

NRP

北京科海培训中心

[美] Tim Ritchey 著
雷凌翼 孙学东 译
廖卫东 审校

编程指南

JavaScript

for Netscape

清华大学出版社
西蒙与舒斯特国际出版公司

(京)新登字 158 号

北京市版权局著作权合同登记号:01-96-1212 号

内 容 提 要

本书系统地讲述了 JavaScript 和 Java 两种编程语言,介绍 JavaScript 和 Netscape 的对象以及如何将这些对象应用到 WWW 程序设计和 Home Page 的制作。

全书共分两大部分,第 1 部分讲述了 JavaScript 语言以及用 JavaScript 可实现的 Netscape Navigator 的对象;第 2 部分介绍 Java 语言以及如何运行 Java 开发工具的编程实例。

全书旨在使你掌握开发 Web 脚本程序和 Applets 所需的技巧和能力。

Programming JavaScript for Netscape 2.0

Copyright ©1996 by New Riders Publishing.

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher.

本书英文版由 Prentice Hall 出版社属下的 NRP 计算机图书出版公司于 1996 年出版。版权为 NRP 公司所有。本书的中文版版权由 NRP 公司授予北京科海培训中心、清华大学出版社和西蒙与舒斯特国际出版公司合作共同出版、发行。未经出版者书面允许不得以任何方式复制或抄袭本书内容。

版权所有,盗版必究。

本书封面贴有 PRENTICE HALL 激光防伪标签,无标签者不得销售。

书 名:JavaScript for Netscape 编程指南

原著者:Tim Ritchey

译 者:雷凌翼 孙学东

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:北京门头沟胶印厂

发 行:新华书店总店北京科技发行所

开 本:16 印张:17.75 字数:429 千字

版 次:1996 年 11 月第 1 版 1996 年 11 月第 1 次印刷

印 数:0001~6000

书 号:ISBN 7-302-02402-2/TP·1204

定 价:30.00 元

前言

本书向您展示了自从第一个 NCSA Mosaic 软件问世以来,Internet 浏览器最让人吃惊的两个变革——JavaScript 和 Java。Netscape 新版的 Navigator 2.0 支持上述两种为 Internet 所设计的独立于平台的语言。通过将 JavaScript 和 Java 加入到其广受欢迎的浏览器中,Netscape 公司使得应用编程人员和 Web 页面设计者只用 Java 语言就可以编写出在所有适合于 Navigator 2.0 的平台上都可以运行的程序。

作为一个 Web 页面设计人员,你总是想尽可能地让你的 Web 页面跟上最新的技术潮流。这本书将助你纵览这两种令人兴奋且功能强大的编程语言,并且设计出动态和交互的 Web 页面;之所以把这两者一起介绍,是因为 JavaScript 是建立在 Java 基础上的,两者的语法极其相似,也就是说,学完 JavaScript 后,学 Java 就极其容易。你可以首先学 JavaScript,并立即用它实现动态交互的 Web 页面。与此同时,这本书的 Java 部分也对你的 JavaScript 知识有所增益,从中可以看出这两种语言的异同之处。学完 Java 后,你就可以在你的 Web 页面上加入 Java Applet。这时,你就可以在你的 Web 页面设计上运用自如;该用 JavaScript 的地方用 JavaScript,该用 Java 的地方用 Java。两者的混合编程,一定会使你得到最满意的 Web 页面。

本书的目的

本书旨在让你学会 JavaScript 和 Java 这两种编程语言,同时帮助你建立开发 Web 脚本程序和 Applets 所需的技巧和能力。

假如你是个想发挥这两种语言最大潜能的 Web 页面设计者,本书将告诉你如何用 JavaScript 实现用户及其顾客之间的交互作用。把你的脚本程序下载到客户的机器上,你就可以放心大胆地放弃复杂而又繁琐的 CGI 脚本程序,这意味着你可以摆脱在维护 HTML 文档和保证服务器上的各种编程文件和目录不被破坏之间的两难境地。如果使用 JavaScript,事情将会很简单,所有的程序将在文档中编制完成。需要改动时,只要载入这个文档,对它进行简单的操作,就像任何其他 HTML 文档一样。逐渐掌握了 JavaScript 之后,你会发现在你面前呈现的将是一个更复杂,但更有力的 Java Applet 的世界。

作为一名经验丰富的编程人员,如果你想学习这两种久闻大名的新型语言,那么毫无疑问你应该阅读这本书,因为它将教会你如何掌握这两门语言,并且向你指出它们与其他语言,如 C 和 C++ 的异同之处。由于一定程度上,Java 和 JavaScript 是以 C++ 为基础发展而成的,因此你应该可以很快转向 Java 和 JavaScript 语言,从而尽快开始用 Java 和 JavaScript 编写程序。

假如你对 JavaScript 或 Java 编程只是略有所闻或是从未接触过,没关系,这本书可以作为一本这两种语言的查询手册来使用。书末的索引部分将会对你很有帮助,特别是如果在编写一个很长的脚本程序,忽然忘了 Java 的某个方法或其参数的时候。

本书结构

本书分为两部分。第一部分从第1章到第7章,讲述了JavaScript语言以及用JavaScript可实现的Netscape Navigator的对象。

第1章 Java 家族简介

第2章 JavaScript 基础

第3章 JavaScript 的控制流和函数

第4章 用JavaScript使用和实现对象

第5章 面向对象的JavaScript

第6章 Netscape Navigator 的对象:文档对象

第7章 表格、历史和存储地址对象

第8章到第13章介绍的是Java语言以及运用Java Development Kit的一些Java编程实例。

第8章 向Java转化

第9章 Java 基础

第10章 Java 的对象——类

第11章 多线程和例外情况处理

第12章 Java 语言基础类库、Utility 类库和 I/O 类库

第13章 使用 applet 类库和 awt 类库

附录是有关 Javascript 语言和对象的层次参考资料。

本书约定

开始阅读本书以前,你应该了解一些本书印刷所使用的几种不同字体:

- 小字体印刷的是源代码。
- 在源代码中用斜体字印刷的词代表变量——由于本身地址和其他因素不同,这个词将会随之改变。
- 斜体字也用于指明一个新词,它的后面跟着的就是它的定义。

同时,这本书还用了一些图标,分别用以标明注释、提示和警告。作者使用它们是为了告诉你一些有关知识。

注释 用以告诉你一些有用信息。

提示 向你提供快速掌握Java的捷径。

警告 提醒你在一个进程中,一个特殊的步骤可能会引起何种危险,对此必须仔细阅读。

知道以上这些以后,你读这本书时会减少许多麻烦,而且成效会更大。

开始之前

阅读这本书的 JavaScript 部分时,你只需要 Netscape Navigator 2.0 浏览器以及一个文本编辑器。你可以从 <http://home.netscape.com> 下载一些有关资料。开始转向 Java 时,将需要 Sun 公司的 Java Development Kit (SDK)。或者另外的 Java 编辑器/解释器,例如 Metrowerker 的 Code Warrior 或者 Native Intelligence 的 Roaster。

Java 是一种处于 Internet 技术前沿的语言,它已被广泛应用到虚拟现实模型语言 (VRML) 中,一些公司纷纷推出以 Java 为基础的产品或工具,现在是学习 Java 的最好时刻,这样才跟得上时代的发展。

祝你好远,顺利!

Timothy Ritchey

Cambridge, England

作者简介

Tim Ritchey 曾于 Ball 州立大学获得过物理和人类学的学士学位,目前正在英国剑桥大学攻读考古学博士学位。他曾经从事人工智能、高性能并行体系结构以及计算机视觉方面的研究,并发表过有关借助结构化光学开发廉价扫描仪方面的论文。Tim 目前的兴趣主要在人工智能、分布式计算、VRML 等方面,当然也包括 Java。除计算机和考古外,Tim 还喜欢深海潜水、飞行,并热衷于驾驶他那辆 Harley Davidson 摩托车。Tim 的博士论文的研究课题主要包括如何把非线性力学和人工智能应用到考古理论之中。

目 录

第 1 章 Java 家族简介	(1)
1.1 WWW 与 HTML	(1)
1.1.1 Internet 的历史	(2)
1.1.2 WWW 的发展	(2)
1.1.3 HTML 语言的特性	(3)
1.1.4 用 HTML 设计的 Web 页面	(4)
1.2 Java 语言	(6)
1.2.1 Java 语言的特点	(8)
1.2.2 作为 Java applet 的 Hello,World!	(11)
1.3 JavaScript 语言	(14)
1.3.1 JavaScript 语言的特性	(15)
1.3.2 用 JavaScript 实现 Hello,World!	(16)
1.4 Java 与 JavaScript 的比较	(17)
1.5 Netscape 2.0 和 Java 家族	(18)
1.5.1 内置的 JavaScript 解释功能	(18)
1.5.2 内置的 Java 运行时引擎	(19)
1.6 小结	(19)
第 2 章 JavaScript 基础	(20)
2.1 JavaScript 和 HTML	(20)
2.1.1 加入脚本	(21)
2.1.2 载入源脚本文件	(21)
2.1.3 注释掉 JavaScript 源码	(22)
2.1.4 其他与 HTML 相关的考虑	(22)
2.1.5 处理顺序	(23)
2.2 JavaScript 的结构和代码编号	(23)
2.2.1 注释	(23)
2.2.2 常量	(25)
2.2.3 标识符	(28)
2.2.4 关键字	(29)
2.2.5 分隔符	(29)
2.3 变量声明	(30)

2.3.1	用“var”定义变量	(31)
2.4	表达式和运算符	(32)
2.4.1	变量转换	(33)
2.4.2	算术表达式	(34)
2.4.3	布尔类型的逻辑运算符	(36)
2.4.4	实数运算符	(38)
2.4.5	字符串运算符	(38)
2.5	小结	(38)
第3章	JavaScript 控制流和函数	(40)
3.1	JavaScript 的流程控制	(40)
3.1.1	利用模块和注释组织代码	(41)
3.1.2	if-else 语句	(44)
3.1.3	for 语句	(45)
3.1.4	while 语句	(47)
3.1.5	用 break 和 continue 控制循环	(47)
3.2	在 JavaScript 中定义函数	(48)
3.2.1	在函数中使用“return”	(50)
3.2.2	在函数中使用“var”	(50)
3.2.3	变量参数	(50)
3.3	使用函数和事件	(51)
3.3.1	用事件驱动器触发函数	(53)
3.4	小结	(54)
第4章	在 JavaScript 创建并使用对象	(55)
4.1	在 JavaScript 中使用对象	(55)
4.1.1	对象的属性	(56)
4.1.2	对象的方法	(58)
4.1.3	使用 this	(59)
4.2	创建对象	(60)
4.2.1	构造函数	(60)
4.2.2	生成数组	(62)
4.3	为 Web 页面创建一个目录	(64)
4.4	小结	(76)
第5章	JavaScript 的内置对象	(77)
5.1	内部对象与函数	(77)
5.1.1	String 对象	(78)
5.1.2	Math 对象	(82)

5.1.3	Date 对象	(87)
5.1.4	eval, parseInt, parseFloat, escape, unEscape 和 isNaN 函数	(90)
5.2	小结	(91)
第 6 章	Netscape Navigator 中的对象:文档对象	(92)
6.1	Navigator 对象层次简介	(92)
6.1.1	窗口对象的属性	(93)
6.2	文档对象	(94)
6.2.1	文档对象的 HTML 等价物	(94)
6.2.2	文档对象的属性	(96)
6.2.3	文档对象的方法	(103)
6.2.4	文档对象的事件句柄	(106)
6.3	小结	(106)
第 7 章	表格、窗口、历史和位置对象	(107)
7.1	使用表格对象	(107)
7.1.1	属性	(108)
7.1.2	表格对象的方法	(109)
7.1.3	访问表格信息	(110)
7.1.4	表格的元素	(114)
7.1.5	表格对象元素中的事件句柄	(121)
7.2	控制窗口对象	(123)
7.2.1	窗口对象的属性	(124)
7.2.2	窗口对象的方法	(125)
7.2.3	窗口对象的事件句柄	(127)
7.3	使用历史和位置对象	(127)
7.3.1	历史对象	(128)
7.3.2	位置对象	(128)
7.4	小结	(129)
第 8 章	Java 简介	(130)
8.1	设置 Java 编程环境	(131)
8.1.1	用 Java 解释器运行独立应用程序	(132)
8.2	使用小应用程序浏览器	(134)
8.2.1	Netscape 2.0	(135)
8.3	提供 Applet	(137)
8.4	你的第一个 Java 程序——Hello, World!	(140)
8.4.1	介绍类的概念	(141)
8.4.2	HelloWorld 作为 applet	(143)

8.4.3	HelloWorld 作为独立应用程序	(147)
8.4.4	故障查找	(149)
8.4.5	解释器错误	(150)
8.5	小结	(150)
第 9 章	Java 语言基础	(152)
9.1	Java 代码的体系结构	(153)
9.2	Java 标记集	(154)
9.2.1	注释	(154)
9.2.2	标识符	(155)
9.2.3	关键字	(155)
9.2.4	常数	(156)
9.2.5	运算符和分隔符	(158)
9.3	声明变量	(159)
9.3.1	声明整型	(159)
9.3.2	声明浮点型	(160)
9.3.3	声明字符型	(160)
9.3.4	声明布尔型	(160)
9.3.5	声明数组	(160)
9.3.6	作用域	(161)
9.4	运算符	(162)
9.4.1	整数运算符	(163)
9.4.2	布尔型运算符	(166)
9.4.3	浮点数运算符	(167)
9.4.4	数组运算符	(167)
9.4.5	字符串	(168)
9.4.6	转换类型	(168)
9.5	控制流	(169)
9.5.1	if-else 分支语句	(169)
9.5.2	while 和 do-while 循环	(171)
9.5.3	switch 分支语句	(171)
9.5.4	for 循环	(173)
9.5.5	标 签	(174)
9.6	小结	(175)
第 10 章	Java 中的对象——类	(176)
10.1	Java 中的对象	(177)
10.2	Java 中的类	(177)
10.2.1	Java 类中的实例变量	(178)

10.2.2	Java 类中的方法	(178)
10.2.3	静态成员	(181)
10.2.4	对象的创建和销毁	(181)
10.2.5	理解 Java 包装	(185)
10.2.6	在 Java 中使用继承	(186)
10.3	小结	(190)
第 11 章	理解线程和异常情况处理	(192)
11.1	理解多线程	(193)
11.1.1	Thread 类和 Runnable 接口	(193)
11.1.2	在类中使用 Thread 修饰符	(197)
11.1.3	Threadsafe 修饰符	(200)
11.2	在 Java 中使用异常情况处理	(200)
11.2.1	异常情况处理	(201)
11.2.2	运行时异常	(202)
11.3	小结	(203)
第 12 章	Java 的语言基础类库、Utility 类库和 I/O 类库	(204)
12.1	类库简介	(204)
12.1.1	Java 类库	(204)
12.1.2	applet 和 awt 类库	(206)
12.2	Java 语言基础类库	(206)
12.2.1	Object 类	(207)
12.2.2	类型封装类	(207)
12.2.3	Math 类	(211)
12.2.4	String 类	(212)
12.2.5	System 类	(215)
12.3	I/O 类库	(216)
12.3.1	stdin, stdout 与 InputStream, PrintStream	(217)
12.3.2	文件存取	(220)
12.3.3	File 类	(220)
12.4	Utility 类库	(226)
12.4.1	Date 类	(226)
12.4.2	Stack 类	(227)
12.5	java.net 类	(228)
12.6	小结	(231)

第 13 章 使用 applet 和 awt 类库	(232)
13.1 applet 类库	(232)
13.1.1 扩展 Applet 类	(232)
13.2 另一个窗口工具集(awt)类库	(238)
13.2.1 Graphics 类	(238)
13.2.2 Color	(239)
13.2.3 控件.....	(240)
13.2.4 创建用户界面的外观.....	(244)
13.2.5 处理事件.....	(245)
13.2.6 创建一个制表控件(Tab Control)	(246)
13.3 小结.....	(257)
 附录 JavaScript 对象层次参考	(259)
anchor(Client)	(259)
button (Client)	(259)
button	(259)
checkbox(Client)	(259)
checkbox	(260)
Date(Common)	(260)
Date	(260)
document(Client)	(261)
document	(261)
form (Client)	(261)
form	(262)
frame(Client)	(262)
frame	(262)
hidden(Client)	(262)
hidden	(263)
history(Client)	(263)
history	(263)
link(Client).....	(263)
location(Client)	(263)
Math(Common)	(264)
navigator(Client)	(264)
password(Client)	(265)
password	(265)
radio(Client)	(265)
reset(Client)	(265)

select(Client)	(266)
string(Common)	(266)
submit(Client)	(267)
text (Client)	(267)
textarea(Client)	(268)
window(Client)	(268)

第 1 章 Java 家族简介

欢迎来到 Java 大家族。Java 家族包括 Java, 即 Sun 公司推出的开发环境和语言, 和 JavaScript, 一种 HTML(HyperText Markup Language)脚本工具, 由 Netscape 开发。这两种语言支持 WWW 的动态内容开发, 而且成为 Netscape Navigator 2.0 的一部分。因为你正在读这本书, 很明显, 你肯定想知道 Java 能为你做些什么, 能为你的网络的现状做些什么。简而言之——且不谈大的变化——Java 家族能给你的 Web 页面带来新的活力。不管是用 Java 的 applet 还是 JavaScript 的函数, 总之先前许多基于 CGI 脚本的静态 HTML 页面现在已被可提供动态内容并对客户操作进行自动反应的 Web 页面所取代。

WWW 是现今依靠技术驱动的工业中发展最快的多媒体信息传播媒介之一。一年以前它还是计算机行业老板的秘密话题, 而今已是家喻户晓。许多报纸、杂志和电视广告如今都包含了为商业性 Web 页面所做的统一资源定位器(URL)。随着用户的日益增加, 对技术的要求和需求量也越来越高, 昔日建立在文本方式下的终端时代已经彻底结束。这个时代网络公司追求的是结合了立体声的电影视象、以及交互式虚拟现实的界面。这些工作的进展也推动了像 VRML 和 MPEG I 这样的多媒体协议的确立。在这个过程中出现了 Java——一种新的编程语言, 它能进一步利用 Internet 的潜能, 使之更接近于现实。

接下来的部分对 HTML, JavaScript 和 Java 作了简单的总结和比较, 包括它们对网络所带来的新变化, 以及它们各自的不同点。这三种语言可以为发展动态、交互的 Web 页面内容提供一个完整的机制——从 Web 页面的显示, 到事件脚本程序及应用程序的执行。通过懂得如何最有效地利用这些语言, 你可以给那些拜访你的 Web 页面的人以深刻印象。

Netscape 通过在网络浏览器中对 Java 和 JavaScript 的整合, 为实现动态交互的 HTML 页面提供了统一环境。因此, 为了能在用 Java 和 JavaScript 编写程序时对其语言特殊性运用自如, 这一章将讨论这些语言在 Netscape Navigator 2.0 里的具体细节问题, 以及用这些特性运行 Java Applet 和 JavaScript 脚本程序的方法。

1.1 WWW 与 HTML

WWW(World Wide Web)和 HTML 是 Internet 的一个子系统。但由于它是 Internet 中使用最为频繁的协议, 又常被看作是 Internet 的同义词。事实上, 它只是 Internet 上多种信息传输协议中的一种。协议是指在 Internet 上传输信息所用的一种公认的格式, 它规定了数据及其传输的实际格式, 这些协议包括 ftp, gopher, http(World Wide Web), 和 e-mail 等。为领会 JavaScript 和 Java 作为“客户端执行”语言的特性, 弄清 HTML 协议也是很重要的, 这包括弄清它与网络浏览器中 Web 页面的格式和显示的关系。

学习怎样设计 Web 页面时, 设计人员要仔细熟悉的第一件事就是 HTML 的规则。这本书不可能仔细讲述全部 HTML 语言的细节, 因此, 假如你还不知道怎样设计 HTML 的文档, 那么最好找一本易懂的 HTML 的书——比如由 John Wiley and Sons 公司出版的 Ian

Graham 的《The HTML Sourcebook》。这本书会通过例子告诉你当 Java, JavaScript 和 HTML 在一个事例中交互作用时, HTML 是怎样发挥其独特作用的。假如你对 HTML 很熟悉, 你可能会对下面 Internet 导论中的一些基本知识有很深的了解。如果你不太熟悉, 那么你可能会对 Internet 及 WWW 和 HTML 的历史渊源很感兴趣。

1.1.1 Internet 的历史

1957 年, 前苏联发射了 Sputnik 号卫星, 这让美国大为吃惊。为了赶上和缩小在空间竞赛的差距, 美国政府建立了 ARPA 机构——国家高级研究计划局。ARPA 的主旨是承担开发那些对于私人企业而言过于冒风险, 而在这个受技术驱动的时代里又是极其重要的项目。1969 年, 国防部为研究网络协议而建立了 ARPA 网。从那时起, 人们认识到计算机网络需要基本的协议, 于是, 为了满足这种愿望, 1973 年 TCP/IP 协议宣告完成。这一协议至今仍控制着 Internet 的信息传输过程。

TCP/IP 协议, 即传输控制协议/互连网络协议 (Transmission Control Protocol/Internet Protocol), 是不同网络平台得以交换信息的协议集合。在今天的计算机环境下, 处理网络数据的标准是以太网协议。但是, 当 TCP/IP 发展时, 网络计算机的工作还没有标准存在。TCP/IP 协议作为一个独立于硬件的协议将继续流传下去, 作为任何一个以硬件为基础的协议的上层协议使用。这意味着你可以使用以太网协议、ATM 或任何新投入使用的传输数据的硬件方法, 包括以后将会形成的新的协议。

可是, 并不是协议标准使 TCP/IP 盛行起来的。1983 年, 加州大学伯克利分校发行了带有 TCP/IP 协议的 Unix 操作系统版本。ARPA 网上很多机器用的都是 Unix, 于是 TCP/IP 协议无可争议地成了 ARPA 网的基准。有趣的是, 这与现在 Java 和 Netscape 的关系十分相像。如果单独存在, Java 不一定会畅销, 但 Netscape 公司把 Java 加到他们最受欢迎的 WWW 浏览器中, 这就使得这种语言的支持者和开发商的数量越来越多。

1983 年, Internet 的用户共有 500 人。1986 年, 美国国家科学基金会 (NSF) 加入了竞争。他们在五个超级计算中心之间建立了 NSFnet, 传输速度达到 56Kbps。到这一年底, Internet 用户达到 5000 人。1989 年, Internet 的用户已超过 100,000。NSFnet 升级到 1.5Mbps。第二年, ARPA 在 Internet 中的角色结束了, ARPA 网也停止了存在。它所有的功能几乎全都被更强大的 NSFnet 以及 Internet 所取代。Internet 曾经只是一个纯研究的项目, 但现在它成为校园和大公司所关注的司空见惯的事情。

注释 虽然 ARPA 网从最初受到关注到 1992 年达到 1,000,000 用户花了 20 年时间, 但是一年之后, 这个数字就翻了一番。

截止到 1992 年, Internet 上共有 1,000,000 台主机。随后这个数字就有了迅猛的增长, 到 1994 年底便增加到 4,000,000。在 1995 年, NSF 终止了 NSFnet 计划, 政府作为启动这种技术的角色宣告结束, 私营公司则开始蜂涌进入这个行业。在 1995 年 3 月, Internet 上的商用部分 (.com) 首次超过了教育部分 (.edu) 的规模, 标志着 Internet 的发展已带上了商业和营利色彩。

1.1.2 WWW 的发展

当 Internet 开始发展时, 它被公认为只有 Unix 专家和对此有强烈兴趣的爱好者才会使

用的神秘网络。就许多方面而言,的确是这样。有关这样或那样传送信息的协议令人眼花缭乱。但无论如何,人们总是在逐渐尝试使 Internet 变得更可用。80 年代早期,电子邮件和新闻组的应用使非技术用户增加了对 Internet 的日常使用。当然,这些应用都已过时——传输的文件都是静态的文档或图像,它们一经创建,一般就不会改变。电子邮件只不过是信件电子版,而新闻组也仅仅像个电子版的公告牌。随着越来越多的人加入到 Internet,对新的更有效的软件需求的呼声越来越高。

1990 年,来自欧洲粒子实验室 CERN 的 Tim Berners-Lee 发明了一种在 Internet 网上传输多种不同格式文件的协议。这种协议是 WWW 的萌芽。那时,网络仍然是基于文本的界面,直到位于伊利诺斯大学的美国国家高级计算应用中心(NCSA)开发出具有图形界面的客户程序,也就是所说的 Mosaic,文本界面才失去了主导地位。

WWW 很快成为 Internet 的最佳产物。没有人能不为它井井有条的文字以及神奇而令人兴奋的图形界面所吸引——几年以前 Internet 的形象好像已是很遥远的事情了。文本时代已经过去。事实上,命令行以及提示符的时代也都已经过去了!需要做的很简单:点、按、拖——简直像魔术似的。当然,这并不是魔术。你不仅拥有 GIF 图像及各种声音文件,而且一旦你拥有一些浏览器不能操作的文件,你可以在网上找个软件来对付它。除了这些,用 Mosaic 或其他任何一种图形方式的浏览器设计者们正在不断为之增加更多的功能。虽然你只是点和按,所有的工作仍然是由你一个人来完成的。网络提供的内容仍然和以往一样——只不过无论你要做什么变得更简单了。

1.1.3 HTML 语言的特性

驱动 World Wide Web 的新型语言叫超文本标识语言(HTML)。这种语言在文本文件中加入一系列的标记(tag)和引导符(directive)。这些标记和引导符指明了 Web 页面的各种元素将如何显示。用 HTML,可以实现如下功能:

- 实现一个独立于平台的文档
- 实现和 Internet 上其他文档的连接
- 包括图形和多媒体
- 连接 Internet 上的其他资源

首先,最重要的一点就是 World Wide Web 和 HTML 都是独立于平台的。这意味着 Web 页面的设计者可以实现一个在所有和 Internet 相连的 PC 机、Mac 机以及 Unix 机上都能使用的文件。由于 Internet 的异构性,这个特性非常重要。因为 Web 作者不能肯定他们的文件会在哪种终端上显示,如果依赖平台的话,麻烦可就太大了。

“Hyper”指的就是和其他文档相连的能力。超文本(Hyper text)指的是在一个文档里实现顾客一单击按钮就能得到相关信息的链接。这种文档设计的非线性结构是为了有助于学习。通过让读者走自己喜欢的路而不是强迫他们在线性的进程中学习,来增加他们对主题的兴趣。

除在浏览器窗口显示标准文本外,HTML 还提供了使用多媒体信息,包括声音、图像及动画的方法。另外,Netscape 公司提供了在浏览器中嵌入附加软件的能力,以便可以显示更先进的媒体类型,如 VRML 和 Shockwave Director 文件。所以,HTML 文件能够扩展自己的

能力,而不仅仅是用文本定义实现的功能。

HTML 语言还可以移植到未来可能产生的任何新协议中。它使用的是文本标记格式,假如一个浏览器不能识别一个新的标记的话,它会将其忽略。这个特性使得 HTML 可以使用多种网络协议,只要对语言做一定的扩展就行。

对于 Internet 而言,HTML 被证明是一个用于制作文档并使界面具备高度适应性和多样性的语言。当 HTML 刚刚出现时,Internet 和 World Wide Web 还只是以文本为界面的媒介,最初的网络浏览器,如 Lynx 等,都是只能用于文本方式的软件。事实上,许多页面标明“Text only”,这种浏览器的用户不能依靠视觉媒介(如图像等)来浏览 HTML 文件。直到 Marc Andreessen(不久后他与 Jim Clark 合开了 Netscape 公司)在 NCSA 开发了 Mosaic,网络才能以图形方式的界面来浏览。像 HTML 那样,用简单的文本文件实现复杂的多媒体功能,不能不说是个奇迹。

目前的 HTML 文档中文字所占的比例正越来越小,文档中确实还存在各种标记和引导符(用以告诉浏览器如何显示页面),但它们更多是被用来显示图像,运行动画文件,或者调用一些嵌入程序,如 WebFX 或 Macromedia 的 Shockwave。不过话又说回来,现在的 Web 页面尽管可以变是非常复杂,但却仍然是基于文本方式 HTML 语言的,基于文本的页面是 WWW 的基础。

1.1.4 用 HTML 设计的 Web 页面

自从 Kernighan 和 Ritchie 的《C Programming Language》出版后,以“Hello, World!”命名的小程序往往用来向人们介绍新的编程语言。按照这个传统在介绍 HTML、Java 和 JavaScript 时,“Hello, World!”这个程序将会在我们这本书里被用来向读者展示这几种语言的不同点和相似之处。下面这个“Hello, World!”程序中,我们还会加一些额外的东西以显示这些语言各自的特性。

HTML 告诉用户的浏览器,比如 Netscape 或 Internet Explorer,怎样呈现文本、图像、声音及其他被浏览器和它的应用程序所支持的文件。HTML 通过标记实现这种功能。你可能已经极其熟悉用 HTML 来制作 Web 页面,因此,花大量篇幅叙述这些标记的使用就没有什么必要。只要知道这些标记被浏览器解释,而它指明了怎样显示 Web 页面,这就够了。下面的例子是一个简单显示“Hello, World!”的 HTML 文档:

```
<HTML>
<TITLE>Say Hello;</TITLE>
<BODY>
<A HREF="http://www.mcp.com/~newriders">Hello, World! </A>
<IMG ALIGN=MIDDLE SRC="images/planet.gif">
<HR>
Send me some mail: <A HREF="mailto:tdr20@cus.cam.ac.uk">
<IMG ALIGN=MIDDLE SRC="images/mail.gif" BORDER=0></A>
</BODY>
</HTML>
```

这个文件可在所附 CD 的/JavaScript/HTML/的目录下找到,名字叫作 Hello, World!.