



“九五”国家重点电子出版物规划项目·希望计算机知识普及系列

Web应用

编程

丛书 6



北京希望电子出版社 总策划

李明武 主编

郭胜 夏帮贵 吉雷 编写

网站开发新动力

JavaScript & HTML

精通网页编程

本光盘内容包括：



1. 本版书中的实例程序源文件
2. 本版电子书



北京希望电子出版社

Beijing Hope Electronic Press
www.bhep.com.cn



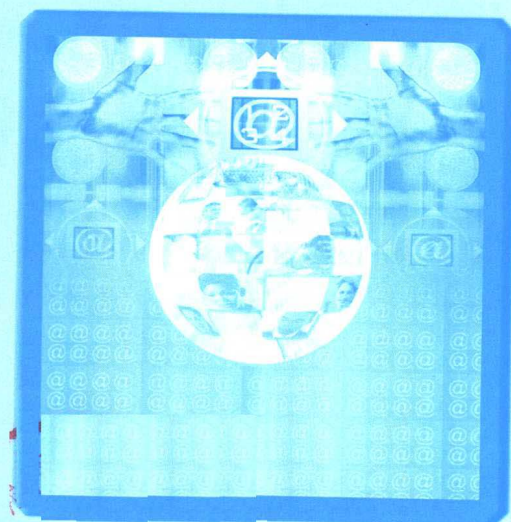
“九五”国家重点电子出版物规划项目·希望计算机知识普及系列

Web应用

编程

丛书 6

TP393.092
252



北京希望电子出版社

李明武

郭胜 夏帮贵 吉雷

总策划

主编

编写

JavaScript & HTML 精通网页编程



本光盘内容包括:

1. 本版书中的实例程序源文件
2. 本版电子书



北京希望电子出版社

Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本盘书是 Web 应用编程系列丛书之一。JavaScript 是页面设计的脚本语言，它的语言结构类似于 C 语言的结构，在网页设计过程中无需具有 C 语言方面的编程经验，就可以轻松地实现交互性动态 Web 页面，因此，JavaScript 是一种简便、实用的 Web 编程语言。

全书由 JavaScript 基础知识和实例两大部分构成。各章知识点具有独立性又紧密相关。全书内容包括：JavaScript 与 HTML 的基础知识与相关内容的介绍，对 Window（窗口）对象、document 对象、文本对象、按钮对象、选择和隐藏对象、location 对象、history 对象、layer 对象、日期对象、数学对象以及数组对象的描述，样式单实例以及实用小程序，JavaScript 语言的扩展以及用 JavaScript 语言构建网上购物系统，实例程序源代码，时钟日历页面实例，最后介绍 JavaScript 服务器编程方法，并对网络的安全性进行了讨论。

全书以实例为主，图文并茂，引导读者使用 JavaScript 进行网页设计与开发，具有极强的实用性、可读性。读者只要稍加修改就可以将其用在自己的页面中，轻松地进行动态页面的实现。

本书适用于网页开发人员、设计人员以及对网页开发有兴趣的初学者，也可作为高等院校相关专业师生教学、自学参考用书和社会相关领域培训班的教材。

本光盘内容包括本版书中的实例程序源文件及本版电子书。

系 列 盘 书：Web 应用编程丛书（6）
书 名：JavaScript & HTML 精通网页编程
总 策 划：北京希望电子出版社
文 本 着 作 者：李明武 主编 郭 胜 夏帮贵 吉 雷 编写
责 任 编 辑：郭淑珍
C D 制 作 者：希望多媒体开发中心
C D 测 试 者：希望多媒体测试部
出版、发行者：北京希望电子出版社
地 址：北京中关村大街 26 号，100080
网址：www.bhp.com.cn
E-mail: lwm@hope.com.cn
电话：010-62562329,62541992,62637101,62637102,62633308,62633309
（发行和技术支持）
010-62613322-215（门市） 010-62521798, 62547735（编辑部）
经 销：各地新华书店、软件连锁店
排 版：希望图书输出中心
C D 生 产 者：北京中新联光盘有限责任公司
文 本 印 刷 者：北京广益印刷厂
开 本 / 规 格：787 毫米×1092 毫米 16 开本 21.50 印张 499 千字
版 次 / 印 次：2001 年 3 月第 1 版 2001 年 3 月第 1 次印刷
本 版 号：ISBN 7-900056-84-X/TP·83
定 价：35.00 元（1CD，含配套书）
说明：凡我社光盘配套图书若有自然破损、缺页、倒页、脱页，本社负责调换。

前 言

JavaScript 是 Netscape 公司开发的一种能够直接嵌入到 HTML 文档中的脚本语言。利用 JavaScript 语句可以很容易地实现 web 页面的交互功能。利用 JavaScript, 页面设计人员可以方便地设计和处理动态数据的查询, 控制页面的显示, 设置各种动态效果, 轻松实现动态页面的设计。

本书用实例详细地介绍了 JavaScript 语言的各个方面, 书中实例完全调试通过, 用户可以按照书中的顺序, 照着书中的例子学习, 这样, 您很快就会掌握 JavaScript 语言, 并成为动态页面设计的高手。我们希望通过该书的学习, 读者能够熟悉动态页面设计的过程, 成为 Web 页面设计的能手。

本书由李明武主编, 郭胜、夏帮贵、吉雷等编著, 全书由李香敏策划、审校。另外, 唐静、张凯、肖莉、李琦、陈辉、冯明龙、曾雨苓等人参与了部分章节写作、插图和录入工作, 蒋蕾、宋玉霞、缪军、杨文镛、杨治国等人参与本书的校对工作。由于编者水平有限, 错误之处在所难免, 敬请广大读者和同行批评指正。

读者在使用本书的过程中如有其它问题、意见或建议可以访问导向科技资讯机构网站 [Http://www.dx-kj.com](http://www.dx-kj.com) 或 dxkj@dx-kj.com, dxkj@21cn.com 以及打电话: (028) 3355939 与我们联系。

导向科技资讯机构

2000 年 11 月

JS522/25



第 1 章 JavaScript 基础.....	1	第 9 章 location 对象.....	111
1.1 关于 JavaScript.....	2	9.1 hash 属性.....	112
1.2 了解 JavaScript.....	3	9.2 Href 属性.....	114
1.3 World Wide Web.....	7	9.3 pathname 属性.....	115
1.4 Web 应用程序结构.....	8	9.4 Protocol 属性.....	115
1.5 JavaScript 与 VBScript.....	13	第 10 章 history 对象.....	118
第 2 章 JavaScript 与 HTML.....	20	第 11 章 layer 对象.....	123
2.1 HTML 基 础.....	21	11.1 layer 属性.....	124
2.2 在 HTML 文档中嵌入 JavaScript.....	24	11.2 layer 对象的方法.....	129
2.3 编写 JavaScript 脚本.....	27	11.3 JavaScript 操作层.....	131
第 3 章 JavaScript 基本语法.....	33	第 12 章 字符串对象.....	133
3.1 JavaScript 基本数据结构.....	34	12.1 转义字符.....	134
3.2 JavaScript 运算符和表达式.....	36	12.2 字符串对象的属性.....	135
3.3 JavaScript 控制结构和循环.....	39	12.3 字符串对象的方法.....	136
第 4 章 Window (窗口) 对象.....	55	第 13 章 日期对象.....	150
4.1 Window (窗口) 对象的属性.....	56	13.1 时间对象的属性.....	151
4.2 Window (窗口) 对象的方法.....	58	13.2 时间对象的设置.....	151
4.3 创建和关闭窗口.....	71	第 14 章 数学对象.....	168
第 5 章 document 对象.....	75	14.1 math 对象的属性.....	169
5.1 document 对象的属性.....	76	14.2 math 对象的方法.....	171
5.2 document 对象的方法.....	83	第 15 章 数组对象.....	183
第 6 章 文本对象.....	89	15.1 数组对象的创建.....	184
6.1 文本对象属性.....	90	15.2 数组对象的扩充.....	185
6.2 文本对象的方法.....	92	15.3 对象类数组.....	187
6.3 文本对象的事件.....	95	第 16 章 样式单实例.....	190
6.4 文本区域对象.....	96	16.1 样式单的实用.....	191
第 7 章 按钮对象.....	98	16.2 样式单的定义.....	193
7.1 button、submit、reset 对象.....	99	16.3 样式单的使用.....	194
7.2 复选框对象.....	101	第 17 章 实用小程序.....	198
7.3 Radio 对象.....	103	17.1 导言.....	199
第 8 章 选择和隐藏对象.....	105	17.2 状态栏滚动信息.....	199
8.1 select 对象.....	106	17.3 计算用户来访次数.....	201
8.2 隐含对象.....	108	17.4 散布页面的星星.....	203
		17.5 永在顶端的图片.....	207



第 18 章 JavaScript 语言的扩展	211	第 21 章 时钟日历	289
18.1 ActiveX 通信	212	21.1 示例特性	290
18.2 调用插入件	216	21.2 源代码	290
第 19 章 网上购物系统	218	21.3 功能概述	295
19.1 示例特性	219	21.4 程序详解	295
19.2 源代码	219	第 22 章 JavaScript 服务器端编程	312
19.3 功能概述	228	22.1 预备知识	313
19.4 程序详解	233	22.2 实例学习	320
第 20 章 2000 珍藏版	256	22.3 功能概述	324
20.1 Cookie 入门	257	22.4 脚本详解	325
20.2 实例特性	257	第 23 章 网络安全性	335
20.3 程序源代码	258	23.1 安全性破坏的种类	336
20.4 功能概述	265	23.2 安全服务	336
20.5 程序详解	267		

第1章

JavaScript 基础

主要内容

-  关于 JavaScript
-  了解 JavaScript
-  World Wide Web
-  Web 应用程序
-  JavaScript 与 VBScript

本章导读



JavaScript 作为一种新的 Web 技术，自出现以来就获得了各方面的广泛支持。本章介绍了 JavaScript 的基础知识以及与其它脚本语言的区别。

1.1 关于 JavaScript

JavaScript 是一种新的 Web 技术。JavaScript 最初的名字是 LiveScript, 它是 Netscape 开发出的一种脚本语言, 其目的是为了扩展基本的 HTML 的功能, 用于代替复杂的 CGI 程序来处理 Web 页表单信息, 为 Web 页增加动态效果。当 Java 出现以后, Netscape 和 Sun 一起开发了一种新的脚本语言, 它的语法和 Java 非常的类似, 所以它最后被命名为 JavaScript。

在 JavaScript 出现以前, 在 Web 页中需要进行的所有处理都必须传回服务器, 由服务器进行集中处理。服务器处理完毕后, 在将处理结果通过网络传回客户端的浏览器中供用户查看。即使是最简单的验证用户在文本框中输入数据的有效性, 比如通过判断输入字符串是否包含 “@” 符号来判断用户输入的 E_mail 地址是否有效, 都必须由服务器来完成。在这种方式下, 当 Web 访问量增加时, 网络和服务器的负担都会增加。这一时期的客户/服务器结构并不是真正意义上的客户/服务器结构。人们期待一种新的技术来实现真正的客户/服务器结构, 即在客户端也可以进行处理, 从而减轻服务器的负担, 加快网络的传输速度。JavaScript 正是在这种背景之下产生的。

JavaScript 至 1995 年诞生以来, 已经取得了广泛的支持, 他们包括 Apple、Borland、Sybase、Informix、Oracle、Digital、HP 和 IBM 等。不仅仅是在浏览器中得到越来越多的支持, 在其它的各种应用程序中也得到应用。新的 Windows 操作系统中也可以使用脚本来制订需要完成的任务。

有经验的 Web 页作者都知道 Java 的小应用程序也可以实现客户端的逻辑处理能力。但作为一种强类型的程序设计语言, Java 并不是制作 Web 页的最佳选择。因为使用 Java 需要 Web 页作者有较高的编程能力, 而这对众多 Web 页作者来说是件难事。Web 作者愿意使用更简单的方法来实现表单的处理。

Microsoft 也意识到了 Web 脚本的重要性。作为软件界的领头羊, Microsoft 自然不甘在 Web 脚本的竞争中落后。由于得不到 Netscape 在技术上的许可, Microsoft 开发了自己的脚本语言——JScript, 并在 Microsoft 自己的浏览器 Microsoft Internet Explorer 3.0 以及更高版本中对其提供支持。由于 Microsoft 在浏览器市场中的优势, JScript 很快得到广泛支持和应用。JScript 1.0 只是很粗糙地和 JavaScript 1.1 兼容, Netscape 在其浏览器 Navigator 3.0 及其以后的版本中也对 JScript 提供了支持。随着 JavaScript 版本的增多和浏览器平台的不同, 让众多的 Web 页作者感到难以取舍, 也增加了额外的工作量。

鉴于脚本语言开发商之间的竞争给 Web 页作者带来的麻烦, Microsoft、Netscape 和其它脚本语言商决定成立一个国际组织, 并将其命名为 ECMA, 该组织专门从事脚本语言标准的制订。ECMA 制订的脚本语言标准被称为 ECMAScript, 所有开发商的脚本语言都支持这一标准。尽管有 ECMA 标准的存在, Netscape 和 Microsoft 都有其各自的脚本语言——JavaScript 和 JScript, 这两种语言都对 ECMA 标准进行了扩展。

Microsoft 除了 JScript 之外, 还有 VBScript 也是一种脚本语言。VBScript 实际上是 Visual

Basic 程序设计语言的一个子集,使得众多的 VB 程序设计员很容易编写自己的 Web 应用程序。Netscape 并没有对 VBScript 提供支持,所以使用 VBScript 的还仅是 Microsoft Internet Explorer 用户。

即使有 VBScript 的竞争,JavaScript 还是成为了标准的 Web 脚本语言。在人们的印象中,JavaScript 只是用来编写客户端的 Web 应用程序。Netscape 为用户提供了服务器端的脚本语言 Netscape Server_Side JavaScript (SSJS),可以在服务器端编写需要的 Web 应用程序,不过使用 SSJS 需要 Netscape Server 3 的支持。Microsoft 也有自己的服务器端脚本编程语言 Active Server Pages (ASP)——需要 JScript 引擎的支持。

1.2 了解 JavaScript

学习 JavaScript 这样的新工具可以说是一种挑战,因为很难理解它是如何使用以及能用于哪些方面。学习 JavaScript 并不是一件非常困难的事,我们先从下面的 10 个方面了解一下 JavaScript 的特点。

1.2.1 JavaScript 是被嵌入到 HTML 中的

JavaScript 的最大特点便是和 HTML 的结合。在客户端的应用中,很难将 JavaScript 程序和 HTML 文档分开。JavaScript 代码总是和 HTML 一起使用的,JavaScript 的各种对象都有各自的 HTML 标记。当 HTML 文档在浏览器中被打开时,JavaScript 代码才被执行。JavaScript 代码使用 HTML 标记<script>……</script>嵌入到 HTML 文档中。JavaScript 扩展了标准的 HTML,为 HTML 标记增加了事件,通过事件驱动来执行 JavaScript 代码。在服务器端,JavaScript 代码可以作为单独的文件存在,但也必须通过在 HTML 文档中调用才能起作用。

下面的程序清单 1.1 中的例子说明了 JavaScript 代码是如何嵌入到 HTML 文档中的。

程序清单 1.1 在 HTML 文档中嵌入 JavaScript 代码

```
<html>
<head>
<title>在 HTML 文档中嵌入 JavaScript 代码</title>
<script language="javascript">
  <!--
    window.defaultStatus="使用 HTML 标记嵌入 JavaScript 代码"
    function getnews()
    {
      document.form1.textbox.value = "在 HTML 文档中使用 JavaScript 代码"
    }
  //-->
```

```
</script>
</head>
<body>
<center>
<h1>使用 JavaScript</h1>
<hr>
<form name="form1">
<input type="text" name="textbox" size=40 value="单击按钮查看信息">
<br><br>
<input type="button" value = "查看信息" onclick = "getnews()">
</form>
</center>
</body>
</html>
```

本例仅说明如何在 HTML 文档中嵌入 JavaScript 代码，在后面的章节中将详细介绍 JavaScript 的使用。图 1-1 显示了本例在浏览器中打开的实际效果。



图 1-1 在 HTML 文档中嵌入 JavaScript 代码

1.2.2 JavaScript 需要环境的支持

JavaScript 是作为一种语言而不是工具出现的。工具是可以不依赖环境而单独使用，但 JavaScript 的运行需要环境的支持。运行 JavaScript 的浏览器，如 Microsoft Internet Explorer 和 Netscape Navigator，或者服务器端引擎是一个解释引擎。JavaScript 只有在这些解释引擎的支持下才能发挥强大作用。如果使用的浏览器不支持 JavaScript，那么嵌套在 HTML 文档中的 JavaScript 代码将会被忽略。最严重的结果是 JavaScript 代码被不支持的浏览器当作文本原样显示在浏览器中。

由于环境因素的存在，在编写 JavaScript 应用程序时必须考虑我们在何时何地使用它以及浏览器是否支持。如果浏览器不支持我们又该如何实现？可以使用非 JavaScript 方法

来解决吗？所有的问题都需要我们在编写 JavaScript 应用程序之前进行考虑。

一个有力的说明是，帧在刚发布的时候，虽然是一种新的、革命性的技术，但这一技术却只有 Netscape Navigator 2.0 支持。这让 Web 作者难以决定什么时候使用帧，而当浏览器不支持时也不知道该如何进行处理。后来 Microsoft Internet Explorer 3.0 开始支持帧，在 HTML 4 中帧已成为了标准。

现在 Web 作者不用再对 JavaScript 的环境依赖性做过多的考虑，JavaScript 已获得了众多浏览器的支持。Netscape Navigator 2.x、Microsoft Internet Explorer 3.x、Opera 3.x 和 HotJava 3.x 等主要浏览器以及它们的更高版本都能支持 JavaScript。一个数字统计表明，大约有 95% 的浏览器现在都对 JavaScript 提供支持，这对 JavaScript 使用者来说是一个福音。

1.2.3 JavaScript 是解释执行的

和大多数脚本语言一样，JavaScript 在浏览器中是解释执行的。应用程序的执行通常有解释和编译两种方式。编译是将程序源代码翻译成可执行的二进制代码文件，如 .EXE 文件，而解释则是翻译一句就执行一句。JavaScript 代码并不被编译为二进制代码文件，而是作为 HTML 文件的一部分。由浏览器解释执行 JavaScript 代码的缺点是代码的执行需要花更长的时间；优点则是 Web 页的维护和更新更加方便。管理员可以直接打开 HTML 文件来编辑修改 JavaScript 代码，而用户则可通过浏览器立即看到新的结果。

值得注意的是，如果是在 Netscape 的服务器端使用 JavaScript，需要用户将所有的 JavaScript 代码和 HTML 文件编译成字节代码，并存储在 .web 文件中。

1.2.4 JavaScript 是一种弱类型语言

JavaScript 与 Java 和 C++ 等强类型语言不同，强类型语言要求用户在程序中使用一个变量之前必须先进行声明。JavaScript 则显得非常灵活，在使用一个变量时，可以先进行声明，也可以不那样做。下面的例子说明了在 JavaScript 中使用变量的灵活性。

```
<html>
<head>
<title>JavaScript 是弱类型的</title>
<Script Language="JavaScript">
  <!--
  var myVar //声明 一个变量
  myVar="JavaScript 是弱类型" //为变量赋值
  alert(myVar) //使用消息框显示变量值
  myVar=3.1415926 //为变量赋不同类型的值
  alert(myVar)
  a="使用未声明变量" //使用未声明的变量
  alert(a)
  //-->
```

```
</SCRIPT>
</head>
</HTML>
```

1.2.5 JavaScript 以对象为基础

JavaScript 中使用了面向对象程序设计 (OOP) 的方法。在后面的章节中, 我们将看到 JavaScript 的各个自定义对象, 如 Window、Document 等对象。

对象实际上是封装了的数据 (属性) 和行为 (方法) 的集合。如果你使用过 Java、Delphi 或 Visual Basic, 则学习 JavaScript 将是一件轻松的事。JavaScript 的对象都是实例化了的, 只可以使用而不能创建继承于这些对象的新的子类。

1.2.6 JavaScript 通过事件驱动执行

我们在后面将学习到 JavaScript 代码是如何使用事件来驱动执行的。HTML 文档中的许多 JavaScript 代码都是通过事件驱动的。JavaScript 本身支持事件。HTML 对象, 如按钮和文本框, 增加了对事件的支持。如果你学习过 Java 或 Visual Basic 等面向对象程序设计, 则你一定知道事件驱动。如果你仅学习过像 C 语言等面向过程的程序设计, 也不用担心, 在后面我们详细讲解 JavaScript 的事件驱动。

1.2.7 JavaScript 不是 Java

当你在网上冲浪的时候, 你可能会看到许多和 JavaScript 相关的 Web 站点在讨论相同主题: JavaScript 不是 Java。我们在前面已经提到过 JavaScript 和 Java 是两个不同公司的产品 (JavaScript 是 Netscape 公司的, Java 是 Sun 公司的), 名称上的相似只是出于市场的原因。在后面的学习过程中, 你会体会到 JavaScript 和 Java 更多的不同。但我们可以从几个方面简单了解一下 JavaScript 和 Java 的区别:

- JavaScript 和 HTML 文档是紧密集成的。JavaScript 代码使用 <SCRIPT> 标签完全嵌入 HTML 文档中。Java 编写的 applet (Java 小应用程序) 程序代码存放在一个单独的文件中, 在 HTML 文档中使用 <applet> 标签建立 Java 小应用程序和 HTML 文档的联系;
- Java 比 JavaScript 更像一个完整的程序设计语言。Java 是强类型的、真正的面向对象、有自己的编译器, 主要用于编写完整的应用程序和应用于 Web 页中的 Java 小应用程序。JavaScript 主要用于编写 Web 页脚本。

从语言本身的角度讲, JavaScript 的语法非常像 Java, 所以两者在学习上有相通之处。

1.2.8 JavaScript 的作用是多方面的

JavaScript 在 Web 中的作用是多方面的, 可从下面的几个方面得到体现:

- 可以为 Web 页增加特殊效果、动画或标志;
- 可以在客户端独立完成数据的验证;
- 用于建立客户/服务器应用程序;
- 开发客户端应用程序;
- 将 HTML 对象、Java 小应用程序、ActiveX 控件和 Netscape 插件组合一起使 Web 网页发挥强大作用;
- 扩展 Web 服务器功能;
- 在不使用 CGI 程序的情况下开发客户端数据库应用程序。

1.2.9 JavaScript 是不断发展的

JavaScript 不是一成不变的, 它不断得到更新和加强, 以满足不断增多的需要。到目前为止, JavaScript 已经有了 6 个版本: JavaScript 1.0 到 5.5。JScript 从 1.0 到 5.5 也经过了多个版本。所以在开发 JavaScript 应用程序过程中, 不仅需要考虑浏览器是否支持 JavaScript, 还需要考虑 JavaScript (或 JScript) 版本之间的兼容性。

如果要随时了解 JavaScript 的发展变化, 你可以访问 Netscape 的网站 <http://developer.netscape.com> 或 <http://www.mozilla.com>。Microsoft 在其网站 <http://msdn.microsoft.com/scripting> 中为用户提供 JScript 的最新信息。

1.2.10 JavaScript 的应用非常广泛

作为一种脚本语言, JavaScript 的应用是非常广泛的。也许许多人都会认为 JavaScript 是用来编写客户端的 Web 应用程序 (这也是本书的重点), 正如我们在前面提到过的一样, JavaScript 还可以用在 Netscape Enterprise 服务器和 Microsoft 的 ASP 中编写服务器端的应用程序。在一些 Web 开发工具中, 如 Borland 公司的 IntraBuilder 和 Macromedia 公司的 Dreamweaver (2.0 及更高版本), 也将 JavaScript 作为脚本开发语言。

另外, 最新的 Windows 操作系统 (Windows 98、Windows 2000) 也支持 JavaScript。通过使用 Windows Script Host (WSH), 你可以编写 JavaScript 程序来自动完成需要执行的任务。所以, 不要老是想 JavaScript 只能编写客户端 Web 应用程序。

1.3 World Wide Web

在过去的 6 年中 Web 技术不断的发展, 使 Internet 世界已经发生了巨大的变化。最初的 Web 页只不过是简单文字, 而现在 Web 已经是一个丰富多彩的世界了。在我们进一步学习如何编写 Web 应用程序前, 先让我们了解一下 Web 的发展过程。Web 的发展大致可以分成四个阶段:

- 第一个阶段: 基于字符的超文本:

第一个 Web 页出现于 1989 年, 由于条件的限制, 人们在使用计算机访问 Web 时根

本没有更好的方法来显示和处理图形。所以人们只能选择更容易的方式来通过 Web 共享信息——使用文本。在那时的 Web 页中，人们使用高亮度的文本来代表超级链接，超级链接可以让人们很容易跳转到相关的 Web 页中去。随着 Web 技术的进步，超文本技术已发生了革命性的变化。

- **第二阶段：基于图形的 HTML 文档：**

1993 年出现了第一个支持图形的浏览器 NCSA Mosaic。Mosaic 是一群大学生和 Netscape 的奠基人 Marc Andreessen 共同为国际超级计算机应用中心（National Center of Supercomputing Applications, NCSA）开发的 Web 浏览器。此时 Web 的概念已经在各个科研机构中得到广泛应用。图形浏览器的出现，为 Internet 开辟了新的广阔天地，使得 Web 的访问更加容易、更加具有趣味性。在同一时期图形环境很快超过基于字符的桌面系统并流行起来。

图形桌面系统加上图形 Web 浏览器，这正是人们期盼以久的梦幻组合。在仅仅几个月的时间内，在计算机界掀起一种狂热的竞争：计算机公司、机构开始为 Web 提供内容或是为客户提供 Web 服务。

这时的 Web 仍然是静态的，Web 的内容包括文本、图形，有些还包括一些声音或视频文件。不过这时的声音或视频文件需要单独下载，然后使用外部应用程序来进行播放。

- **第三阶段：Dynamic HTML 文档：**

在 Web 发展的第一和第二阶段中，Web 页是使用 HTML 文本编辑器来编写，然后放在 Web 服务器中供人们访问。放置于 Web 服务器上的 Web 页在大多数情况下都是不会改变的，除非 Web 作者进行修改。这些保持不变的 Web 页一般能满足大多数用户的要求，而对一些特别的用户来说这还是不够的。比如一个为用户提供产品目录信息的 Web 站点，每个用户需要查看的内容可能会各不相同，如果使用长期保持不变的 Web 页显然不能满足用户要求。

为了实现动态产生 Web 页，Web 页作者开始使用公用网关接口（CGI）来编写服务器端的 Web 页应用程序。CGI 程序使得用户和浏览器可以进行低级的交互行为。

- **第四阶段：Active HTML 文档：**

从 1995 年开始，人们开始在 Netscape Navigator 中使用插件和 Java。这一阶段最大的特点就是客户端——浏览器的功能得到加强，不再单独依靠服务器来运行应用程序和处理用户信息。JavaScript、Java、ActiveX 和其它客户端扩展功能的应用，使得客户端对服务器的依赖性减小，使浏览器和服务端成为真正的客户/服务器结构。浏览器成为一个功能强大的可以运行 Web 应用程序的操作环境。

1.4 Web 应用程序结构

以 Web 作为应用程序开发环境正逐渐成为一种趋势。随着 Java、JavaScript、ActiveX 和其它技术的出现，越来越多的人将兴趣集中到开发 Web 应用程序上。使用 Web 作为应用程序开发环境听起来有点让人感到迷惑，因为 Web 具有分布性的特点，所以 Web 应用程序的构成是多方面的，需要多种技术的支持。

在基于局域网的客户/服务器机构中，非常典型的应用可能就是基于客户端的数据库网络应用程序。也许你曾经使用过一些单独的工具开发过客户端应用程序，如使用 Visual Basic 或 Delphi。而在服务器端的典型开发和维护工具恐怕要数 SQL Server。

图 1-2 显示了 Web 应用程序的各个组成部分。Web 应用程序结构可以看成是各种 Web 技术的集成。开发 Web 应用程序使用的每一种技术都具有一定的局限性，应用范围相当狭小，但它们在一起使用，则可以提供有效的方法来解决 Internet 或 Intranet 应用程序的开发问题。

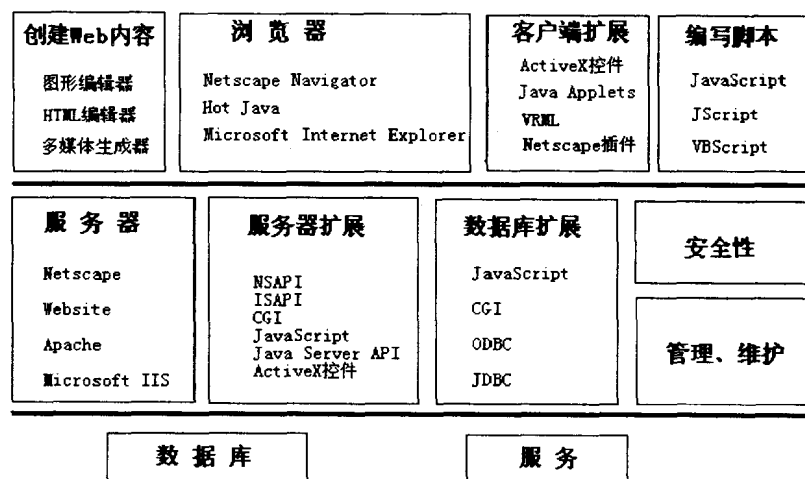


图 1-2 Web 应用程序结构

1.4.1 客户端 Web 应用程序

客户端 Web 应用程序可以划分成以下四个部分：

- 浏览器；
- HTML（Hypertext Markup Language，超文本标记语言）；
- 客户端扩展（Java Applets、ActiveX 和 Netscape 插件）；
- 脚本编写语言（JavaScript、JScript 和 VBScript）。

图 1-3 显示了客户端的各种 Web 技术是如何一起工作的。下面我们将分别介绍客户端的各种技术。

1. 浏览器

浏览器毫无疑问是 Web 应用程序中最重要的组成部分。用户访问 Web 页，Web 应用程序的运行，都必须在浏览器中进行。浏览器的技术相对简单一些，只需要能够正确地 Web 服务器下载 Web 页，并将其显示在浏览中。浏览器并没有一个同一的标准，当前最常用的浏览器是 Netscape Navigator 和 Microsoft Internet Explorer。不同的浏览器对各种 Web 技术有不同程度的支持，所以在不同的环境中选择浏览器对开发 Web 应用程序起着关键作用。

如果开发的是 Intranet (内网) 的 Web 应用程序, 那你可以确定客户端的用户使用一个标准的浏览, 如使用 Netscape Navigator 或 Microsoft Internet Explorer。在确定了浏览器的前提下, Web 应用程序的开发将相对简单。而当我们开发的 Web 应用程序发布到 Internet 上, 由于客户端用户使用的浏览器是多种多样, Web 应用程序的开发将变得复杂起来。表 1-1 列出了各种浏览器对各种 Web 技术的支持情况。

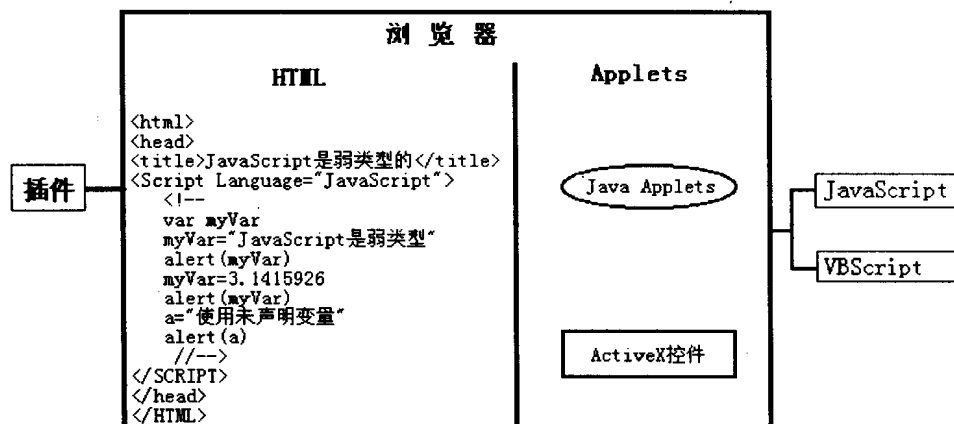


图1-3 客户端 Web 技术

表 1-1

	Netscape Navigator	Microsoft Internet Explorer
JavaScript	2.0+	3.0+
VBScript	不支持	3.0+
Java	2.0+	3.0+
ActiveX 控件	3.0+ 插件	3.0+
Netscape 插件	2.0+	3.0+

2. HTML

HTML 也是进行 Web 应用程序开发的主要技术之一。HTML 是一种标记语言, 严格地讲 HTML 并不能称之为语言。HTML 是使用各种预先定义的标记对文本进行标注, 被标注的文本相当于进行了格式化。HTML 的特点就是简单, 正是这种简单的特性使它更易于使用。

被标注的 HTML 文档最终在浏览器中显示出实际的效果。如果仅使用 HTML, 我们可以创建静态的 Web 文档, 如返回服务器端的查询结果、使用反馈表单或者显示 JavaScript 应用程序结果等。Web 应用程序的开发是离不开 HTML 的, 比如 JavaScript 应用程序便需要使用 HTML 标签嵌入到 HTML 文档中才能起作用。

3. 客户端扩展

客户端扩展功能是扩展浏览器的功能, 随着活动 Web 页功能的不断地加强, 仅仅靠

加强浏览器的功能还是不够的，所以需要对浏览器的功能进行扩展。客户端的扩展都是在浏览器中使用第三方的附加软件来加强浏览器的功能。第三方附加软件需要和浏览器一起工作，但并没有和浏览器绑定到一起。

现在出现的第三方客户端扩展功能在功能上具有相似之处，但仍然有各自的特点。主要的客户端扩展功能有：

- Java applets
- ActiveX 控件
- Netscape 插件

Java 一出现，便迅速得到了广泛的应用。如果你从未听说过使用 Java 来编写程序，那就是真正的孤陋寡闻了。Java 是 SUN 公司开发的具有独立平台的程序设计语言。Java 吸引众多编程人员的一个最重要的特点就是使用 Java 编写的应用程序可以在不同的平台上运行。在浏览器环境中运行的 Java 应用程序我们称之为 Applets（小应用程序）。

Java 小应用程序通过使用 HTML 标签<applet>在 HTML 文档中进行引用。当用户在浏览器中引用了 Java 小应用程序的 Web 页时，Java 小应用程序被下载到用户使用的客户机中。Java 小应用程序是以字节代码的形式被下载到客户机中的，如果浏览器支持 Java，浏览器将对这些字节代码进行翻译并执行它们。如果浏览器不支持 Java，Java 小应用程序将会被忽略，用户将看不到 Java 小应用程序的效果。

ActiveX 控件最早的名称是 OCX，它是 Microsoft 在技术上与 Java 小应用程序的竞争结果。ActiveX 控件和 Java 小应用程序在 Web 中的应用在功能方法上非常相似，都是在浏览器中提供可执行的内容。ActiveX 控件与 Java 小应用程序最大的不同就是 ActiveX 控件只能在 Microsoft 的 Windows 操作系统中使用。ActiveX 控件虽然有操作系统平台的限制，但它并不仅仅应用在 Web 应用程序的开发中。ActiveX 控件还可以应用于 Windows 程序设计语言中，如在 Visual Basic 或 Delphi 中都可以使用 ActiveX 控件。

插件是一种与其它种类稍微有点不同的技术，但它仍是浏览器在客户端的扩展。插件实际上是对 Netscape Navigator 浏览器的常规功能进行了扩展，对附加数据类型和其它的一些特点进行支持。一个突出的例子就是我们可以使用插件在 Netscape Navigator 浏览器中显示特定的 MIME（Multipurpose Internet Mail Extension protocol，多用途的网际邮件扩充协议）类型的文件。在 Netscape 安装文件夹下，Netscape 建立了一个目录 Netscape\Communicator\Program\Plugins，用于存放在系统中注册了的插件。当我们在浏览器中浏览的内容需要插件来显示时，首先在插件文件夹中搜索是否已经有注册了的插件，如果没有，Navigator 会自动从网络中下载需要的插件，然后进行显示。插件安装或从网络中下载后，就成为了 Navigator 的浏览器的一部分。插件的运行不需要用户的干涉，用户也看不到插件是怎样运行的。插件最主要的作用是对多媒体数据的支持，如播放声音、视频，显示图像等。我们将在后面的章节中学习到如何使用 JavaScript 与插件进行交互作用。

4. 客户端脚本语言

客户端脚本语言也是 Web 客户端的一种重要的技术。JavaScript 现在仍然是最常用的脚本语言，但 Microsoft 也推出了 VBScript。由于使用 Visual Basic 进行编程的开发人员