

100%

内容丰富、权威

精通JavaScript强有力
语言以及客户端script
编写环境

将先进的交互性能增
加到Web页面——包
括Java的小应用程序
和动态HTML应用

开拓Netscape
Navigator 4和
Internet Explorer 4
中新的script编写能力

美国IDG“宝典”丛书

JavaScript Bible



[美] Danny Goodman 著

汪厚祥 董京春 杨霞 韩冰 等译

邹远志 审校

JavaScript

宝典

上百个
待运行的
script例子加上
CD-ROM中7个
额外的应用程序



电子工业出版社

Publishing House Of Electronics Industry
URL: <http://www.phei.com.cn>

内 容 简 介

本书是 JavaScript 创始者推荐的,全面介绍了 JavaScript 的各种强有力的功能,从程序设计基础开始介绍,包括多窗口和框架的表单、动态 HTML、各种对象的应用、表单元素输入的验证、鼠标滚动变换图象的界面效果、跟踪查错程序的使用以及 Netscape 和 Microsoft 两公司在 JavaScript 工具中的差异之处等等众多内容。书中的程序示例全在附带的 CD-ROM 中,读者可直接借用以实现自己的代码设计。本书并非一定是专业人员才能阅读,凡是学习和开发 JavaScript 的人员,包括爱好者均可视其为得心应手的参考书。

JavaScript Bible, 3rd Edition by Danny Gookman



Copyright © 1997 by Publishing House of Electronics Industry.

Original English language edition copyright © 1998 by IDG Books Worldwide, Inc.

All rights reserved including the right of reproduction in whole or in part in any form.

This edition published by arrangement with the original publisher, IDG Books Worldwide, Inc., Foster City, California, USA.

本书中文简体专有翻译出版权由美国 IDG Books Worldwide, Inc. 公司授予电子工业出版社及其所属今日电子杂志社。未经许可,不得以任何手段和形式复制或抄袭本书内容。该专有出版权受法律保护,侵权必究。

丛 书 名:美国 IDG“宝典”丛书

书 名:JavaScript 宝典

著 者:[美] Danny Goodman

译 者:汪厚祥 董京春 杨霞 韩冰 等译

责任编辑:嘉 益

印 刷 者:北京天竺颖华印刷厂

装 订 者:三河市金马印装有限公司

出版发行:电子工业出版社出版、发行

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68273574

URL: <http://www.phei.com.cn>

经 销:各地新华书店经销

开 本:787×1092 1/16 印张:60.75 字数:1455 千字

版 次:1999 年 4 月第一版 1999 年 4 月第一次印刷

书 号:ISBN 7 5053-4961-9
TP·2441

定 价:115.00 元(含光盘一张)

著作权合同登记号 图字:01-98-1566

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换

版权所有·翻印必究

译 者 序

在我国 Internet 和 Intranet 迅猛发展的今天,无论是计算机软件专业的程序编写者还是非专业的 Web 网页编写爱好者,对于能有这样一本完整介绍 JavaScript 的书,真是渴望已久的事。本书作为学习和开发 JavaScript 人员的一种工具或技术资料,也是当之无愧的,这是我们有幸翻译此书后,或者说全面阅读书中的内容并理解作者意图后,深深领悟到的。作为既是译者又是计算机行业的研究和开发人员的我们,由于时间和水平的限制,难免有误,一方面希望诸位同仁能够谅解书中出现的错误,另一方面,真切地希望此书能够成为你们手中非常实用的参考书。

在此书出版之际,我们还要一并感谢李新捷、邹远志、李方、张正霞研究生、王志民、胡锋路、张敏文和陈晰等同志翻译和校对了部分章节,汤可桢、高爱旗、李俊峰、孙逸州、徐晓丽、周铃、任娅、王奕之、良晶等协助做了大量的文字处理工作。

译 者

1998 年 12 月

序

作为 JavaScript 的创始者,我很愿意在这儿就 JavaScript 的过去和发展,以及本书将如何帮助你最大限度掌握 JavaScript 等问题谈几句。

JavaScript 的产生是为了让 HTML 使用者可以直接把 script 写入他们的文本中。这在现在看来也许很平常,但在 1995 年春,这是很新颖的,同传统观点(HTML 只能编写静态文本结构)以及 Next Big Thing(Java 小应用程序,它被称为一种真正可以扩展网页并使之有生气的办法)相比,有点奇怪。但我们一旦不用再争辩这些问题时,JavaScript 已在以下几方面取得了迅速进展:

◆**“Java-lite”语言体系** HyperTalk 的“自然语言”体系概念是在一个朋友借了我一本书之后才在我的头脑中形成的,这本书名字是《完备的 HyperCard 手册》,是一个叫 Goodman 的人写的。尽管如此,Next Big Thing 显得还是过于沉重了,特别是考虑到另一个目标:Java 小应用程序的脚本化。编写脚本对大多数人来说就是写一些短小的代码片段,又快又省事。

◆**HTML 元素事件** 按钮应该有 onClick 事件处理程序;文档要从窗口载入或退出,所以窗口应该有载入(onLoad)和退出(onUnload)处理程序;用户和 script 提交表单:onSubmit。尽管还没有象 HyperCard 的信息那样灵活(它的处理程序可以激活 onEvent 命名协议),JavaScript 事件令 HTML 编写者可以从远程服务器控制并迅速响应用户的操作和浏览器的行为。

◆**没有类的对象** 自编程语言通过以原型为基础的继承实现了这一概念。对于 JavaScript,希望每个对象都只有一个单独的原型(为了简单和效率),在缺省情况下,它是建立在使用 new 操作符的函数基础上(为了与 Java 一致)。为了免除区分函数和方法的构造函数,所有函数都以 this 指针指向把它们定义成被访问原型的物体。尽管原型直到 Navigator 3 后才出现,在第 2 版本中它们已通过括起来的文本作为一个对象(String 原型,用户可以把方法与之相连)预示出来。

◆**派生的 HTML (Generated HTML)** 由把 JavaScript 嵌入 HTML 引发了一个想法:让 script 代替 HTML 语言,就象用发送的文本和标注代替 script 本身一样。当来自 JavaScript 对象(这些对象组成一个目录或微型数据库)需用表单表示的变化内容以最小方式出现时,要计算所有表的树(trees of tables),表内所有重复结构出现在一个 script 的循环内,这种可能性超过了自动化趋势或最后修改的日子。

起初,我认为 JavaScript 将主要应用于对 HTML 表单的合法输入验证。但我惊奇地发现了许多派生出的 HTML 和 JavaScript 对象更有力的应用,例如 Bill Dortch 的 <http://hi-daho.com> 页面。从用户的要求和反馈可以清楚地看到用一些图象、HTML 和 JavaScript 可以迅速而有效地完成一些重要应用。最重要的优点是具有相应的补充办法,以满足编写者的需要。关键是要让一些易用的语言发挥它们的能力,而不需要专门的编程技术。

今天 Web 上的 JavaScript 的重要性证实了我们最初对适用于 HTML 编者的 script 语言

价值的信心。通过放低对“pixel-lighting”栏的设计要求,带图象的 HTML 使得成百万的人成为 Web 设计者。同样,通过对“事件处理”栏设计要求的降低,JavaScript 使得许多人成为程序员。

JavaScript 已经成为独立于 HTML 的一门综合性的有用语言。它已被纳入服务器、编辑工具和其它的浏览器中(如三维图象世界的浏览器)。现在它是一项国际标准 ECMA - 262。但和一些语言如 Perl 甚至 Java 相比,它仍然太年青。其中的缺陷有些是值得一提的,以便在今后的版本中予以克服。

◆**缺少十进制计算** JavaScript 使用 IEEE 的双精度浮点标准,但对许多“美元和分”的计算来说,十进制结果没有二进制表达的精确: $70.46 - 39.96$ 得 30.4999999999993 。这是令多数用户望之却步的原因之一;它应该可以用一种可选择的十进制数字模式来表达。

◆**事件的不灵活性** 用户可能喜欢写多个处理程序,用于一个被指向给定对象或对象系统的事件。此处还要多说的是,JavaScript 1.2 在这方面有所改进,就像 HyperCard 爱好者能够看到的一样。

◆**缺少异常和模块** 当我们积累和重用成套的函数和 script 时,这些“功能强大的编程”特点对我们来说就是必须的了。我希望能早点看到这些方面出现周到考虑的扩展,并被 ECMA 技术委员会标准化。

我很清楚,如果没有一个具有创造性、忠诚而耐心的开发小组,JavaScript 是不能得以发展的,我向他们每个人致以深切的谢意。这些成员不仅促成了 Beta 版的 Navigator 2,而且通过电子邮件和网上新闻传播,他们成为这门语言的创造者。开发人员不断地支持与反馈,使 JavaScript 在现有基础上取得了成功。

你手上的这本书是专家的指导。Danny 还不知道从他的 HyperCard Book 中我受到多么大的激励,但在 1995 年开发 JavaScript 的全过程中始终在我的书桌上。它的活力、动情和清晰的叙述帮助我坚持追求一种全能语言的目标。这本书已经赢得了这么多“满意的读者群”,为这本书的第三版写序言是一种极大的荣幸。

对于任何想学习 JavaScript,尤其是迄今为止只写了几个脚本或程序的 HTML 编写者来说,我极力推荐 Danny Goodman 的这本《JavaScript 宝典》。有一个可靠的向导在你身边,在“脚本道路”上你可能会得到终身的乐趣。

Brendan Eich
Netscape 公司

前言

在过去的十五年里,我编写了一些书,这些书本是我希望其他人编写以便帮助我学习或使用一项新技术的。我喜欢在最初开发阶段就关注新的事物,感受其中的苦与乐,并与读者一起分享我奋斗的成果。《JavaScript 宝典》的第三版代表了我差不多两年工作于 JavaScript 的成果,以及持续关注各层次 script 编写者所面临的新问题、新困难和新挑战的成果。我的目的就是要使你避免经历与我和其他一些人在用脚本编写浏览器的三代发展期间,所经历过的相同的挫折与困惑。

因为本书是关于 JavaScript 的,不是 Microsoft 的 JScript 语言版本,书中的引用结构来自 Netscape 的语言工具。在许多地方,我都指出了 Netscape 与 Microsoft 工具间的区别,你不会看到只针对 Microsoft 的语言与对象的问题。你在本书学到的有关 JavaScript 的知识大部分完全可以在 Internet Explorer 4 上运用。

本版本的组织与特点

如果你看过本书前两版,你会发现本版是全新的结构。为了要涵盖大量 Netscape Navigator 中 JavaScript 1.2 的 script 性能所必须的新内容,为了使学习语言更容易,并把本书作为今后的参考资料,本版本重新考虑了内容的结构。主要变动包括:独立的指导部分,内容更广泛的参考部分,这些部分提供了专门的对象章节,而这些章节是围绕着语言体系按字母顺序排列的处理办法而组织起来的。

以下是关于本书这种新结构的一些细节。

第 1 部分

第 1 部分的开始一章介绍了 JavaScript 与 Java 的对比并讨论了它在 World Wide Web 中的作用。自从 JavaScript 推出后,Web 浏览器和 script 领域经历了重大的变革。这就是为什么第 2 章要着力介绍 script 编写面临挑战的原因,这些 script 编写者必须为迅速变化的标准而努力,开发出既适于单平台浏览器读者又适于交叉平台浏览器读者的应用程序。第 3 章对 JavaScript 先进行了介绍,在这儿你可以开始着手编写你的第一个 script。

第 2 部分

第 2 部分所有内容都是为了引导 JavaScript 者完成一次旅行,共九章内容,循

序渐进地为你讲述浏览器的本质、基本编程技巧和真正的 JavaScript 的脚本编写。每课后面附有练习,帮助你巩固所学知识,激励你使用所学的新知识(你可在在附录 C 中找到练习答案)。这次旅行的目的就是以足够的经验来武装你,让你可以马上开始用 script 编写简单的主页,同时令你更易于理解本书其余部分的深入讨论和范例。当最后一课结束后,你会知道如何编写多框架环境的 script,创造出靠鼠标滚动翻转的图象和转换效果的图象,这在今天很多的 Web 页上是很流行的。

第 3 部分

第 3 部分,也是本书最大的一部分,深入探讨了这一语言。多数章节都把重点放在文档内的特殊对象上,但你会发现本书完全包括了其他语言特性,如控件结构、函数和这门语言的任何其他部分。由于大量语言和浏览器的版本都包含 JavaScript,所以每种语言项都包括一个相应的章节,这样你可以看看哪些浏览器支持这门语言。第 14 章到第 36 章的特色就是指导主页黑栏内的词,这可以帮助你容易地寻找有关 JavaScript 术语方面的信息。

第 4 部分

第 4 部分,开始用 JavaScript 处理业务。这部分是 JavaScript 的应用方面内容,如第 37 章是客户端表单数据有效性,第 38 章是把 Java 小应用程序溶入主页中。第 39 章深入探讨了 Navigator 4 中新的事件模型。而第 40 章则详细介绍了支持应用程序的安全性问题。动态 HTML 占了恰好三个章节,向你展示 DHTML 应用程序如何针对 Netscape Navigator 4、Internet Explorer 4 以及两个浏览器同时都可用。尽管本书的观点是根源于 Netscape 的 JavaScript 工具,你还会了解到与用 Microsoft 实现的差别,这正是第 44 章的内容。调试 script 是第 45 章的重点,包括关于理解错误信息的警告,建立你自己的调试工具和使用 Netscape 的调试器。

第 5 部分

最后,本书后面的附录提供了具有帮助价值的参考信息,这些资料包括:附录 A 的 Netscape Navigator 对象的路径图,附录 B 中的 JavaScript 保留字列表,附录 C 中的第 2 部分练习的答案,以及附录 D 中的 Internet 资源。在附录 E 中,你会找到关于如何使用随书附带光盘的介绍。

CD-ROM 光盘

附带光盘内容包括完整的 HTML 文件,这些文件就是在第 3 部分中多数 JavaScript 关键字例子的那些文件。你可以用支持 JavaScript 的浏览器运行这些例子。我本可以给你一些脱离上下文的幽默片段,但我认为应用这些概念查看一些

富于技巧的 HTML 文件(它们也许很简单)是很重要的。我有意略去了本书第 2 部分中的 script 列表,就是要鼓励你自己去写出 script。我相信随着你习惯于连续地把 script 写入文件,即使通过模仿书中的例程,你也会学到很多。你会发现光盘上也有许多第 1 部分和第 4 部分的 script。

光盘上还有七个富于技巧的、增强 JavaScript 的应用程序。这些例子多数也运行于我的 Web 地址上(<http://www.dannyg.com/javascript/>),所以你既可以在光盘上使用它们,也可以从服务器上看看它们在线工作的情况。每个应用程序都以实例说明了重要的概念,而这些概念是你在自己的程序中可能会包括的。你会读到 JavaScript 代码的每个片段,学到我在设计这些程序时作出的执行决定(这是你在你的网址上同样需要作的决定)。

附录 A 中的对象路径图是以 .pdf 格式存在光盘上的,如果需要的话,你可以把它打印出来,作为一本便携式参考手册。Adobe Acrobat Reader 也附在光盘中,你可以用它来读这个 .pdf 文件。最后,为了便于查找本书的内容,我们也用 .pdf 文件形式把本书的文本存到了光盘上。

学习 JavaScript 的前提条件

尽管本书并不要求你具备丰富的编程经验,你用 HTML 创建的主页越多,就会越容易理解 JavaScript 是如何与你通常放在主页上那些熟悉元素发生关联的。偶尔,你会需要调整 HTML 标记符以充分利用 JavaScript。如果你已经熟悉这些标记符,就会比较容易消化 JavaScript 的增强性。

表单及其元素(文本区域、按钮和选择表)在多数典型 JavaScript 作品中扮演着特别重要的角色,你必须熟悉这些元素和它们的 HTML 属性。幸运的是,你不必了解 CGI script 或从表单到服务器的信息传输。这里的重点是客户端 script 的编写,这在增强 JavaScript 的 HTML 主页完全载入浏览器后,就可以独立于服务器进行操作了。

目前的 HTML 标准和 Netscape 扩展知识是你必须具备的工作知识。例如当我们开始使用图象时,重点就会落在编写这些元素的 script 上,而不是用它们设计主页。

如果你以前从没编过程序

对于那些只由薄薄的一本指导手册学过 HTML 的人来说,这本书的规模之大一定是令人气馁的。JavaScript 也许不是世界上最好学的语言,但相信我,它绝不需要学习一门完整的编程语言,如 Java 或 C。不同于开发一个富于技巧的独立应用程序(如你在商店买到的多数程序),JavaScript 让你编写小应用程序代码片段完成大事情。建立在所有可用脚本编程的浏览器内的 JavaScript 解释器可以为你做很多的技术工作。

从最基本层次上讲,编程不过就是写一系列的指令让计算机执行。我们人类

就是执行指令,即使我们并没有意识到这一点,去一个朋友家就是一系列小指令:沿那条路走三个街区;从这儿向左拐;在那儿向右拐。伴随这些指令我们要做出一些决定:如果交通灯是红色的,就要停下;如果是绿色的,就通行;如果是黄的,就把汽车的加速器压到底;偶尔地,我们必须重复某些动作几次(如必须绕一个街区转圈,直到有车位空出)。一个计算机程序不仅包括步骤的主要过程,它还要预测需要决定或重复来完成目标(如怎样处理不同的交通灯状态,或者如果有人抢占了你要停的车位怎么办)。

学习编程的最初障碍是要按照编程语言确定的词语的方法和它们的指令组织数字。这些规则叫做语法体系,和生活中的语言一样。由于计算机通常是沉默的电子物体,如果你不用它们能理解的特定语言与之交流,它们是不会十分宽容的。当和另一个人谈话时,你可以把句子的语法搞的一团糟,你还有机会让其他人理解你的完整意思。但对计算机编程语言并不适用,如果语法不是完全正确(或者至少在该语言范畴内它还可以改正),计算机就会告诉你:你已经犯了一个语言错误。

你能做的最好的事就是面对那些在学习过程中遇到的语法错误。即使经验丰富的程序员也会遇到这些错误。你遇到的每次语法错误,你由于犯错误而重写的每一种解决办法,都可以增加你的语言知识。

如果你从前只写过一些程序

在学习 JavaScript 时,用某种过程化语言如 BASIC 或 Pascal 编程的经验几乎会成为一种障碍,而不是帮助。尽管你可能偏爱语法的准确性,关于程序如何适应世界的主要概念可能是极不同于 JavaScript 的。一部分是要用 script 完成典型的任务(执行特定的任务,响应用户在 Web 主页上的操作),但大部分也是处理面向对象编程的本质。

在典型的过程化语言中,程序员要负责所有出现在显示屏上的东西以及所有发生在遮光板下的事情。当程序一开始运行,许多代码特地用来安装启动可视化环境。也许在显示屏上会包括几个文本输入区的可单击按钮。要用户单击哪个按钮,程序要检测单击处的坐标,并把它们和屏上所有按钮的坐标相比较,然后程序执行分出用于执行单击那一区域所对应的指令。

面向对象编程几乎恰好是这一过程的反过程。一个按钮被当作一个对象,即某种有形、可触及的东西。一个对象具有属性,如它的标签、大小、对齐方式等等。一个对象也可能包含一个 script。同时,系统软件和浏览器也可能一起向对象发出消息以激活 script,这取决于用户做什么。例如,如果用户单击了文本输入域,系统/浏览器就告诉这个域有人在这儿单击了(这个域已被设置),让它决定对此要做什么,那里就是 script 入手的地方。script 被连到这个域,包括当用户激活它时它要执行的指令。另一套指令可能控制用户输入的词条和记录,或在域外的单击,从而改变域内内容。

你要编写的一些 script 可能看起来在结构上是过程化的:它们包含一串简单的依次执行的指令。但当处理来自表单元素的数据时,这些指令按 JavaScript 本

身的基于对象的属性工作。表单是一个对象,每个单选按钮或文本域也是对象,script 于是就依据这些对象的属性执行操作完成某项工作。

对你来说,从过程化编程到面向对象编程的转变,也许是最困难的事。当我多年前第一次接触面向对象编程时,我也不能马上掌握它。但经过长时间深思熟虑的探索后,这一概念在头脑中激起许多灵感时,我认为我可能要走出困境了。从那时起,面向对象看起来成了唯一切合实际的编程途径,如果你曾用 C 语言编过程序的话。

如果你曾经用 C 编过程序

由于借用 Java 的语法(而 Java 又是由 C 和 C++ 演生而来的),所以 JavaScript 和 C 有许多类似的语法特点,熟悉 C 语言的程序员会感到得心应手的。操作符、条件结构和循环都沿袭了 C 语言的内容。在 JavaScript 中使用象在 C 中的数据类型。在 JavaScript 中,一个变量不严格限制于任何特定数据类型。

由于有这么多 JavaScript 语法是你所熟悉的,所以你可以专注于 JavaScript 独一无二的属性上,主要的是对象系统。不过,你还需要有足够的 HTML 知识(特别是表元素)来学习 JavaScript。

如果你曾经用 Java 编过程序

如果你曾用 Java 编过程序,尽管名字很相似,但这两种语言却只有表面上的共同点:循环和条件结构、象 C 一样的“圆点”对象的引用、一组语句两端的大花括号、一些关键字和几个其他属性。然而,变量说明是完全不同的,因为 JavaScript 是一门类型限制宽松的语言。一个变量可以在一个语句中含整型值,而在下一个语句中含字符串(尽管我不认为这是一种风格)。Java 调用方法(与预定义的对象有关时)和函数(用于用户定义的操作)可以返回任意类型的值,而不用事先声明数据类型。

也许在编写 JavaScript 时,影响 Java 的最重要因素就是类、继承、实例和消息传送(message passing)等面向对象的想法。在编写 script 时这些方面不构成问题。然而,这门语言的设计者也考虑到你可能有一些很难改掉的习惯。例如,JavaScript 尽管不需要在每一语句后都有分号,但如果你在 JavaScript 源代码中写了分号,JavaScript 解释器也不会排斥。

如果你曾写过 script(或宏)

如果你用其他编辑工具写过 script 或者写过程序中的宏,这些经验会有助于你掌握 JavaScript。概念也许最重要的,就是把一段特定任务的指令行语句与某些数据结合起来。例如,你可以用 Microsoft Excel 写一个宏,让它处理来自另一台计算机的公司财务报告上的日常数字的热气转换。这个宏被放入宏菜单,无论什么

时候传来一组新的数字,你都可以通过选择菜单选项来运行它。

很多完善的 script 技术,如在 Toolbook 或 HyperCard 中看到的,与 JavaScript 中面向对象的内容很相似。在那些环境下,屏幕对象包括用户与那些对象交互作用的 script。你在 JavaScript 中要大量做的脚本编写工作与该模式完全匹配。事实上,那些环境就像另一种方法的 JavaScript:它们提供一组属性和行为确定的预定对象。这种可预测性使得学习整个环境和规划一个应用程序变得更易于完成了。

格式及命名约定

在本书中,由于篇幅限制,script 程序行常常会非正常中断。这种情况下,script 的剩余部分会在下一行出现,与程序的左侧平齐,就象在有的文本编辑器中一样。如果在把 script 程序输入到你的文件中时,这些行中断会带给你麻烦,我建议你调出光盘上的相关列表,看看你应该怎么输入它。

这本书的编写与 Netscape 和 Microsoft 的第四版浏览器的发布是一致的,于是我把这些产品统视为第四代浏览器。至于 Netscape 的特殊产品,公司称之为适于客户端应用的 Communicator,这种浏览器和 Netscape 的产品统称为 Navigator。当指最新版本时,我称之为 Navigator 4,实际上,它可能是你所安装的 Communicator 软件的一部分。有时我简单地用开头字母和版本号如 IE3 来指代 Microsoft 公司的 Internet Explorer 3, IE4 指 Internet Explorer 4。除非我特别指出其一个品牌浏览器的次级版本(如,4.03 版)。



“注意”和“当心”图标偶尔会出现在书中,指出重要问题。

目 录

前言	(1)
第 1 部分 JavaScript 初步	(1)
第 1 章 JavaScript 在万维网上的地位	(3)
网上竞争	(3)
标准的沿革	(4)
CGI 的 script 过程	(4)
关于辅助程序、插件和小应用程序	(5)
插件	(5)
Java 的小应用程序	(5)
JavaScript: 语言的集大成者	(6)
LiveScript 变成了 JavaScript	(6)
走到一起来	(7)
微软世界	(7)
JavaScript: 一种活学活用的好工具	(8)
第 2 章 浏览器竞争中写作工具的挑战	(9)
相互竞争	(9)
相互趋同	(10)
今天的兼容性问题	(11)
区分诸对象的不同语言	(11)
核心语言标准	(11)
阶梯排列的风格页	(13)
动态 HTML	(13)
script 编写策略	(14)
第 3 章 JavaScript 初步	(15)
软件工具	(15)
选择文本编辑器	(15)
选择浏览器	(16)

设置你的写作环境	(16)
窗口	(17)
MacOS	(17)
script 首次要做什么	(18)
输入初始的 script	(19)
检查 script	(20)
<SCRIPT> 标记符	(20)
一个 script 对应所有浏览器	(21)
文本显示	(22)
享受乐趣	(23)
第 2 部分 JavaScript 指南	(25)
第 4 章 浏览器和文档对象	(27)
script 运行演示	(27)
JavaScript 在行动	(28)
交互的用户界面	(28)
少量数据的查找	(29)
表单确认	(30)
交互数据	(30)
多重框架	(31)
动态 HTML	(32)
什么时候用 JavaScript	(32)
文档对象模型	(34)
严格的分级结构	(35)
当文档下载时	(35)
一个简单的文档	(36)
添加一个表单	(36)
添加一个文本输入单元	(37)
添加一个按钮元素	(37)
对象引用	(38)
对象命名	(38)
一个简单的文档	(39)
添加一个表单	(39)
添加一个文本输入元素	(40)
添加一个按钮元素	(40)
关于点语法	(40)
新闻组的组织模型	(41)
对象定义的是什么	(42)
属性	(42)

方法	(43)
事件处理程序	(44)
练习	(45)
第 5 章 script 和 HTML	(47)
script 放在文档哪儿	(47)
<SCRIPT>标记符	(47)
标记符位置	(48)
处理老版本的浏览器	(50)
JavaScript 语句	(51)
script 语句什么时候执行	(51)
当文档下载时立刻执行	(51)
延时 script	(52)
script 与编程	(54)
练习	(55)
第 6 章 程序设计基础之一	(57)
处理信息	(57)
变量	(58)
创建变量	(58)
变量名	(59)
表达式和求值	(59)
script1.htm 中的表达式	(60)
表达式与变量	(60)
数据类型转换	(62)
转换字符串为数值	(63)
转换数字到字符串	(64)
运算符	(64)
算术操作符	(64)
比较操作符	(65)
练习	(65)
第 7 章 程序设计基础之二	(67)
决策与循环	(67)
控制结构	(68)
if 结构	(68)
if...else 结构	(69)
循环结构	(69)
函数	(70)

函数参数	(71)
变量值域	(72)
花括弧	(74)
数组	(74)
建立数组	(75)
存取数组数据	(76)
关联数组	(76)
数组中的文档对象	(78)
练习	(78)
 第 8 章 窗口及文档对象	(81)
文档对象	(81)
窗口对象	(81)
访问窗口属性及方法	(82)
建立窗口	(83)
窗口属性及方法	(84)
window.status 属性	(85)
window.alert() 方法	(85)
window.confirm() 方法	(86)
window.prompt () 方法	(86)
onLoad = 事件处理程序	(87)
位置对象	(87)
历史对象	(88)
文档对象	(88)
document.forms[] 属性	(88)
document.title 属性	(89)
document.write() 方法	(89)
连接对象	(91)
练习	(92)
 第 9 章 表单与表单元素	(93)
表单对象	(93)
表单对象和容器	(93)
建立一个表单	(94)
form element[] 属性	(94)
文本对象	(95)
按钮对象	(96)
复选框对象	(96)
单选钮对象	(97)

选择对象	(99)
向函数传递表单数据和元素	(100)
提交表单	(102)
练习	(104)
第 10 章 字符串、数学和日期	(105)
核心语言对象	(105)
字符串对象	(105)
连接字符串	(106)
字符串方法	(107)
改变字符串大小写	(107)
字符串搜索	(108)
获取字符和子字符串	(108)
数学(Math)对象	(109)
日期对象	(110)
日期计算	(112)
练习	(113)
第 11 章 用脚本编写框架和多窗口	(115)
框架:父和子	(115)
引用家族成员	(117)
父到子引用	(117)
子到父引用	(117)
子到子引用	(118)
框架的脚本编程提示	(118)
控制多框架——Navigator 栏	(119)
窗口引用的其他内容	(121)
练习	(122)
第 12 章 图象和动态 HTML	(125)
image(图象) 对象	(125)
能互相转换的图象	(126)
预缓存图象	(126)
创建图象的 rollover	(128)
HTML 中其他的动态内容	(131)
练习	(131)
第 3 部分 JavaScript 对象和语言引用	(133)

第 13 章 JavaScript 基础	(135)
语言 and 文档对象	(135)
核心语言标准——ECMAScript	(135)
文档对象标准	(136)
对象层组织	(136)
层次路线图	(137)
JavaScript 文档对象路径图	(138)
创建 JavaScript 对象	(139)
对象属性	(139)
对象的方法	(140)
对象的事件处理程序	(141)
作为方法的事件处理程序	(141)
作为属性的事件处理程序	(142)
在文档中嵌入 script	(143)
<SCRIPT> 标记符	(143)
JavaScript 与 JScript 和 VBScript	(144)
隐藏以前浏览器中的 script 语句	(144)
如何完全隐藏 script	(145)
script 库	(145)
兼容性问题	(146)
JavaScript 实体	(147)
浏览器版本判定	(147)
JavaScript 在运行吗	(148)
使用<NOSCRIPT>标记符	(148)
其他不能用 script 编写的浏览器	(149)
不同浏览器的脚本编写	(150)
建立目标	(150)
完成分支索引页	(151)
一个链接——交替目的	(152)
多层 script	(153)
编译事件处理程序的属性	(154)
兼容性设计	(155)
对待测试版的浏览器	(155)
参考章节的兼容性等级	(156)
本书的对象属性	(157)
 第 14 章 窗口对象	 (159)
窗口术语	(159)
框架	(160)