 脚本
164	8.3.1	Microsoft 脚本调试工具
165	8.3.2	错误调试技术
166	8.4	制约 ASP 效率的因素
166	8.4.1	提高 HTML 页面性能的几个技巧
167	8.4.2	提高 ASP 程序反应时间的几个技巧
169		习 题
170	9	XML 语言与 ASP 编程
170	9.1	XML 语言概述
170	9.1.1	XML 的形成
170	9.1.2	XML 语言的特点
171	9.1.3	支持 XML 语言的环境

172	9.1.4 XML 应用分类
172	9.1.5 XML 优点
175	9.1.6 XML 发展前景
176	9.2 XML 文档和 DTD 文档
176	9.2.1 XML 文档
176	9.2.2 XML 文档的术语
178	9.2.3 DTD 文档
179	9.3 XML 语法
179	9.3.1 XML 语法规则
180	9.3.2 元素的语法
180	9.3.3 注释的语法
181	9.3.4 CDATA 的语法
181	9.3.5 Namespaces 的语法
181	9.3.6 Entity 的语法
182	9.3.7 DTD 的语法
183	9.3.8 DTD 示例
184	9.4 ASP 与 XML
184	9.4.1 DOM 与 DOMDocument 对象
186	9.4.2 XML 在 ASP 中的应用
195	习 题
196	10 电子商务应用实例
196	10.1 网络广告应用
196	10.1.1 漫游网页的小精灵
199	10.1.2 重定向文件
200	10.2 网上购物系统
200	10.2.1 网上购物系统设计
201	10.2.2 网上购物系统实现程序
209	习 题
210	参考文献

1 动态网页与 ASP 概述

1.1 静态与动态网页概述

1.1.1 HTML 的起源

万维网 (World Wide Web, WWW) 起源于设在瑞士的 CERN 实验室。Tim Berners-Lee 及其开发小组, 花费大量时间, 研究建立了一种以一定格式传输信息的方法, 这就是众所周知的超文本传输协议 (简称为 HTTP)。该协议使用了超文本标记语言 (HTML), 允许在 Web 浏览器及其他兼容的应用程序中显示文本及图像, 文档的某些部分可以成为超链接, 即当其被选择时, 可以显示不同的页面或同一页面的不同部分。

而标记语言就是由特定字符分隔的基本元素, 指定基本元素中所包含的文本或其他条目如何显示。例如, `<EM>This is some emphasized text</EM>`。HTML 是一种广泛基于标准通用标记语言 (Standard Generalized Makeup Language, SGML) 的标记语言。

SGML 则是描述语言的一种方法, 本身并不是创建网页的语言。HTML 语言从简单开始, 其后不断增加一些附加的功能, 最终形成了今天所用的 HTML 4.0 版本。这些新增的特性提供了更灵活的文本字体风格 (例如 `<FONT>` 元素), 以及对输出页面布局的更多控制 (如窗体和帧的使用)。早期的网页所缺乏的是动态的内容。刚开始时, 这并不是问题, 因为 HTML 的开发是为了在计算机、网络及操作系统之间显示和传送信息 (特别是技术和科学信息)。这种标准化的文档仅是一种简单的文本和标记, 其中的图像和其他非文本的内容以单独文件的形式存放, 它们可以在各种网络上自由传送。另外, 由于信息的格式是固定的, 其组成元素的含义也是在 HTML 中定义好的。对于一个“阅读器”或浏览器应用程序, 相对来说, 用任何程序语言在任何平台或操作系统中都是比较容易实现的。只要建立这些信息网页, 文本文件和图像便可通过 Web 服务器应用程序传送给用户, Web 服务器应用程序只需简单地从磁盘中读取它们, 并且把输出转换成能够在网络中传送的正确的 HTTP 协议。在客户或用户端, 浏览器接收传送来的信息流, 并转换成为能够显示的页面。

HTML 和 HTTP 除了具有跨平台特性之外, 其最主要的长处就是其文档可以包含有关内容的相关信息、作者意欲表达的方式等。它可以被应用程序阅读而不一定显示出来, 例如盲人可以使用特殊的程序把它转换成为语音。同样, 其内容可以特殊方式显示出来, 以便那些视力差或有其他缺陷的人们进行访问。因此, 技术界经常用“用户代理” (User Agent) 这一通用术语代替“浏览器” (Browser)。

而通常所说的静态站点中的所谓“静态”, 指的就是站点的主页内容是“固定不变”的。

目前 Internet 上的许多站点, 仍然提供“静态”(Static)的主页内容。浏览器通过 Internet 的 HTTP (Hypertext Transport Protocol) 协议, 向站点服务器 (Web Server) 要求提供主页的内容, 站点服务器收到要求后, 会传送原已设计好的静态的 HTML 文件数据给浏览器。一个“静态”站点, 若要更新主页的内容, 必须手动更新其 HTML 的文件数据。

1.1.2 动态页面的起源

作为商业网站的设计者, 必须要设计出更具实用性和交互性的“动态”网站。通常绝大多数商业网站都具有大量的数据和信息, 而建网的初衷也就在于方便客户查询资料, 方便同客户的交流, 及时获得信息反馈。因此必然会面临如何让使用者在界面中, 通过互联网或内联网 (Intranet) 查询 Web 数据库的资料, 甚至输入、更新和删除 Web 服务器上的资料, 于是出现了动态页面技术。

动态页面中的所谓“动态”, 指的就是站点的主页内容是“可变”的, 即可以随时刷新站点的主页内容。当在浏览器上填好表格 (Form) 的输入数据, 以满足 HTTP 请求时, 可以在站点服务器中执行应用程序, 而不只是一个 HTML 文件。站点服务器收到请求执行的应用程序, 分析表格 (Form) 的输入数据, 将生成的结果以 HTML 的格式传送给浏览器, 产生执行动态, 交互式、高效率的效果。

对于新类型的文档内容, 特别是那些设计成为允许在页面上输入信息的文档 (例如 HTML 中 `<INPUT>` 等的窗体控件), 开发出一种能够读取这种信息并加以利用的应用程序就成为必然。显然, 传递从其他应用程序中得到的内容, 特别是那些特定类型的数据库的内容, 需要一种新的方法。而每次都需要重写一个基于文本的页面, 对于提供实时性很强的信息当然就不是一个理想的方法。

对于 Web 服务器来说, 提供一个接口使其他应用程序能够与之相连成为一种常用的方法。通过这个接口, 定制的可执行程序能够接收来自客户端的信息, 包括通过点击超链接或在浏览器中键入统一资源定位符 (URL) 所提出的页面请求的细节。应用程序对客户端的请求能够生成相应的响应, 而不是从服务器磁盘上读取文本或标记文件。从这些早期方法开始, 逐渐形成了一套完整的系列方法, 动态地创建网页, 以响应用户的请求或信息的变化。

用于这些应用程序的接口仍然在使用, 并被称为公共网关接口 (Common Gateway Interface, CGI), 这是一种可用任何语言 (如 C 语言) 实现的标准。它产生于那些使用 cgi-bin 目录的应用程序 (这里的“bin”代表二进制代码, 而不是文本)。早期的应用程序都是编译后的程序, 通常用 C 或 C++ 编写。然而这就要求懂得 C 语言的编程方法, 并且每当对程序做很小的文字上或标记上的变动, 都必须重新编译、生成可执行文件, 这就限制了 CGI 和动态页面的使用。取而代之的是开发了一种用脚本语言创建网页的方法, 这种语言就是实用摘要和报告语言 (Practical Extraction and Reporting Language), 或简称为 Perl, 它允许信息的创建者以一种非常类似于简化版 C 或 C++ 的语言编写代码。在 Perl 脚本中, 可以“写”文本和标记, 用标准输入 (Stdin) 和标准输出 (Stdout) 函数通过 CGI 与 Web 服务器通信, 输出到浏览器。Perl 在 Web 上仍然是一种流行的语言, 特别是在基于 UNIX 或 Linux 的系统上。

然而这种语言掌握起来不是很容易，特别是对于那些没有 C 或 C++ 语言基础的 Web 开发人员更是如此。现在，一些新的脚本语言出现了，使得创建网页更加容易，下面是详细介绍。

1. 服务器端脚本技术

对于服务器端的脚本需要用某种类型的中间应用程序，或插件程序来连接，必须能够接受用户请求，读取并解释合适的基于服务器的脚本文件，接着创建输出页，并传送给 Web 服务器，在那里作为响应发送给客户端。

在某些情况下，这个任务可划分为两个部分：

- ▶ 一个应用程序或插件程序处理与 Web 服务器的往来通信（一般通过 CGI）；
- ▶ 另一个处理解释和执行脚本。

这就是 ASP 中的情况，至于脚本引擎的使用与其他环境下相同。

Perl 是第一个流行的服务器端脚本语言，目前已经又出现了很多其他的语言。现在在 UNIX 及基于 Linux 的系统上，一种新的称之为 PHP（Personal Home Page）的语言正变得越来越流行。还有一些语言的目标是针对一些特定类型的用户，例如 TCL 就是一种能在科学环境下使复杂的数学运算简单化的语言。

2. 微软的动态页面创建技术

微软随 Windows NT 3.51 推出了它们的 Web 服务器软件——Internet Information Server（IIS）1.0。这是一个提供了很多功能的相当标准的软件，支持 CGI。同时微软也提供了另外一个接口，允许使用 C 和 C++ 等编译语言生成可执行文件，使操作更加高效，这就是 Internet 服务器应用编程接口（Internet Server Application Programming Interface，ISAPI）。它能够比传统 Perl 引擎和其他技术所依赖的 Stdin 和 Stdout 更为广泛的对 Web 服务器的访问。

此后微软和其他第三方开发商推出了许多能通过 ISAPI 连接到 IIS 的应用软件，这也就是 ASP 能够连接到 IIS 以及其他微软服务器端动态技术的原因所在。而在 ASP 以前，用得最广泛的是 Internet 数据库连接器（Internet Database Connector，IDC）。ASP 为 Web 开发者使用微软的平台开辟了一个新的天地，使得采用数据库中的数据创建动态网页更为容易。特别是它引入了模板（Template），即包含了能够插入数据库查询结果的文本和标记的模板文件。

现有的（或将过时的）其他微软服务器端页面创建技术有 db Web 和 OLE ISAPI。事实上，db Web 就是 OLE ISAPI 的实现，而对于多数人来说，迄今只是证实了这一技术在实际应用中的不适应性。OLE ISAPI 是通过特殊版本的 ISAPI 接口与 IIS 通信的一些 COM 对象。Web 服务器软件调用 COM 对象中单个指定函数并以参数为用户的请求提供细节。COM 对象返回的是作为字符串的页面的文本和标记，然后作为响应发送给客户端。

OLE ISAPI 首创了 COM 对象的动态 Web 页面创建，为编程人员提供通过编译的 Active XDLL 创建动态页面的能力，然而它所使用的特定的实现和数据通信技术对于较大的任务和内部网（Intranet）工作缺乏有效性和可扩展性。同时，每当改变页面的文本和标记时，也需要重新编译动态链接库，如图 1.1 所示目前页面创建技术的相互关系。

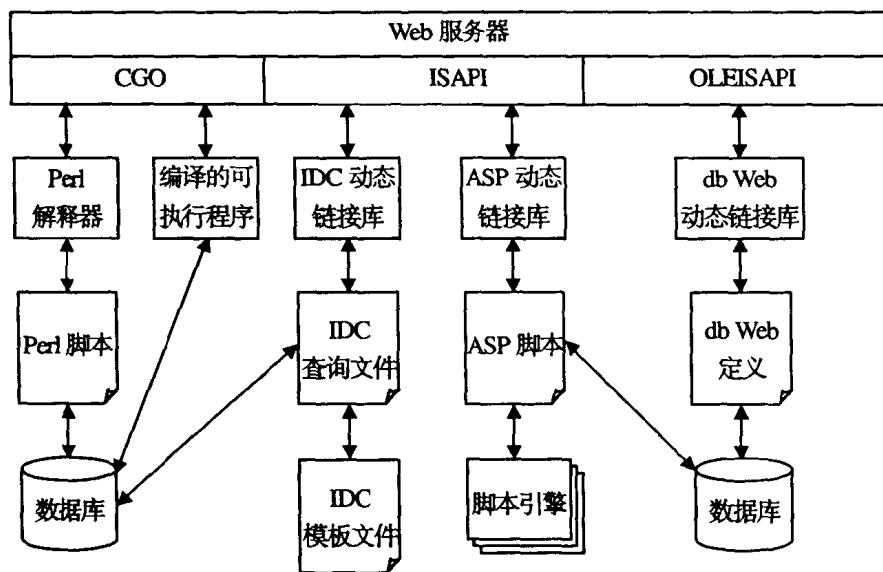


图 1.1 页面创建技术的相互关系

1.2 ASP 概述

1.2.1 什么是 ASP

Microsoft 的 Active Server Pages 称为动态服务器主页，简称 ASP。ASP 是由服务器方动态产生的页面，它可以调用其他的程序去做存取数据库之类的事情，也能根据不同的浏览器类型给出不同的结果，总的来说所有用 CGI 做的事情它都可以做。ASP 就像编写本机应用程序一样高效，因为它是作为一个服务来运行的，并且能够利用多线程结构的优点。

微软对于 ASP 的权威定义是：“Active Server Pages 是一种开放式的非编译应用环境，能够组合 HTML、Scripts 和可重用的 ActiveX 服务器组件来建立动态的、强大的基于 Web 的商业应用。Active Server Pages 支持基于 IIS 上的服务器端脚本并且完全支持 VBScript 和 JScript”。

ASP 是一种“开放技术框架”，意思是无需使用微软的产品就能编写它的代码（虽然使用微软产品仍是最好的方法）。如今，越来越多的人开始选择使用 ASP，因为所有这些相应的微软开发工具都很好，能节省大量的时间。ASP 还具有以最小的代价利用 COM 和 DCOM（组件对象模型和分布式组件模型）对象的优势。

Active Server Pages (ASP) 实际上是将标准的 HTML 文件拓展了一些附加特征，像标准的 HTML 文件一样包含 HTML 语句并且在一个浏览器上解释并显示。但它为 HTML 编写人员提供了在服务器端运行脚本的环境，使 HTML 编写人员可以利用 VBScript 和 JScript 或其他第三方脚本语言来创建 ASP，实现过去需要编写复杂的 CGI 程序才能实现的有动态内容的网页，如计数器等。一个 ASP 文件的后缀为 .asp，其中包含实现动态功能的 VBScript 或 JScript 语句。如果去掉那些 VBScript 或 JScript 语句，它和标准的 HTML 文件没有任何区别。

ASP 提供了一些内建对象。利用这些内建对象，可使脚本更加强大，同时这些对象允许

从浏览器中接收和发送信息。例如，利用 Request 对象，可以接收用户的 HTML 表单中的信息并加以处理。ASP 包含标准的 ActiveX 组件，这些组件可以实现一些复杂的功能，如使用数据库等。不过，你并不会被此局限，仍可以轻松创建属于你自己的附加 ActiveX 组件，意味着可以不加限制地拓展你的 ActiveX 组件。ASP 还可以和诸如 SQL Server 这样的数据库进行挂接，这也是 ASP 一个非常强大的重要功能。在线商务以及在线论坛等各种非常高级的、动态更新的站点都需要数据库的支持，而且需要随数据库内容的更新而自动更新，这样一来，完全可以利用 ASP 建立很多类似的高级站点。通过上面的描述，可以了解到，ASP 就是由服务器端脚本、对象以及组件拓展过的标准网页并为其提供了运行的环境。但它的网页并不是在建立初期存在的，而是当某个浏览器向它提出请求时，它才根据需要产生所需要的标准网页，克服了过去 HTML 编写的网页的静态缺点，从而使网页上可以有許多动态的信息存在。

1.2.2 ASP 的功能

ASP 是一套服务器端的脚本运行环境，当用户从浏览器向 Web 服务器请求 .asp 文件时，ASP 脚本开始运行，然后 Web 服务器调用 ASP，ASP 全面读取请求的文件，执行所有的脚本命令，并将标准的 Web 页传送给浏览器，却并不包含所写的脚本命令。对于 Web 服务器来说，ASP 与 HTML 有着本质的区别，HTML 不经任何处理送回给浏览器，而 ASP 的每一个命令都首先被用来生成 HTML 文件，因此 ASP 允许生成动态内容。另一方面，对于浏览器来说，ASP 和 HTML 几乎是沒有区别的，仅仅是后缀为 .asp 和 .htm 的区别。当我们在客户端提出对 ASP 的申请后，浏览器接收的是 HTML 格式的文件，因此它适用于任何浏览器。

根据以上特性，可以用 ASP 方便地实现诸如表格信息收集、计数器、留言簿、公告板、聊天室甚至电子商务等过去必须由 CGI 才能实现的功能。

ASP 还可以轻松地实现对页面内容的动态控制，为不同的浏览者定制不同的页面内容，实现个性化的网站。

1.2.3 使用范围

利用 ASP 实现各种功能时几乎没有什么限制，只要网站存在并正常运行即可。以下是简单的应用示例：

- ▶ 在网站主页顶部添加一个滚动显示的广告栏；
- ▶ 从 HTML 的表单中接受信息并且存储到数据库中；
- ▶ 根据不同访问者显示不同内容，创建个性化主页；
- ▶ 在主页中添加点击计数器；
- ▶ 根据用户浏览器的版本、类型和能力显示不同档次的内容；
- ▶ 链接多个主页便于网际导航；
- ▶ 跟踪用户网站上的活动信息并且存入日志文件；
- ▶ 使用基于 Windows NT 的 Intranet 建立 MIS 应用。

用 ASP 编写出的系统，完全可以达到原来使用 Client / Server 的网络数据库系统的水平。

最大的好处是,在对程序进行升级时,只需修改服务器上的 ASP 文件,而不需修改客户端程序。

1.3 ASP 的环境与开发

1.3.1 环境的建立

ASP 是微软开发的服务器端脚本环境,内含于 IIS (Internet Information Server) 或 PWS (Personal Web Server) 中,开发 ASP 应用程序系统必须符合以下三个要求之一。

① 如果使用的是 Windows NT Server 系统或 Windows 2000 系统,那么需要安装 Internet Information Server。安装好后, IIS 会在 Windows NT 所在的硬盘上建立其主页发布路径 \inetpub\wwwroot,而 IIS 会在机器启动后自动执行,把编写好的 ASP 文件复制到这个路径下。当然,应该具有在这个路径的读写权限。打开浏览器,在地址输入框内输入 `http://localhost/test1.asp` (假设编写的 ASP 文件的名字是 `test1.asp`),接着,便能看见运行结果了。

② 如果使用的是 Windows 95/98 系统,那么需要安装 Personal Web Server (PWS),它可以在 Windows 98 安装盘的 Tools\PWS 路径下找到。安装好后, PWS 会在 Windows 所在的硬盘上建立其主页发布路径 \inetpub\wwwroot,而 PWS 会在机器启动后自动执行。把编写好的 ASP 文件复制到这个路径下,打开浏览器,在地址输入框内输入 `http://localhost/test1.asp` (假设你编写的 ASP 文件的名字是 `test1.asp`),接着,便能看见运行结果了。

③ 如果拥有租用的虚拟空间,并且有 CGI 和 ASP 的权限,则可直接把文件复制到管理员所给的路径上。

ASP 自身提供了 VBScript 和 JScript 的驱动程序,可以将可执行的脚本直接嵌入 HTML 文件,HTML 开发和脚本开发在同一开发过程中就可完成。

与一般的程序不同,ASP 程序无需编译,它的控制部分,是使用 VBScript、JScript 等脚本语言来设计的。当执行 ASP 程序时,脚本程序将一整套命令发送给脚本解释器(即脚本引擎),由脚本解释器进行翻译并将其转换成服务器所能执行的命令。当然,同其他编程语言一样,ASP 程序的编写也遵循一定的规则,如果想使用所喜爱的脚本语言编写 ASP 程序,那么服务器上必须要有能解释这种脚本语言的脚本解释器。安装 ASP 时,系统提供了两种脚本语言,即 VBScript 和 JScript,而 VBScript 则被作为系统默认的脚本语言。当然也可以根据自己的喜好改变系统默认的脚本语言。

ASP 本身并不是一种脚本语言,它只是提供了一种使镶嵌在 HTML 页面中的脚本程序得以运行的环境。但是,要学好 ASP 又必须掌握它的语法和规则。

ASP 程序其实是以扩展名为 .asp 的纯文本形式存放于 Web 服务器上的,可以用任何文本编辑器打开它。ASP 程序中可以包含纯文本、HTML 标记以及脚本命令,只要将其放在 Web 服务器的虚拟目录下(该目录必须要有可执行权限),就可以通过 WWW 的方式访问它。

要学好 ASP 程序的设计,必须掌握脚本的编写,其脚本是由一系列的脚本命令所组成的。

如同一般的程序,脚本可以将一个值赋给一个变量,可以命令 Web 服务器发送一个值到客户浏览器,还可以将一系列命令定义成一个过程。要编写脚本,必须要熟悉至少一门脚本语言,如 VBScript。脚本语言是一种介于 HTML 和诸如 JaVa、Visual Basic、C++ 等编程语言之间的一种特殊的语言,尽管它更接近后者,但却不具有编程语言所有的复杂、严谨的语法规则。如前所述,ASP 所提供的脚本运行环境可支持多种脚本语言,比如 JScript、REXX、PERL 等,这无疑给 ASP 程序设计者提供了广泛的发挥余地。

ASP 的出现使得广大设计者不必为客户浏览器是否支持而担心,实际上就是在同一个 .asp 文件中使用不同的脚本语言,都不必为此担忧,因为所有的一切都将服务器端进行,客户浏览器得到的只是一个程序执行的结果,而所要做的只需在 .asp 文件中声明使用不同的脚本语言即可。

1.3.2 开发工具

实际上 ASP 的应用程序就是以 .asp 为扩展名的纯文本文件,可以用任何文本编辑器打开并编辑它(如记事本)。ASP 程序中可以包含注释、HTML 标记以及脚本命令,只需将 .asp 程序放在 Web 服务器的虚拟目录下(该目录必须要有可执行权限),就可通过 HTTP 方式访问 ASP 程序。使用那些带有 ASP 增强支持的编辑器更能提高效率,如 Microsoft Visual InterDev。Visual InterDev 是数据库工具,能轻松地建立“开放数据库连接”(ODBC)的数据库。如果从未使用过 HTML,请考虑先使用 Microsoft FrontPage,而且使用 FrontPage 创建文档和格式化文本就像使用文字处理工具一样简单。接下来还可以使用 Insert Script 命令在 FrontPage 创建的 HTML 页中插入简单的 ASP 命令。

1.3.3 ASP 程序编写简述

在 ASP 中,所有脚本命令都由定界符 <% 和 %> 包含,任何包含在这个符号中的内容都被认为是一个脚本,可以在其中插入任何命令,只要这个命令对正在使用的脚本语言有效即可。下面是个例子:

```
<html>
  <head><title>asp script 示例</title></head>
  <body>
    这是个
    <% for i=1 to 5 %>
      非常,
      <% next %>
      非常长的句子。
  </body>
</html>
```

在记事本中输入以上代码,保存为 \inetpub\wwwroot\test.asp,打开浏览器,在地址输入框内输入 http://localhost/test.asp。该段 Active Server Pages 执行后在浏览器上的显示如图 1.2 所示。