



JavaScript 与 HTML 实用教程

罗小明 付胜兵 吴磊 编著



国防工业出版社

P312

1980

JavaScript 与 HTML

实用教程

罗小明 付胜兵 吴 磊 编著

国防工业出版社

· 北京 ·

图书在版编目(CIP)数据

JS233/17

JavaScript 与 HTML 实用教程/罗小明等编著. —北京:
国防工业出版社, 1998. 1
ISBN 7-118-01800-7

I. J... I. 罗... II. Java 语言 IV. TP312

中国版本图书馆 CIP 数据核字(97)第 15851 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)

北京北商印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 22½ 517 千字

1998 年 1 月第 1 版 1998 年 1 月北京第 1 次印刷

印数: 1—4000 册 定价: 30.00 元

(本书如有印装错误, 我社负责调换)

21

前 言

目前,World Wide Web(全球信息网,又称 WWW)是 Internet 上数百万人用来交换信息的媒介。利用它,人们可以共享文本、图像、声音等数据。此外,越来越多的人还试图使他们的 Web 页面具有交互性。从许多方面讲,我们正处于从以文档为中心的计算向以网络为中心的计算迁移这样的信息革命中,该迁移的焦点就是 Netscape Communications 以及日趋流行的 Web 浏览器,即 Netscape Navigator。Navigator 2.0 是 Netscape 的 Web 浏览器的功能最强大的版本。Navigator 2.0 除了组合进诸如邮件客户、新闻阅读器,以及用于开发 HTML 3 标准的改进支持这样极为有用的 Internet 访问工具外,还增加了几种新的功能,从而提高了 Web 开发者开发完整的、与平台无关的应用程序的能力,这些应用程序在 Web 浏览器中被调用和执行。Navigator Gold 除具有 Web 浏览器的功能外,还在软件包中增加了编辑与开发工具。

这些新增加的功能中,包括 1 个用于插件(plug-in)的应用程序的编程员的界面(API)。插件是可动态地扩充 Navigator 的功能以处理新型数据和信息的程序模块。如果与 JavaScript 和 Java 一起使用,插件还可给 Web 页面增加灵活的编程能力。

通过将 Sun Microsystems 公司的 Java 与其本身的 JavaScript 脚本编写语言组合到 Netscape Navigator 浏览器的当前版本中,Netscape 促进了 World Wide Web 的迅速发展,这种发展是以建立交互式文档与信息为目的的。

Java 是一种与平台无关的编程语言,可以开发在 Internet 上传播的应用程序。JavaScript 目前提高了日趋流行的 Netscape Navigator Web 浏览器的功能,使得 Web 开发者能够创建随着用户动作的不同而改变的内容,即创建使信息更易于访问、更易于组织与吸收的动态数据。

本书首先概要介绍 Netscape Navigator,然后深入讨论 JavaScript 语言的各个方面,并说明如何将它们应用于 Web 上的实际网景中。

本书适合于已具有使用 HTML 以及设计 Web 站点(包括使用 Netscape 扩展功能)经验的 Web 开发人员。本书会根据需要对所用到的高级或复杂的 HTML 标识符进行解释。

很自然,编程知识以及对 Common Gateway Interface(公共网关接口——用于在服务器终端上增加交互性,又称 CGI)的理解会有助于学习本书中对 Web 开发的讨论,但对于学习 JavaScript 来说不是很重要,对 HTML 标识符有适当了解的读者即可学习本书课程。学完本书后,读者不仅能写出复杂的 JavaScript 程序,而且能学会使用 Netscape 对 HTML 扩充的一些较新功能。

对 JavaScript 这样的编程语言感兴趣的并不仅仅局限于 Web 开发者与设计者,越来越多的希望掌握最新技术的编程人员热衷于学习 JavaScript。此外,试图在图形用户界面领域利用 Navigator 推广与平台无关技术的应用程序开发者,将很快发现,JavaScript 在实

现他们的应用程序时将起到关键的作用。

除署名作者外,参与本书编写的人员还有:朱华健、魏宁、刘焯、李蕾、朱志坚、唐磊。全书由庞有善教授和朱尽染研究员审核。此外,李纪鸿为本书的编排付出了辛勤劳动。在此对以上同志深表感谢。

限于水平和时间,书中的错误和不妥之处,敬请读者批评指正。

作 者

目 录

第一章 JavaScript 概述	1
1.1 Netscape Navigator	1
1.1.1 从哪里得到 Netscape Navigator	1
1.1.2 框架	2
1.1.3 插件	3
1.1.4 Java 及 JavaScript 的作用	4
1.2 Java 与 JavaScript 的比较	5
1.2.1 Java——综合、完整的面向对象的编程语言	6
1.2.2 JavaScript——简单、以对象为基础脚本编写语言	7
1.3 JavaScript 的现状	8
1.4 JavaScript 的特征	9
1.5 JavaScript 中对象的使用	10
1.5.1 内置对象的使用	10
1.5.2 自定义对象的使用	11
1.6 JavaScript 的优缺点	12
1.7 小结	13
第二章 编写一个简单的脚本	15
2.1 将 JavaScript 组合到 HTML	15
2.1.1 SCRIPT 标识符	15
2.1.2 将 JavaScript 包括到 HTML 文件中的方法	16
2.1.3 将脚本隐藏于其他浏览器的方法	16
2.1.4 SCRIPT 标识符存在的问题	16
2.1.5 JavaScript 代码放入位置	17
2.1.6 为 JavaScript 程序使用外部文件	17
2.2 JavaScript 的基本命令语法	20
2.3 JavaScript 的命令块	20
2.4 输出文本	21
2.5 超出文档窗口的限制	25
2.6 对话框的使用	25
2.7 与用户交互	27
2.8 小结	30
第三章 数据类型、变量、表达式与 if 结构	31
3.1 数据类型	31
3.2 类型转换	33
3.3 变量	33

3.3.1 变量的声明	33
3.3.2 变量在脚本中的使用	34
3.4 表达式	37
3.4.1 赋值表达式	37
3.4.2 运算符	38
3.4.3 运算符的优先级	40
3.5 表达式与运算符范例	41
3.6 if-else 结构	43
3.7 小结	46
第四章 函数与对象	47
4.1 函数及其使用	47
4.1.1 函数的定义	47
4.1.2 参数传递	48
4.1.3 函数的返回结果	49
4.1.4 文件头部的函数	49
4.1.5 函数应用范例	49
4.1.6 递归函数	51
4.2 对象	54
4.2.1 对象的属性	55
4.2.2 作为对象属性的对象	55
4.2.3 对象的方法	56
4.2.4 如何定义自己的对象	57
4.3 关联数组	59
4.4 综合范例	62
4.5 小结	70
第五章 事件	71
5.1 事件及其处理程序	71
5.1.1 this 关键字	72
5.1.2 事件处理程序	72
5.1.3 模拟事件	73
5.1.4 事件处理程序 onLoad 与 onUnload	74
5.1.5 window.open() 的使用	76
5.2 事件与窗体	78
5.3 事件程序范例	79
5.4 小结	81
第六章 交互式窗体	82
6.1 form 对象	82
6.1.1 form 对象的属性	82
6.1.2 form 对象的方法	83
6.1.3 form 对象的事件处理程序	84
6.1.4 窗体元素的使用	85
6.2 利用表格建立计算器	95

6.3	elements[]数组	100
6.4	建立乘法表范例	100
6.5	forms[]数组	102
6.6	带文本字段的提示	103
6.7	交互式窗体范例	106
6.8	小结	114
第七章	循环	115
7.1	for 循环	115
7.1.1	for...in 循环	117
7.1.2	for 循环范例	118
7.2	while 循环	120
7.3	break 和 continue 语句	125
7.4	循环范例	126
7.5	数组	136
7.6	小结	136
第八章	框架、文档和窗口	137
8.1	框架	137
8.1.1	FRAMESET 标识符	138
8.1.2	FRAME 标识符	138
8.1.3	嵌套框架	140
8.1.4	NOFRAMES 标识符	141
8.1.5	命名框架	143
8.1.6	在 JavaScript 中框架的使用	143
8.2	嵌套框架集应用范例	152
8.3	document 对象	162
8.3.1	BODY 标识符	162
8.3.2	document 对象的属性	163
8.3.3	document 对象的方法	165
8.3.4	在颜色测试器中 document 对象的使用	165
8.4	window 对象	168
8.4.1	状态条的使用	169
8.4.2	窗口的打开和关闭	169
8.4.3	用 Timeouts 暂停	171
8.5	状态条消息处理程序的建立	171
8.6	Navigator 2.0 的颜色处理	177
8.7	应用程序范例	177
8.8	小结	182
第九章	Cookies 的使用	183
9.1	何谓 Cookies	183
9.1.1	超文本传送协议 HTTP	183
9.1.2	Cookies 与 HTTP 头部	184

9.1.3	Cookie 与 Set-Cookie	184
9.1.4	有关 Cookie 的例子	185
9.1.5	Cookie 与 CGI 脚本	185
9.2	在 JavaScript 中 Cookie 的使用	187
9.2.1	在 Cookie 中存储用户的选择	187
9.2.2	对 Cookie 进行编码	192
9.3	建立新的搜索页面	196
9.4	navigator 对象	212
9.5	记事日历程序	213
9.6	小结	224
第十章	string、Math 和 history 对象	225
10.1	string 对象	225
10.1.1	length 属性	225
10.1.2	string 对象的方法	225
10.1.3	建立搜索与替换工具	233
10.2	Math 对象	240
10.2.1	三角方法	241
10.2.2	log() 与 exp() 方法	242
10.2.3	abs() 方法	242
10.2.4	计算几何测量	242
10.3	history 对象	246
10.4	小结	247
第十一章	游戏程序的设计	248
11.1	卡通游戏的设计	248
11.2	火柴游戏的设计	256
11.3	小结	260
第十二章	电子表格程序设计	264
12.1	通用电子表格程序	264
12.1.1	通用电子表格脚本	264
12.1.2	基本脚本功能的扩充	278
12.2	成本核算工作表程序	287
12.3	小结	298
第十三章	关于 Navigator Gold 2.0	308
13.1	简介	308
13.2	Navigator Gold 2.0 的浏览器窗口与编辑器窗口	310
13.3	用 Navigator Gold 2.0 建立 HTML 文档	312
13.4	用 Navigator Gold 2.0 开发 JavaScript 应用程序	316
13.5	小结	317
第十四章	从 JavaScript 到 Java	318
14.1	Java 的基本概念	318
14.2	用 APPLET 标识符在 HTML 中组合 Java Applet	322

14.3	Java 的使用	324
14.4	从 JavaScript 到 Java	326
附录 A	词汇表	328
附录 B	Internet 网上的 JavaScript 资源	331
附录 C	JavaScript 语言参考	333
C.1	对象	333
C.2	函数、运算符与保留字	344
C.2.1	函数	344
C.2.2	运算符	344
C.2.3	保留字	346
参考文献		348

第一章 JavaScript 概述

JavaScript 是 Navigator 2.0 中的关键组件之一,它帮助 Web 开发人员在 Internet 上生成交互式应用程序。Navigator 以插件、框架和 Java 为特征。Java 是编译型的、面向对象的、复杂且不同于 HTML 的语言;JavaScript 是解释型的、以对象为基础的、易于使用与学习且紧紧集成于 HTML 的语言。JavaScript 将当前浏览器会话的属性提供给了编程者,这些属性包括当前加载的 HTML 页面的元素,如窗体、框架以及链接。

本章将讨论 JavaScript 语言的主要特征和主要优缺点,详细介绍在 JavaScript 中什么是对象以及如何使用对象,还将全面介绍 JavaScript 中的内置对象。读者将会看到,JavaScript 有许多优点,其中包括开发周期短,与平台无关,但它也存在一些缺点。

1.1 Netscape Navigator

虽然 Netscape Navigator 是从基本的 Web 浏览器问世的,但随着它的日益流行,该软件已具有更多的功能。与早期浏览器以及当今的基本 Web 应用程序不同的是,Navigator 2.0 给开发者提供了众多的工具,从而超出了 HTML 的传统约束。利用 Navigator 2.0,开发者不再只是将文本、图画、声音以及图像做简单的组合,还可以对文档的布局、字体以及颜色做精确的控制;能够利用插件和 Java 扩充浏览器的功能;能够通过使用 JavaScript 来生成交互式的应用程序。

只要浏览一下 Netscape Web 站点就会看到,当今的 Navigator 所能做的工作远远超过了早期的版本,甚至使 Web 开发者无需进行专门的编程。采用软件公司提供的免费插件,Web 开发者就可以在他们的文档中加入正宗的 CorelDRAW 图形和 Microsoft Word 文件,还可以观察 VRML(虚拟现实建模语言)世界,以及用 Adobe 的与设备无关的 Acrobat 格式化的文档。

最为重要的是,Navigator 2.0 提供了一些工具,Web 页面开发者和编程者能够利用这些工具来改善他们的文档,增加他们与 Internet 上提供的信息的动态交互。

1.1.1 从哪里得到 Netscape Navigator

目前,只能在 Netscape 的 Navigator 2.0 Web 浏览器中找到 JavaScript 功能。Navigator 2.0 已于 1996 年初从 β 版软件过渡为实际发行的软件。为了学好本书中的内容,读者应复制一份 Navigator 2.0,以便上机试验本书中的各个例子。在大多数计算机平台上可得到 Navigator 2.0,其中包括 Windows,Mac OS 的所有版本,以及包含 Sun OS,Solaris 以及 Linux 的多种 UNIX 变种。目前,Netscape Navigator 本身没有 OS/2 版本。

如果需要下载 Navigator 2.0 当前版本的拷贝,可从位于 <http://home.netscape.com/>

的 Netscape 主页面或从 Netscape 的众多 ftp 服务器或它们的许多镜像中得到：

```
ftp://ftp.netscape.com/
ftp://ftp2.netscape.com/
ftp://ftp3.netscape.com/
ftp://ftp4.netscape.com/
ftp://ftp5.netscape.com/
ftp://ftp6.netscape.com/
ftp://ftp7.netscape.com/
ftp://ftp.leo.chubu.ac.jp/pub/WWW/Netscape/(Japan)
ftp://sunsite.ust.hk/pub/WWW/Netscape(Hong Kong)
ftp://sunsite.huji.ac.il/Netscape/(Israel)
ftp://ftp.adelaide.edu.au/pub/WWW/Netscape/(Australia)
ftp://sunsite.doc.ic.ac.uk/computing/information-systems/WWW/Netscape/(United Kingdom)
ftp://ftp.informatik.rwth-aachen.de/pub/mirror/ftp.netscape.com/(Germany)
ftp://wuarchive.wustl.edu/packages/WWW/Netscape/(U. S. A. )
ftp://sunsite.unc.edu/pub/packages/infosystems/WWW/clients/Netscape(U. S. A. )
```

1.1.2 框架

框架是 Navigator 2.0 中最值得注意的 HTML 的扩充。开发者可以利用框架将屏幕分成多个矩形部分,各个部分彼此独立,且在每一框架内加载不同的 URL。此外,可以利用某一框架内的链接更新另一个框架,而且不会干扰其他框架。

目前,人们利用框架给 Web 页面、给总是位于屏幕上但不需要重新加载与卸载的菜单以及给总能使用窗体的以窗体为基础的检索提供固定的、标签式的题头。

除了框架技术的明显的可视化要求以外,该 HTML 扩充还可以减小需从服务器请求或需传送到客户处的数据量。对于急剧拥挤的 World Wide Web 来说,这种减少可稍微提高 Web 浏览的效率,并且会减轻过载的 Web 服务器上的一些负载。

图 1.1 给出了一个如何利用框架来分割屏幕以及在若干个具有动态信息的框架邻近

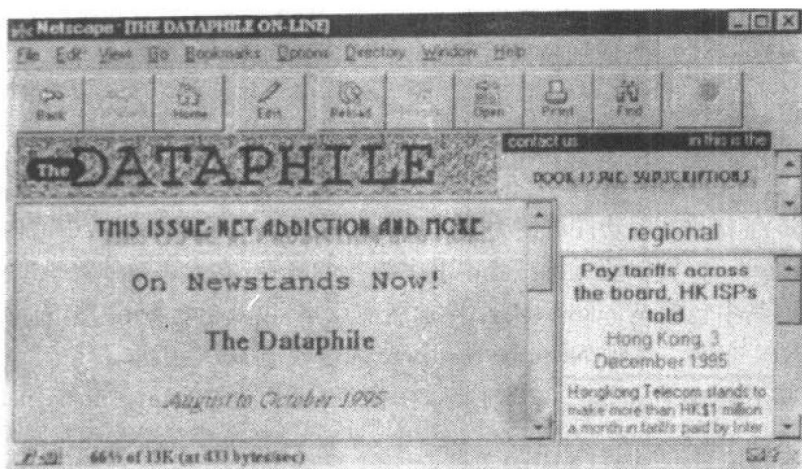


图 1.1 利用框架提供固定的和动态的元素

提供固定题头与永久性菜单的例子。从图 1.2 可以看到,通过选择菜单项能够更新屏幕左下角部分的信息,但不会影响窗口中的其余部分。

为了更好地理解框架的概念,可到下列站点地址去查看 Netscape 对框架的介绍:

http://home.netscape.com/comprod/products/navigator/version_2.0/frames/index.html

在该页面中有若干关于使用框架来改进信息访问的例子。

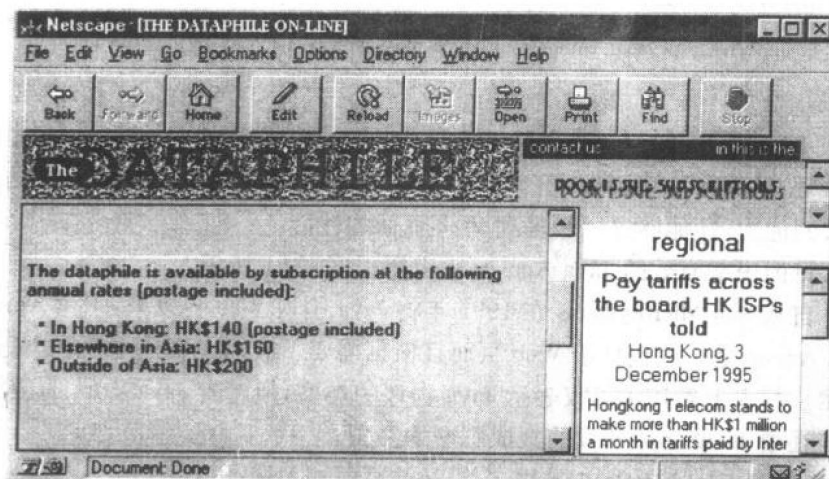


图 1.2 利用菜单更新某一框架

1.1.3 插件

Navigator 2.0 的另一个特征是开放的插件技术。Netscape 通过给插件提供一个开放的应用程序编程界面,使得第三方软件销售商能够在 Navigator 浏览器窗口中提供观察多种

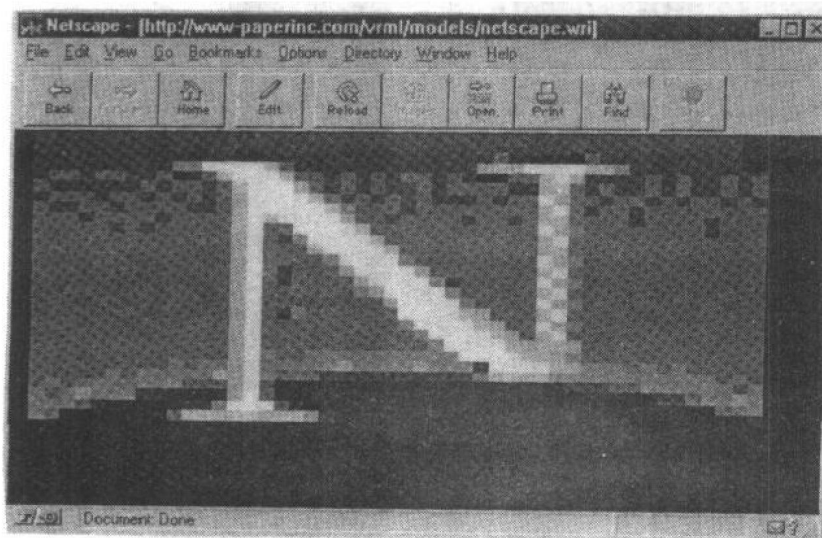


图 1.3 Paper Software 公司的 WebFx™ 插件

数据与文档的能力。Netscape 把通过插件显示的对象称为 Live Objects(活对象)。

到此为止,Corel,Paper Software 以及 Adobe 等销售商已开发出了具有自己格式的插件(图 1.3 给出了其中一种公司的格式)。其中许多插件能够扩充来支持 JavaScript,使得 JavaScript 程序能直接与插件交互,并在 Netscape Navigator 的新版本中进一步提高 JavaScript 的性能。

由于可用插件的数量不断改变,Netscape 在下面站点中提供了最新的可用插件清单:

http://home.netscape.com/comprod/products/navigator/version_2.0/plugins/index.html

1.1.4 Java 及 JavaScript 的作用

利用 Java 与 JavaScript,Web 开发人员具有在单独的框架中加载多个文档以及观察新文件格式的能力,还能够进一步提高 Web 页面的性能。

目前常见的一些有效的 Java Applet(小应用程序)如下所示。

术语 自从 Sun Microsystems 在 1995 年将 Java 引入 Web 界以来,术语 Applet 已被广泛地使用。Applet 是一些包括于 Web 页面且根据需要下载以由客户浏览器执行的小应用程序。虽然从技术上看该术语仅表示 Java 程序的类型,但随着 JavaScript 的应用,该术语也可用来描述用 JavaScript 开发的集成脚本的类型。

- 滚动文本 Applet:Java 正被用来给 Web 页面加入选择框。
- 动态数据输入:动态股票报价机就是这方面的例子之一,例如由 Sun Microsystems 公司的 Jim Graham 开发的金融有价证券演示,如图 1.4 所示。

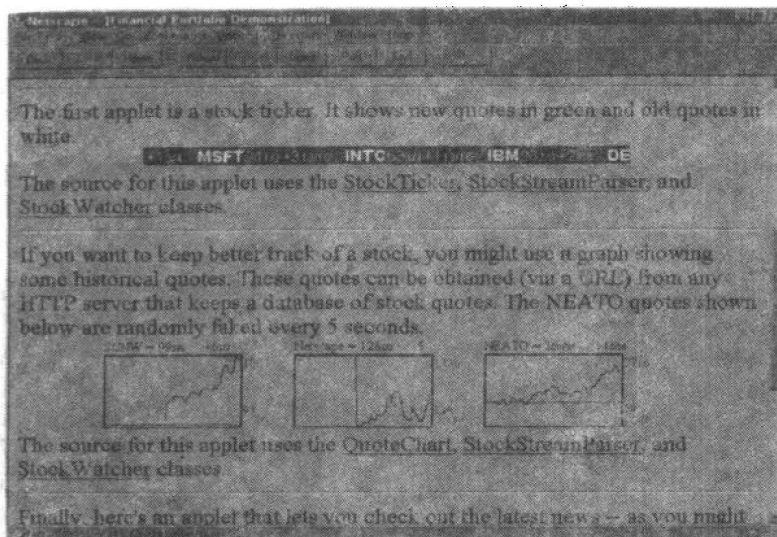


图 1.4 Sun Microsystems 公司开发的 Web 页面动态数据输入

- 搜索引擎:可以利用 Java Applet 建立数据库查询,然后将该查询传送到远程数据库中。
- 给 Web 浏览器加入协议:就像在 Internet 上开发协议一样,Java 所提供的潜在可

能性之一是能根据需要用 Java 扩充 Web 浏览器的功能。

- 新的文件格式:Java 允许 Web 浏览器通过下载显示所需要的文件的 Applet 来显示新的文件格式。

Java 的广泛流行意味着其应用范围在未来将会增大,但目前 Java 主要用于对 Web 页面做装饰性的改进。

相比之下,JavaScript 用来生成对用户以及环境事件起作用的脚本,同时将来会成为使 Java Applet 更平滑地挂到 Web 页面上的粘结剂。下面是一些应用实例。

- Web 开发人员在他们的文档中试验不同的背景颜色与文本颜色的交互式颜色采集器。
- 计算器:用于 Web 的实例,包括单位转换计算器和贷款利率计算器。
- 以当前环境以及用户以前的漫游为基础的动态输出。
- 窗体校验:JavaScript 能够保证所输入的窗体数据在传送到服务器以前是正确的,而不是在数据提交后由服务器来校验窗体的内容。图 1.5 说明了窗体的校验。

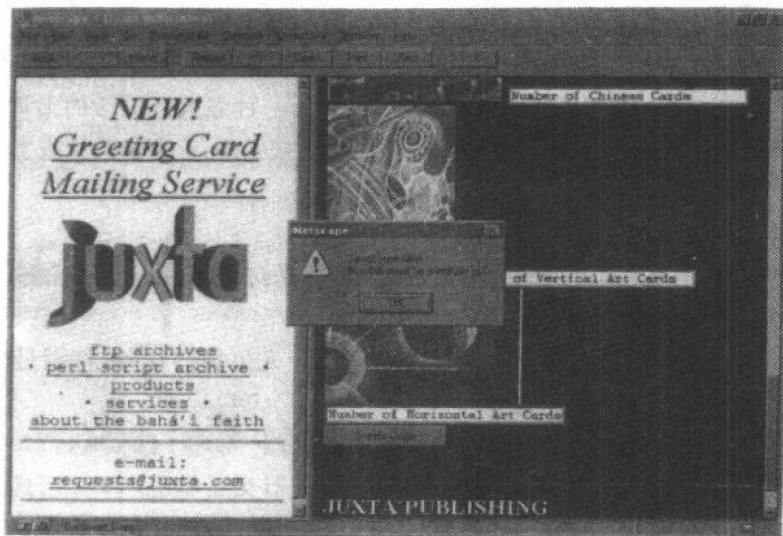


图 1.5 用户在将窗体传送到服务器以前输入正确的信息

- 建立 URL:根据用户在窗体中的选择,用 JavaScript 建立定制 URL。
- 利用 JavaScript 能够为客户端处理替换许多 CGI 脚本,从而缓解繁忙的 Web 服务器的带宽需求,且减小了服务器负载。

1.2 Java 与 JavaScript 的比较

Java 与 JavaScript 之间存在着差别,虽然它们是相关的,JavaScript 借用了 Java 的大多数语法,但它们在本质上是不同的,而且服务于不同的目的。可以说 Java 与 JavaScript 是互为补充的,而不是相互竞争的。

1.2.1 Java——综合、完整的面向对象的编程语言

Java 不仅仅是开发以 Web 为基础的应用程序的语言,它也可以与市场上已趋成熟、一般目的的程序设计语言(如 C, C++, Pascal 以及 FORTRAN)相竞争。与以前的语言不同的是,Java 的平台是独立的,既可以用于应用程序的开发,又可以为 Web 页面开发内置的 applet,或小应用程序。

像 C++ 和 Smalltalk 一样,Java 是面向对象的,它继承了 C++ 的语法与风格。

1. 编译型语言

Java 与大多数编程语言不同,它并不按传统的方式编译。Java 编译器并不将源代码编译成本地机器代码,它会把源代码转变成 Java 字节代码(称为结构中性字节代码),这种代码与运行的平台无关。然后该代码在与设备相关的运行解释器中运行。这样,开发人员只需要开发并维护一套源代码,并将该代码编译一次,编译后的代码就可以通过任何机器的运行解释器来运行。

2. 完全可扩展性

真正的面向对象语言的基本特征是可延伸。也就是说,编程人员能够创建自己的类(即对象成组和数据结构)来扩充作为编程语言部分的基本类。

术语 在面向对象的编程中使用的术语类(Class),表示一套共享共同特性的相关的对象。类以及创建新类的能力使面向对象的编程成为一种强大且灵活的编程模型。

Java 没有超出这种规则,Java 编程者可以按照常规给一组基本工具或类创建它们自己的扩展。

3. 陡峭的学习曲线

面向对象的编程语言具有陡峭的学习曲线,特别是对于非程序员来说更为如此。Java 也存在着这种难度。

初级编程人员的一般感受是,学习 Java 是一件令人生畏的任务,特别是想到 Internet 上有效的联机文档的复杂性时更为如此。

4. 使客户/服务器交互作用

随 Java 配套的一组基本类非常适用于客户/服务器交互作用。Java 已具有与 URL 工作以及同 HTTP 服务器交谈的能力。Java 对 Applet 的支持增加了在客户 Web 浏览器中与用户事件交互作用的能力。

此外,来自于 Sun Microsystems 的演示浏览器 HotJava,演示了如何利用 Java 使浏览器动态地学习处理新协议。

注意 当 Java 仍处于初级开发阶段时,Sun 开发了 HotJava 来演示 Java 在 World Wide Web 中对分布式应用程序的潜能,以及演示浏览器如何动态地学习处理新协议与文件类型。对于 Solaris 和 32 位的 Windows (Windows NT 与 Windows 95),可从位于 <http://www.javasoft.com/> 的主页面得到 HotJava。

5. 开发独立的应用程序与 Applet

Java 能够开发在 World Wide Web 上传送以及在客户 Web 浏览器中执行的 applet,同时,Java 也可以开发完整的、与平台无关的、使用 Java 运行解释器的 GUI 应用程序。

6. 提供高度的安全性

由于 World Wide Web 的高度开放与大众化,对于 Java 和 Java Applet 来说,安全问题

就成为主要的问题。从未知的远程机器中给自己的计算机系统下载应用程序代码并执行它,毕竟具有潜在的危险。这些 Applet 可能包含病毒,而且很可能就是试图销毁数据并使计算机不能运行的应用程序。

为防止这种问题,Sun 从 Java 开发的最初阶段开始,就采取了严格的安全措施。这些措施包括字节代码的校验(保证它们没有违反访问约束等)以及可配置的网络安全性。这种安全性范围包括从不允许网络访问,到限制一个 Applet 只访问初始生成代码的主机,直到完全自由地访问网络为止。

7. 与 HTML 的区别

虽然 applet 是 World Wide Web 的特征之一,且作为内联 Applet 包括在 HTML 文件中,但它们与 HTML 以及 HTML 文件是有区别、相分离的。新的 HTML 标志符强迫 Web 浏览器产生与服务器的一个新连接,并告诉浏览器在 Web 页面的什么地方显示 Applet 的输出,但除此之外,Applet 与 HTML 是有区别、相分离的。

1. 2. 2 JavaScript——简单、以对象为基础的脚本编写语言

JavaScript 比 Java 简单、容易使用,它比 Java 拥有更容易的学习曲线,它具有强大的工具,用户只需稍做努力,就可以给 Web 页面加入交互作用性。

1. 起源于 Java

JavaScript 的出现在很大的程度上归功于 Java。虽然两者的功能范围以及编程风格大不相同,但 JavaScript 的语法和基本结构与 Java 是相似的。JavaScript 最初是作为 Netscape 本身的脚本编写语言以 LiveScript 为名出现的。但在 1995 年末,Sun 认可了该语言,且该语言更名为 JavaScript。

JavaScript 不仅保留了 Java 的基本语法和结构,它也借用了 Java 的大部分流程结构,并采用了某些相同的安全措施,如防止 Applet 写到本地磁盘上。

2. 解释型语言

JavaScript 与 Java 不同,它是解释型语言。在 Java 中,源代码在运行以前编译,而在解释型语言中,JavaScript 中源代码文件则直接在运行时执行。

解释型语言具有若干优点,同时也具有若干缺点。像 JavaScript 这样的解释型语言一般要比编译型语言简单,且易于掌握。由于解释型语言的程序每做一次修改后都需要重新编译,因此解释型语言的程序更易于开发、修改、排除故障。

解释型语言的缺点是,对于某些解释型语言,当程序运行时要对命令进行解释,这会降低程序的性能,但对于 JavaScript 来说,似乎不存在这样的问题。

当脚本下载时,JavaScript 脚本被编译成字节代码(与 Java 相似)并执行。对于大多数脚本来说,我们会感到它的性能是非常优秀的。

3. 不是完全可扩展的

与 Java 不同,JavaScript 不是完全可扩展的。利用 JavaScript 模型足以建立客户端应用程序的有限的一组基本对象、特性、方法以及数据类型。

虽然用户利用 JavaScript 可以建立他们自己的对象并编写函数,但这与 Java 以及其他面向对象的编程语言中的类及继承机制是不一样的。

由于 JavaScript 是一种以对象为基础的脚本编写语言,本书将重点介绍 JavaScript 中