

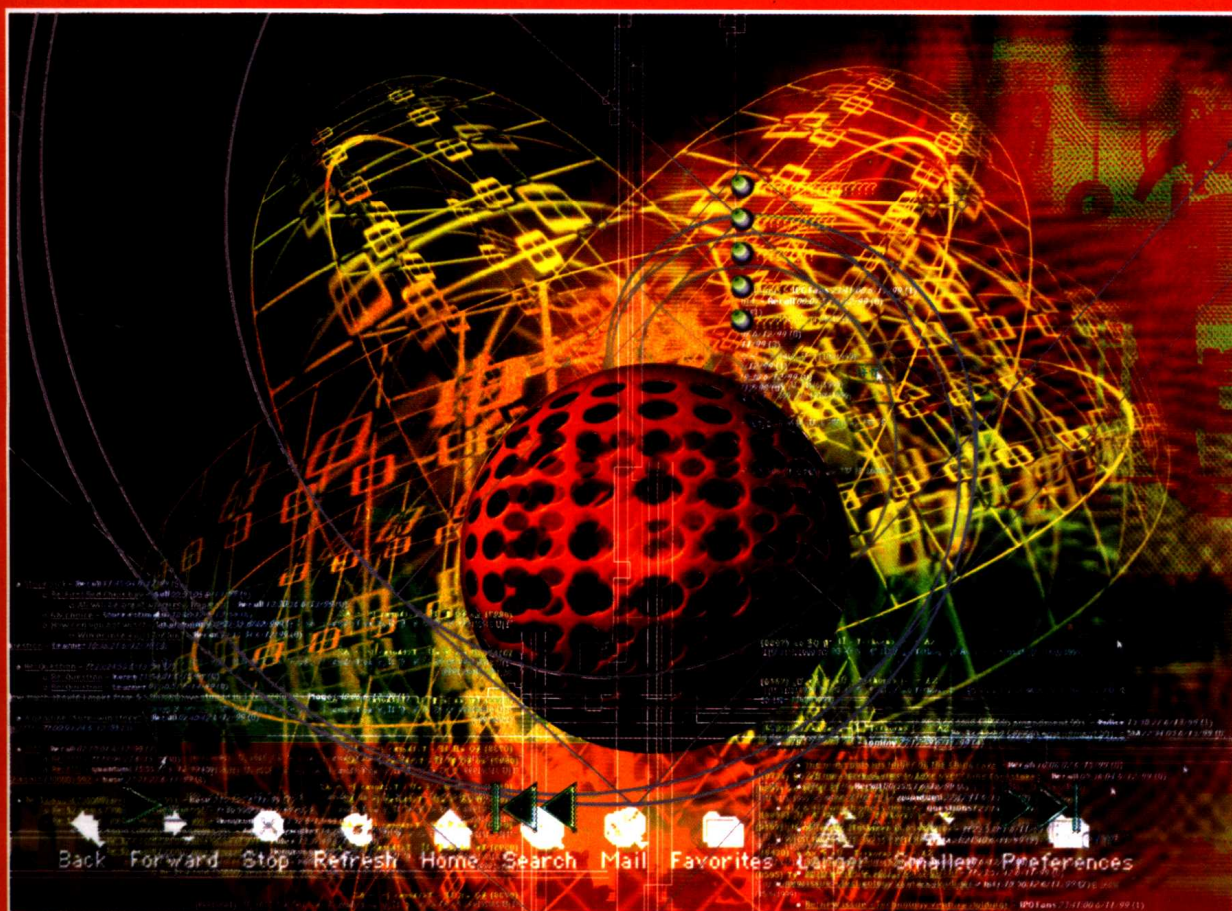
ASP

动态网站建设

高 成 主 编 董长虹 姚德忠 副主编
王 振 崔 桦 编 著

NATIONAL DEFENSE INDUSTRY PRESS

国防工业出版社



动态网站建设丛书

ASP动态网站建设

高 成 主 编

董长虹 姚德忠 副主编

王 振 崔 桦 编 著

国防工业出版社

·北京·

图书在版编目(CIP)数据

ASP 动态网站建设/王振编著. —北京:国防工业出版社,2002.6

(动态网站建设丛书/高成主编)

ISBN 7-118-02804-5

I. A... II. 王... III. 主页制作—程序设计
IV. TP393.092

中国版本图书馆 CIP 数据核字(2002)第 007592 号

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 21 $\frac{3}{4}$ 502 千字

2002 年 6 月第 1 版 2002 年 6 月北京第 1 次印刷

印数:1—3000 册 定价:29.00 元

(本书如有印装错误,我社负责调换)

编 委 会

主 编： 高 成

副主编： 董长虹 姚德忠

编委会成员：

潘时龙	高海滨	朱 奇	马 迪	黄志正
郭兴成	陈 洁	崔益亮	张乐乐	张华弟
杨连池	贾立峰	王 震	余 杰	崔 桦
谢 华	郭明玄	张 龙	张建宁	张雪莲
廖 俊	李耀华	李品一	李元瑾	李 瑾
何 熙	王 剑	杨广军	赵 阳	汪 洋

前 言

有人将 Internet 的出现比喻为“洪水猛兽”，因为几乎就在一夜之间，Internet 的浪潮已经席卷了全球。如今，利用一台小小的计算机，就可以足不出户而知天下之事；可以轻松的和远在万里之外的好友聊天；还可以在闲暇时享受一下网上超市的方便与快捷。

对现代人来说，网上冲浪已经是一种精神食粮，而当在网上浏览着五彩缤纷的网页时，大多数人早已不满足静态页面的浏览，而是更倾心于交互式的动态网页；同时，由于近年来 WindowsNT 的普及，越来越多的企业将微软的 IIS 作为 Web 服务器，于是，ASP 顺理成章的成了 Web 应用程序的开发工具。

ASP (Active Server Pages) 的目的就是为了建立交互式的网页，像常见的访问留言，网上超市，聊天室，搜索引擎等；而 ASP 与 IIS 的结合，将使网络的分布式应用成为现实。Microsoft 的动态服务器网页(Active Server Pages , ASP)技术目前已发展到了 4.0 版。对很多人来说，用 ASP 来创建 Windows 服务器平台上的动态 Web 网页、整个站点和基于 Web 的应用程序已经成为极其自然的方法。

本书共分六个部分，第一部分为开始前的准备工作：讲述了当前 Web 技术的发展状况，同时对基本的 HTML 语言做了简单介绍；第二部分为步入 ASP 的殿堂：主要讲述了 ASP 技术的概况以及 ASP 中的脚本语言 VBScript。第三部分为揭开 ASP 核心技术的面纱：主要讲述了 ASP 的内置对象的应用；第四部分为 ASP 与数据库的结合；第五部分为大量的实例；第六部分为附录。

第一、二部分是基础；第三、四部分为核心技术，也是 ASP 的精华所在；第五部分为实例应用。这样的编排，可以方便不同程度的读者的学习。但在此要声明的是：任何一种知识的学习都不是孤立的，而是需要很多其它知识的辅助，而一本书不可能包括所有必须的知识，因此，要想学好 ASP，还应该去看一些相关的书，例如 HTML、Visual InterDev 等等。

本书的目的在于通过大量的实例来讲解 ASP，让大家在实例中轻松掌握 ASP 的技巧。也许看完这本书后，您会不禁感叹：开发 Web 应用程序竟然如此简单！

在此，对协助本书完成的海龙工作室表示衷心的感谢。同时，由于 Web 技术发展迅速，也由于编者本身水平有限，本书难免有不当之处，敬请批评指正。

编 者

内 容 简 介

本书是针对 ASP 领域的中高级读者而写。根据作者在实际开发中所积累的经验，本书采用通俗易懂的语言，由浅到深的详细介绍了 ASP 中绝大部分的知识点。

本书总体上分三大部分：第一部分从基础的 ASP 知识开始，逐渐深入到 ASP 的内建对象，最后结合数据库将 ASP 技术的应用进行了全面的剖析；第二部分通过大量的实例，使读者感到使用 ASP 开发 Web 程序是如此简单，充分享受一个优秀 ASP 程序员的快乐；第三部分是介绍当前一些比较高级的技术与 ASP 的结合。

目 录

第 1 章 序 言	
1.1 热火朝天的 Web 技术	1
1.2 网页编程语言的发展	2
第 2 章 HTML 简介	
2.1 HTML 概述	4
2.1.1 HTML 的基本语法	4
2.1.2 HTML 中的重要约定	5
2.1.3 HTML 文档的总体 结构	6
2.2 HTML 页面技术	7
2.2.1 文字布局	7
2.2.2 字体设置	8
2.2.3 建立表单	9
2.2.4 建立表格	10
2.2.5 加入图像和超链接	12
2.2.6 制作多窗口页面	13
2.2.7 制作多媒体页面	14
2.3 层叠样式表	15
第 3 章 VBScript	
3.1 初识 VBScript	18
3.1.1 什么是 VBScript	18
3.1.2 VBScript 代码的基本 格式	19
3.2 VBScript 的基本知识	22
3.2.1 VBScript 中的变量	22
3.2.2 VBScript 中的运算符	24
3.2.3 VBScript 的过程与 函数	25
3.2.4 VBScript 中的数据 类型	29
3.3 VBScript 的语法	30
3.3.1 循环语句	30
3.3.2 条件语句	32
3.3.3 Err 对象与 On Error 语句	34
3.4 ASP 脚本的调试	35
3.4.1 脚本错误	36
3.4.2 错误调试技术	36
第 4 章 ASP 概述	
4.1 ASP 技术的工作原理	38
4.2 ASP 技术的特点	38
4.3 ASP 的运行环境	39
4.3.1 Win9x 和 PWS	39
4.3.2 Win2000 和 IIS	41
4.3.3 Web 页的发布	42
4.4 可视化的编程环境——Visual InterDev 6.0	42
4.5 ASP 与脚本语言的关系	43
第 5 章 Web 程序设计的技巧	
5.1 Web 应用程序的目录及文件 架构	44
5.2 ASP 文件的设计惯例	45
5.2.1 声明、定义变量	45
5.2.2 撰写 ASP 程序	47
5.2.3 执行 LLS 与设计 ASP	

程序的小技巧	47	与方法	94
第 6 章 Request 和 Response 对象		8.2.3 Application 对象的 集合	95
6.1 客户端和服务器的交流	50	8.3 Session 对象的应用	97
6.1.1 ASP 的内置对象	50	8.3.1 Session 对象概述	97
6.1.2 客户端和服务器的 交流	51	8.3.2 Session 对象的属性和 方法	98
6.2 Request 对象	53	8.4 Application 和 Session 的事件	102
6.2.1 Request 对象的集合	53		
6.2.2 Request 对象的属性和 方法	64	第 9 章 Active Server 组件	
6.3 Response 对象	64	9.1 Ad Rotator 组件	105
6.3.1 Response 对象的属性	65	9.2 Content Linking 组件	108
6.3.2 Response 对象的方法	68	9.3 Counters 组 件	112
6.4 Cookies 集 合	73	9.4 Browser Capabilities 组件	114
6.4.1 Cookies 背景介绍	73	9.5 Content Rotator 组件	117
6.4.2 Cookies 集合的使用	74	9.6 Page Counter 组件	119
第 7 章 Server 对象		9.7 Permission Checker 组件	120
7.1 服务器的进程	78	9.8 Tools 组 件	121
7.1.1 动态网页中服务器端的 处理	78	9.9 FileSystem 组件	123
7.1.2 服务器端包含的指令	80	第 10 章 第三方组件与脚本 运行期库对象	
7.2 Server 对象的属性	81	10.1 SA - FileUp 组件	128
7.3 Server 对象的方法	82	10.2 JMail 组 件	132
7.4 ASP 的事务处理	85	10.3 ASPHTTP 组件	134
7.5 Server 对象的错误处理	86	10.4 脚本运行期库对象	135
第 8 章 Session 和 Application 对象		第 11 章 ADO 对数据库的访问	
8.1 Web 上的状态管理	91	11.1 SQL 简介	142
8.1.1 状态的准确定义	91	11.1.1 SQL 概念	142
8.1.2 状态的重要性	92	11.1.2 SQL 对表的操作	143
8.1.3 在 Web 上创建状态	92	11.2 ADO 的基本概念	146
8.2 Application 对象的应用	93	11.3 ADO 的对象模型	148
8.2.1 Application 对象概述	93		
8.2.2 Application 对象的属性			

11.4 与数据库的连接	151
11.4.1 连接字符串方法的 使用.....	152
11.4.2 其他连接方法的 使用.....	154
11.4.3 连接语法.....	155
11.4.4 连接缓冲池.....	155
11.4.5 创建数据驱动程序.....	156

第 12 章 ADO 中对象的使用

12.1 Connection 对象	158
12.1.1 Connection 对象的集合、 属性和方法集.....	158
12.1.2 返回一个记录集.....	160
12.1.3 操作命令.....	160
12.2 Command 对象	161
12.2.1 返回记录集.....	161
12.2.2 操作命令.....	162
12.2.3 存储过程.....	162
12.3 RecordSet 对象	167
12.3.1 RecordSet 对象中的 一些概念.....	167
12.3.2 RecordSet 对象的 属性.....	170
12.3.3 RecordSet 对象的 方法.....	171
12.3.4 记录集的分页.....	175
12.3.5 一个 RecordSet 对象的 例子.....	175
12.4 ADO 的使用技巧	179
12.4.1 通用技巧.....	179
12.4.2 针对不同任务所采用的 技巧.....	180

第 13 章 ASP 与客户端数据

13.1 远程数据服务	184
13.1.1 RDS 组件.....	185

13.1.2 数据的绑定与更新.....	186
13.2 在服务器和客户之间传输 数据	189

第 14 章 访问计数器

14.1 应用程序结构	192
14.2 程序设计	193
14.3 图形模式的访问计数器	195
14.4 小结	197

第 15 章 网上投票

15.1 应用程序结构	198
15.2 数据源的建立	199
15.3 程序设计	201
15.4 图形模式的网上投票系统	205
15.5 小结	207

第 16 章 留言板

16.1 应用程序结构	208
16.2 数据表的结构	209
16.3 程序设计	209
16.4 小结	213

第 17 章 WWW 聊天室

17.1 应用程序结构	215
17.2 数据表的结构	216
17.3 程序设计	216
17.4 小结	229

第 18 章 在线订票

18.1 应用程序结构	230
18.2 数据表的结构	231
18.3 程序设计	231

18.4 小结·····	240
--------------	-----

第 19 章 BBS 论坛

19.1 应用程序结构·····	242
19.2 数据表的结构·····	242
19.3 程序设计·····	244
19.4 小结·····	264

第 20 章 网络搜索引擎

20.1 应用程序结构·····	265
20.2 数据表的结构·····	266
20.3 程序设计·····	266
20.4 小结·····	280

第 21 章 XML 技术

21.1 XML 技术概述·····	281
21.1.1 标记和元素·····	281
21.1.2 名称空间·····	282
21.1.3 文档对象模型·····	282
21.2 ADO 中的 XML·····	283
21.2.1 存储为 XML 形式的 ADO 记录集·····	283
21.2.2 ADO 记录集名称空间 ·····	284
21.2.3 打开记录集·····	285
21.3 XML 的发展·····	285

第 22 章 COM 技术

22.1 COM 技术·····	287
------------------	-----

22.1.1 COM 技术的基本 内容·····	287
22.1.2 从 ASP 中调用 COM 组件·····	288
22.1.3 COM 与 ATL 环境·····	289
22.1.4 IUnknown 接口·····	291
22.1.5 自动控制·····	292
22.2 COM+ 技术·····	296
22.2.1 什么是 COM+·····	296
22.2.2 COM+ 的结构·····	298
22.2.3 组件服务·····	299

附录 A Visual InterDev 6.0 介绍

A.1 安装 Visual InterDev·····	301
A.1.1 系统需求·····	301
A.1.2 安装方案·····	302
A.1.3 安装 Visual InterDev ·····	303
A.2 Visual InterDev 的结构·····	304
A.3 创建和维护应用·····	305
A.3.1 创建 Web 项目·····	305
A.3.2 Web 项目类型·····	305
A.3.3 在项目中管理文件和 文件夹·····	306
A.4 在 Visual InterDev 中添加 数据库·····	307

附录 B VBScript 函数参考

第1章 序 言

本章导读：

本章简要介绍当前最火爆的 Web 技术，并对网页编程语言的发展进行回顾。

1.1 热火朝天的 Web 技术

大家都知道，WWW（World Wide Web——万维网）已经诞生很多年了。长期以来，万维网完全是静态的，也就是说，它是基于一种简单的传输协议，并用于向用户发送静态的文本，最初设计 WWW 完全是为了共享数据信息，或者是帮助学术界解决一些研究论文的共享问题，因此它只是通过超级链接将一些系统的文件连接起来。不过在当时看来这已经足够了，因为毕竟比原来 Gopher 浏览器看到的東西要好多了。

在引入了可以进行与用户交互的脚本的 Web 服务器后，人们突然发现原来 Web 页面是如此精彩。然后由于 Java 的出现又使一切发生了变化，使用这种语言，应用程序可以被建立，然后下载到客户浏览器上运行。此后人们所熟悉的客户/服务器的开发模式被引入到 Web 上，这标志着万维网使用方式的根本变革。这种新的 Web 并不是建立在一种小规模应用基础上的技术，而是建立在使用人们熟悉的工具和方法上，将安全可靠的系统移植至 Web。

实际上当今活跃的 Web 技术的框架是一系列“活跃”的交互操作。通常人们用客户/服务器术语来描述 Web，这是一个相互的概念，一般把提出请求的一方称为客户端，而把响应请求的一方称为服务器。这种简单的模型是静态的，它们只对双方的激励作出响应。而在“活跃”的 Web 中，客户机与服务器都是动态的，因此它们的结合具有强大的交互性。

从上面的叙述可以看出，现在看到的崭新的 Web 是一系列先进技术快速发展的结果。在 Web 的空间里，几十万至几百万个站点相互间正进行着激烈的竞争，它们想尽一切办法来吸引用户的注意。因此，简单的、静态的页面对用户是不会有太大的吸引力。动态的、有条理的数据加上友好的、交互性强的界面再加上丰富的内容，这才是用户愿意访问的。当然，数据的自动更新也是非常重要的。在短短的时间内，Web 的面孔已发生了很大的变化。现在能在 Web 页内创建应用程序、访问数据库，使其无论在感觉、操作以及用途上都与 Windows 应用程序非常类似。商家们所需要的不仅仅是一个单纯的宣传媒体，而且是一个交互性很强的应用平台。利用它，商家们可以与目前的客户、潜在的客户、员工以及其他入之间进行沟通，并实现一些在线的服务类商业活动（例如：网上购

物、网上订单、网上就医等)。

为了占领 Internet 这个诱人的市场,各个公司和厂家都拿出了自己的解决方案。IBM 推出了 Visual Age for Java 和 NetData。前者是一个集成开发环境,用于调试 Java 程序、JavaBeans 或应用程序;而后者是专门为非编程人员设计的,它可以按用户的要求生成动态 Web 页面。它们都可以调用 DB2 强大的数据库功能。著名的数据库厂商 Sybase 也在其开发工具 PowerBuilder6.0 的版本中加入了 Web 特性,使用户可以在浏览器中直接访问“Data Window”等著名的控件,并且可以使用它开发 Web 应用程序。同样,曾靠着浏览器在 Internet 上飞黄腾达的网景公司,现在也已推出自己的 Web Server 及完整的 Web 开发工具。所有的商家都想在这里占领一席之地。这些都说明了 Web 能给他们带来巨大的利润。当然作为软件巨头的 Microsoft 公司一直在控制着局面,它领导了 Web 技术的一次次革命。

1.2 网页编程语言的发展

任何一种技术都经历着从不完善到完善的过程,网页编程语言的发展也是这样,从下面的介绍可以看到网页编程语言的发展过程。

1. HTML

WWW 的最初创建是为了分享世界范围内的文本文件,这些文本文件主要作为联机引用材料。我们把这些文件称为静态,是因为请求的用户不能与从网络服务器上传来的内容进行交流。这些文件从它们最初的形式转变成 HTML,而一旦转成了 HTML 形式,这些文件就再也不能够用 HTTP 把它们从网络服务器传到网络浏览器上。这种创建和阅读网络页面和网络内容的静态方法,至今仍用于许多地址上,以演示个人信息、公司轮廓或是无需经常变动的材料。

2. CGI

这种广告版的方法,或是在网络中张贴静态信息的方法,适合一般信息传播的需要。然而,网络用户群体开始要求 Internet (国际互联网) 这一种广泛信息媒体有更为强大的功能。“动态”这个词,描述了根据被送到网络服务器上的信息以创建 HTML 内容的过程。网络服务器可以处理信息,并把处理完的信息转变成 HTML,得到的 HTML 是为特定用户输入和应用而设置的。

相信许多朋友对 CGI 都不会感到陌生,即 Common Gateway Interface——通用网关接口,在 ASP 出现以前,几乎所有的网络上的复杂功能都是由 CGI 实现的,它是用来创建动态 HTML 的最早方法之一。CGI 在 HTTP 服务器和可执行源程序之间建立直接对话。在 CGI 上编程为请求客户机浏览器、网关程序,HTTP 服务器提供了标准对话及处理机制。CGI 程序帮助 HTTP 服务器创建了一个标准界面,使得无需学习特定的 Hypertext Transfer 协议。

CGI 程序通常是用源程序语言,如 Practical Extraction 和 PERL 编写的。早期的 PERL 源程序是在 UNIX 上运行的,因为早期的 HTTP 服务器只存于 UNIX 平台上。然而,随着 HTTP Server For NT 的出现,Internet Information Server 也可以支持 PERL5.0 源程序。

应该说,对于大多数 Internet 应用程序来说,CGI 为开发应用提供了一个很好的平台,但在性能上 CGI 却有着不可回避的缺点:

首先,一个 CGI 程序不可能为多个用户共享。当在网络服务器上调用 PERL 源程序时,网络服务器把 PERL 源程序当作独立的可执行文件,这种可执行程序不仅仅限于一个源程序,它也可以由在 N 台独立的机器上的多个源程序组成。然而由于服务器主机是把 CGI 应用当作独立可执行的,因此每一个 CGI 的应用都会创建一个新的进程,这将会严重的占用资源,尤其是在网址和应用增多而考虑可扩充性的问题时,结果将是灾难性的。

其次,CGI 不能为上下文提供永久性的信息。有时候,用户需要查询一个大型的数据库,结果可能是上万行的数据,而为了显示结果清晰,用户只想每次显示 10 行的数据,但对 CGI 来说,每执行 10 行的数据就要重新运行一次 CGI 程序,显然这效率很低。

再次,CGI 应用不能通过应用程序分享信息,不能与其他的实例化 CGI 程序的记忆空间动态分享信息,因为每一个新的 CGI 进程只在自己的记忆空间中创建。

3. ISAPI

在 CGI 之后,一种简洁、易用的技术出现了,它就是 ISAPI 技术。ISAPI 技术是在改进了 CGI 缺陷的基础上建立起来的。ISAPI 具有与 CGI 同样的功能,但是源程序被执行的方式却不同于传统的 CGI 程序。ISAPI 依靠把源程序装入 HTTP 服务器的记忆空间,以减少创建新进程造成的资源流失。但这也同时带来了另一个问题,由于 ISAPI 应用与 HTTP 服务器分享相同的记忆空间,一旦 ISAPI 应用有问题,则 HTTP 服务器的工作也将产生问题。如果 ISAPI 应用没有清除,不干净的记忆空间就会终止,这将会导致记忆泄漏或者处理丢失。

ISAPI 可以采用下面的两种方式执行:

- (1) 作为 ISAPI 应用。引用 ISAPI 应用的方法与引用 CGI 可执行文件的方法相同。
- (2) 作为 ISAPI 筛选程序。这种方法用来处理要送到 HTTP 服务器上的信息,然后处理 HTTP 服务器响应。筛选程序作为一个分析工具,用来监测被传送的信息以及由 HTTP 服务器产生的信息,还可作为另外一层以保护 HTTP 服务器。使用 ISAPI 筛选功能可实现很多有价值的应用,例如提供公共或自定义的密码、压缩方案,开发核准方法,创建高级登录特性或扫描进来的请求,以减少可能的对安全性的攻击等。

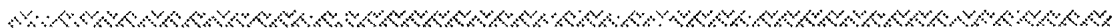
4. ASP

ISAPI 技术的不断发展,逐步演变成今天的 ASP 技术。新兴的 ASP 技术完全摆脱了传统技术的局限,它不但可以进行复杂的数据库操作,而且生成的页面具有很强的交互性,允许用户方便地控制和管理数据。

为了更加灵活、方便地控制数据,服务器端 ASP 引擎支持一套可以方便地访问 Web 服务器上数据库系统的对象模型 ADO (ActiveX Data Objects),ADO 与大家所熟悉的数据对象模型 DAO (Database Access Object) 非常类似,使开发者在不用关心底层数据库指令的情况下即可以完成各种复杂的数据库操作。同时还提供了对 IIS(Internet Information Server)内置的多个对象的支持。除此之外,ASP 技术还可以和微软的 BackOffice 配合使用,形成功能更加强大的应用。例如,在与 Exchange Server 的组合后,可以使用 Exchange Server 所提供的 CDO (Collaboration Data Objects) 对象,直接对邮件系统进行各种控制。

如今,ASP 技术如雨后春笋一样发展起来,它代表着现今 Web 技术的发展趋势。

第2章 HTML 简介



本章导读：

接入 Internet 网，浏览 WWW 中丰富多彩的主页，对于一般用户已不陌生。现在，如何编写自己的主页，在网络上展示自己的风格，已逐渐成为热门话题，而超文本标记语言（HTML）是在 Internet 网络中建立主页(Home Page)的基础。

在本章中，将简单介绍 HTML 的使用，包括文字布局与字体、表单和表格的建立、如何加入图像以及如何制作多窗口和多媒体的页面。由于本书重点是 ASP 的讲解，因此对 HTML 这种基础知识只作简单介绍，起一个引导作用，在实际应用中用户可查阅相关书籍。

2.1 HTML 概述

2.1.1 HTML 的基本语法

超文本标记语言 HTML(HyperText Markup Language)是制作 Web 信息的语言，是设计制作页面的基础，因此 Web 编程者必须熟练掌握 HTML。

HTML 文档通过标记和属性对超文本的语义进行描述。因此，HTML 本质上虽然不是一种编程语言，但有一套严格而简明易懂的语法规则，在编程时应该注意这些规则的使用。

HTML 是一种标记语言，用来创建与系统平台无关的文档。也就是说，按照 HTML 语法规则建立的文档可以被运行于不同硬件和操作系统平台上的应用程序所理解。

HTML 文档是纯文本文件，“超文本”的意义在于，HTML 允许在纯文本中嵌入一些标记，以指导应用程序对文本进行操作。

1. HTML 元素

一种文档应用必须定义表示文档结构和所需特性的各种元素，HTML 也不例外，HTML 中，典型的元素由三部分组成：起始标记、内容和结束标记。

元素的起始标记写作<ElementName>，其中 Element-Name 为此元素的名称。元素结束记为在元素名称前面加一斜杠，形如</ElementName>。例如：

```
<head>HTML 的基本语法</head>
```

有一些 HTML 元素可以没有内容。例如，行结束元素 BR 就没有内容，它的作用是结束文本的一行。

此外，HTML 的元素名是不区分大小写的。也就是说，<head>与<HEAD>效果是一

样的。

2. HTML 元素的属性

HTML 元素具有相关属性，可对这些属性赋予适当的值。属性名和属性值出现在元素起始标记的结束符之前。一个元素可以具有多个属性，它们都必须出现在该元素起始标志前，彼此以空格分割，各属性的顺序可以是任意的。例如，下面的代码将 H2 元素的 align 属性设为 center:

```
<H2 align="center"></H2>
```

属性值使用英文双引号或单引号括起来。当使用双引号时，属性值内可使用单引号，反之亦然。某些情况下，HTML 中也可以不使用引号指明属性值。

应该注意，属性名也是不区分大小写的，同时属性值中只能包含字母、数字、连字符或者句点。

3. HTML 文件的注释

HTML 文件注释的基本格式如下:

```
<!--此处为 HTML 文件的注释-->
```

同任何其他语言的注释一样，HTML 注释只用来表示备注信息，应用程序将不处理注释中的任何内容，仅仅是用来增强程序的可读性。正是由于这个原因，在创建 HTML 文档时应该注重注释的使用，这是保证程序清晰、易懂的关键所在。

2.1.2 HTML 中的重要约定

1. HTML 文件的扩展名

HTML 文件的扩展名应该是.html 或者.htm，但对于不同的操作系统来说，又有着各自不同的约定。

早期的 Windows 3.x 是基于 DOS 的操作系统，其文件名长度受到限制，因此只能使用.htm 的扩展名；而在 Windows 95 以后的版本中，则允许.html 的扩展名。

UNIX 操作系统中规定，HTML 文件的扩展名必须是.html。如果使用了三个字符的.htm，那么当网上的 UNIX 系统用户浏览 HTML 文档时，所看到的只是 HTML 源文件，而不是显示结果。

2. HTML 文档的显示

HTML 文档是加有标记的纯文本文件，所以其源文件不包含任何格式，而用元素来表示其字符和段落格式。也就是说，HTML 文档可用任意纯文本编译器创建或修改其文档格式，并由浏览器解释执行。因此，同一 HTML 文档，在不同操作系统、不同浏览器下的显示是不同的。

例如，HTML 中用元素 B 表示加粗字符，用 I 表示斜体字符，因此下面的代码:

```
<B>这里显示加粗字符</B>
```

```
<I>这里显示斜体字符</I>
```

在浏览器中的显示结果是:

这里显示加粗字符 *这里显示斜体字符*

由于浏览器是根据 HTML 标记和属性来显示文档的格式的，所以它对源文档中的回车换行符并不理解。因此在 HTML 中要想强行换行，必须使用换行元素 BR。

HTML 的一个重要特性是支持超级链接。在浏览器中，HTML 源文件中的超级链接通常以不同颜色的字符显示，一般加有下划线。

3. HTML 文档字符

为了使 WWW 能真正全球化，HTML 必须能够表达尽量多的语言，HTML 中使用复杂的字符表示方法来保证这一点。

HTML 的文档字符集是 ISO 10646 规定的通用字符集 UCS。为了使 HTML 用户程序能够正确接受或输出文档，必须使用字符编码。字符编码是文档字符集的子集。

由于有了字符编码，HTML 文档作者就可以用文档字符的子集进行工作，例如可用中文写作 HTML 文档。为了实现这一点，浏览器必须正确的将 Unicode 字符与所用的字符编码进行对应。实现这一点的基础是：Web 服务器根据 HTTP 协议发送 HTML 文档时，会使用 HTTP 头标 Content-Type 字段中的 charset 参数指明所用的字符编码方式。例如，下面的 HTTP 头标指定字符编码方式为“gb_2312-80”：

```
Content-Type:text/html;charset= gb_2312-80
```

但是并不是所有 Web 服务器都会发送字符编码信息。因此 HTML 规定了一种方法，使 HTML 文档可以告知浏览器所用的字符编码。这是通过 HTML 中的 META 元素来实现的，如下所示：

```
<META http-equiv="Content-Type" Content="text/html;charset= gb_2312-80">
```

2.1.3 HTML 文档的总体结构

HTML 文件的一般结构如下所示：

```
<html>
<head>
<title>...</title>
</head>
<body>
    <font size=5>Hello World!</font>
</body>
</html>
```

可以看到，HTML 文件中的标记一般都是成对出现的，例如<html>和</html>。各种标记代表的含义如下：

```
<html>...</html>
```

指明文件包含 HTML 编码信息。文件扩展名.html 也指明该文件是一个 HTML 文档。

```
<head>...</head>
```

HTML 编码文档中包含标题 title 的第一部分。标题是作为你的浏览器窗口的一部分来显示的。

```
<title>...</title>
```

这一对标志含有文档标题并且作为一种全局上下文来识别其内容。标题通常显示在浏览器窗口的某个位置（通常在顶端），而不是在文本区。标题的选择应当是描述性的、独特的和相对简洁的。

`<body>...</body>`

HTML 文档的第二部分，也是最大的部分即正文 `body`。它含有文档的内容（显示在浏览器窗口文本区的部分）。

2.2 HTML 页面技术

2.2.1 文字布局

1. 行控制

浏览器是忽略 HTML 文档中的自然分行与段落的，分行和段落设置必须通过 HTML 标记来实现。

(1) 分段元素

HTML 分段元素 `P` 用来定义文档中的一个段落。段落的缩进、第一个词前面的空格以及其他特性主要由浏览器决定，也可能受其他元素或版面风格的影响。通常浏览器会给一个段落前后各加一个或半个空行，在有的浏览器中可能第一行是缩进的。

标记 `<P>` 可看做是段落的开始，而标记 `</P>` 则可视作段落的结束。单独一个 `<P>` 会添加一个空行。结束标记符 `</P>` 可以省略不写，但在有 `ALIGN` 属性时则不能省略。

(2) 换行元素

换行元素 `BR` 是在 HTML 文档中增加一个换行。与 `P` 元素不同的是，它只是进行了换行而不改变段落设置和其他字符。

2. 文本对齐方式

文本对齐包含水平方向和垂直方向两个方面。在此主要指水平方向的对齐。

(1) `ALIGN` 属性

`ALIGN` 属性用来表示文本在水平方向的对齐方式。如下所示：

```
<P ALIGN=LEFT/CENTER/RIGHT></P>
```

等号右边的三个值分别表示文本靠左边、中间和右边对齐。

(2) `CENTER` 元素

`CENTER` 元素的作用是使文本居中。处于 `CENTER` 元素起始标志符中间的任何文本，包括标题、正文或其他插入对象都将在浏览器窗口中居中显示。

3. 列表元素

文档中的列表是很重要的，它将使文档一目了然。在 HTML 中的列表分为无序列表、有序列表以及自定义列表。

(1) 无序列表：UL 元素和 LI 元素

`UL` 元素可用于创建一个以原点等项目开始的无序列表。列表中的每一项用 `LI` 元素加以识别。如下所示：

```
<UL>
  <LI>Red
  <LI>Green
  <LI>Blue
```