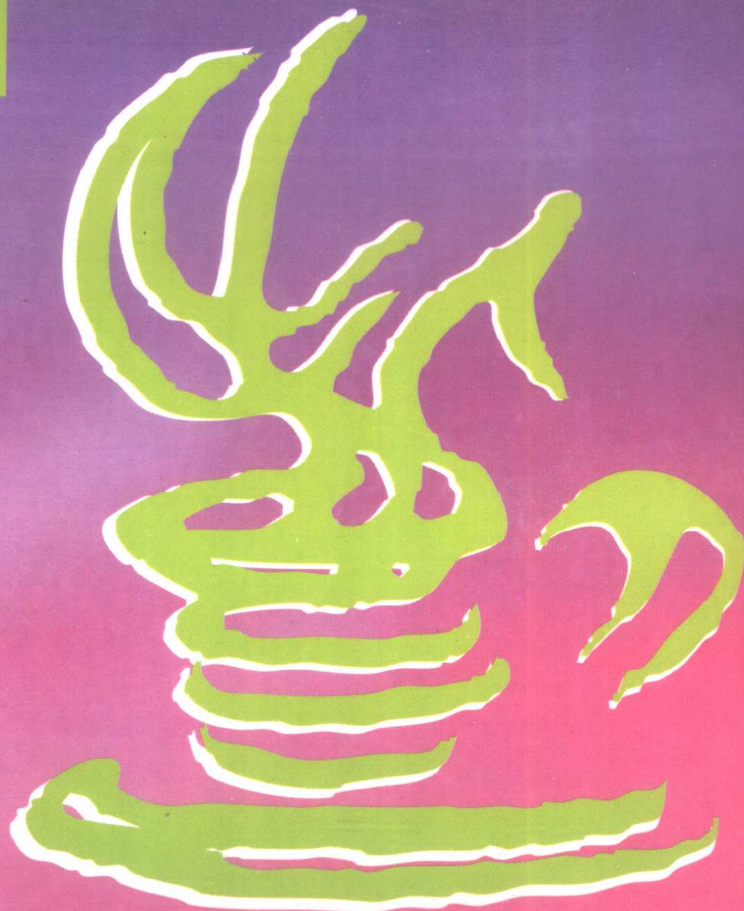


金地时代 π



JavaScript 程序设计入门 学得快

岡 超 编著

科学出版社

JavaScript 程序设计入门学得快

田 超 编著

科 学 出 版 社

2000

内 容 简 介

JavaScript 是现在 WWW 网中常用的一种脚本语言。本书共分为 10 章, 分别介绍了从 JavaScript 的基本概念和语法特点, 讲解了在 JavaScript 中使用对象概念的一般方法, 针对 JavaScript 的内置对象和 NetScape 中对象的特点和使用方法, 对它们进行了逐个的剖析。本书通过清晰、简明的讲解和简单、实用的例子, 使读者快速地建立起有关 JavaScript 程序设计的概述, 掌握在 Web 中加入动态交互元素的技巧, 具备使用 JavaScript 开发实用网页的能力。

本书面向初、中级的网页设计者和希望学习网页设计的初、中级的程序设计者。

图书在版编目(CIP)数据

JavaScript 程序设计入门学得快/田超编著. -北京: 科学出版社, 2000
ISBN 7-03-007580-3

I. J… II. 田… III. Java 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 41778 号

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

北京双青印刷厂印刷

科学出版社总发行 各地新华书店经销

*

2000 年 1 月第 一 版 开本 787×1092 1/16

2000 年 1 月第一次印刷 印张: 12 3/4

印数: 1—5 100 字数: 277 000

定价: 19.50 元

(如有印装质量问题, 我社负责调换〈环伟〉)

丛 书 序

本套丛书由《网络漫游学得快》、《Photoshop 5.0 中文版实例入门学得快》、《电脑办公实务学得快》、《JavaScript 程序设计入门学得快》、《Delphi 4.x 程序设计入门学得快》、《常用工具软件学得快》六种图书组成,涉及知识面广,内容丰富,且在选题设计上注意兼顾不同读者需求,每种书各有侧重,既有面向初、中级读者的常用工具软件的介绍,又有面向高级读者及编程人员的对理论知识、高级编程技巧的讲解。

针对很多读者反映电脑图书错误较多的情况,本套丛书对于技术问题严格控制,书中所涉及的技术环节不仅经过作者的反复验证,丛书编辑还再次进行核实,具有极高的正确性,而且对于重要技术问题的审查尤其严格。例如所有的程序都经过了丛书编辑的逐个运行验证,因此程序错误的概率几近为零,我们认为这是对读者权益的最大保障,也是本套丛书的最大优势。

在写作方式上,注意吸取前人经验教训和针对不同读者群的学习特点,推陈出新,突破计算机类图书长期以来所形成的实用性差的弊端,从实际操作事例入手,各章各节在讲解操作流程、说明理论和概念时,均配以凝聚作者多年实践经验、实用性极强的例子,且语言流畅,通俗易懂,使读者能够根据书中所述,轻松完成各项操作,快速掌握各项功能,不断获得成功的乐趣。这是本套丛书的又一特点。

在版式编排上力求美观大方、鲜明生动,在方便阅读的同时给读者带来美的享受。这一切无一不是编著者精心制作的成果。

“准确、实用、美观,满足各层次读者需求”,是本套丛书的编写宗旨。我们虽倾心而作,但因水平有限,不足和错误之处在所难免,恳请读者不吝赐教和指正,我们定会全力改进,用更多、更好的图书来回报您的厚爱。

丛书编委会

1999年8月

丛书编委会

主编: 张彬

编委: 袁方 夏梦 彭雨 钱君 于戈子 张墨珍

前 言

JavaScript 是一种适合于在 WWW 中使用的简单的编程语言。它是由 Netscape 公司在 1995 年的 Netscape 2.0 中首次推出,并且在随后的几年中被大多数的浏览器所支持,这种语言提供了方便地实现静态网页动态化的方法。

就目前使用最广泛的两种浏览器来说, Netscape 是 JavaScript 的开发者,而 Microsoft Internet Explorer 3.0 之后的版本都支持 JavaScript 和 VBScript 这两种脚本语言,因此我们无需怀疑 JavaScript 的通用性。

最初 JavaScript 只被认为是一种客户端的编程语言,但是现在它已经可以完成大多数服务器端编程的任务了。

从最初投入市场到今天,经过几次升级,目前的 JavaScript 已经是一个比较成熟的产品,其版本号是 1.2 版。这个版本与以前的版本相比,更加注意了网络安全性等因素,同时增强了服务器端编程的能力。

尽管 Java 语言也是一种通用的网络编程语言,但是与 JavaScript 相比,它则显得过于复杂了。

一、本书的目标

本书的目标在于简洁而生动地介绍 JavaScript 在客户端的应用。对于不具有大量计算机专业知识,而且希望逃离复杂痛苦的网络编程的有识之士来说,通过本书来学习 JavaScript 与 HTML 共同开发网页的技术,是开发出有吸引力网页的捷径。

通过阅读本书,读者将能够独立读懂在现今网页中出现的,或是一些网站提供的 JavaScript 资源中的绝大多数 JavaScript 源程序。我们希望读者不是停留在本书介绍的内容上,而是通过继续学习这些已经在使用中的源程序,掌握使用 JavaScript 的更多技巧,从而真正成为一个 JavaScript 的编程高手。

本书之所以不介绍服务器端的应用,有三个原因:

一是服务器端的编程相对复杂,而且调试比较困难。

二是由于服务器端的编程需要一定的网络环境(比如说自己的服务器),这对于一般的用户可能比较困难。

三是客户端的编程是 JavaScript 最主要的功能,也是目前为止应用最广泛的范畴。

通常认为 JavaScript 是一种能成功实现客户端动态化的语言,因此是首选的客户端编程语言。但是这种语言在服务器端编程方面却存在相当多的竞争对手,这使得它在服务器端的发展显得不是那么令人满意。

二、本书的结构

本书分为三大部分:

第一部分包括前三章,主要介绍 JavaScript 的相关背景和语法,这些知识是学习后面章节的基础。

第二部分包括第四章到第七章的内容，主要介绍怎样在 JavaScript 中使用对象和一些 JavaScript 内定义对象的使用。这是本书的主体，可以说学会了 JavaScript 中的对象的使用，就是学会了使用 JavaScript。

第三部分包括第八章到第十章的内容，主要介绍在网页中经常使用的 JavaScript 的一些相关内容，进一步讲解怎样使 JavaScript 与 HTML 融合在一起。这一部分是第二部分的加深。另外，这一部分还简要介绍一些经常被提及的高级技术。

三、本书的读者对象

本书不是探索 JavaScript 奥秘和高级使用技巧的专著，它是可以使读者快速入门的速成教材。它面对希望提高网页交互性和动态性的网页开发者，同时适合于对 Web 开发怀有兴趣的任何读者。

本书要求读者有基本的使用 HTML 的能力，这样在学习本书时，将没有必要查阅相关的书籍。但是如果读者恰好对 HTML 一无所知，那也并不要紧，我们会在新概念出现时加以一定的解释。不过，这种情况下最好的方法或许是寻找一本有关 HTML 的参考书放在手边，这样在学习时基本不会有问题了。

如果读者有一定的 C 语言或者其它结构化语言的编程基础，将会有助于理解 JavaScript 的相关概念，如果没有也不会影响本书的学习。

四、软件与硬件需要

本书中的例程除了极个别是在 Microsoft Internet Explorer 上运行的，其余的全部在 Netscape 4.0 上运行通过。尽管大多数的程序也可以在 Netscape 3.0 以前的版本上运行，但是并不保证一定可以得到预期的效果。因此，希望读者使用 Netscape 4.0。

读者使用的操作系统应是 Windows 3.2, Windows 95 或 Windows 98, 这是因为 Netscape 4.0 是一个 32 位的浏览器，如果您仍在使用 Windows 3.1，可能不得不升级您的操作系统，或者选择 16 位的 Netscape。请注意只有 Netscape 2.0 以上的版本才支持 JavaScript。

使用本书时，并不是一定需要多个浏览器来运行程序，但如果条件允许的话，在多个浏览器中试验同一例程，会得到意想不到的收获。如果真正实用地制作网页，就应该尽可能在不同的计算机、不同的浏览器上试验程序，这些计算机和浏览器对同一个 JavaScript 程序的运行结果可能会有很大差异。

读者可以利用任何的文本编辑器来编辑 JavaScript 的源程序，不需要这方面的应用软件，这个问题在后面我们还会提及。

不用多说，读者还需要一台计算机。需要利用它来运行一些小程序，验证在书中学习到的内容。就 JavaScript 本身的需要来说，这台计算机的配置不需要太高。任何可以运行 Netscape 的计算机都可以满足要求。但是如果希望运行 Netscape 4.0，而且使用 Windows 95 或者 Windows 98 操作系统，那么相信读者自己也会认为至少需要一台一般配置的 Pentium 机，这样才可以保证速度，等候的时候不会那么令人痛苦。

五、一点建议

在正式开始学习本书之前，谈一下本书的使用方法。

我们推荐这样使用本书：打开计算机，在读书的同时如果有任何想法，就试验一下；

遇到书中的例程，都输入执行，并可以做适当的修改，看一下结果有何不同。这种边学边试的方法对于本书来说非常合适。

至于其他的工作，只有读者自己努力了。

作 者

1999 年 8 月

目 录

第 1 章 JavaScript 简介	1
1.1 从 Internet 到 WWW.....	1
1.1.1 Internet 的起源和现状.....	1
1.1.2 什么是 WWW	2
1.2 HTML 和 WWW	3
1.2.1 什么是 HTML.....	3
1.2.2 HTML 的功能.....	5
1.3 从 HTML 到 JavaScript.....	6
1.3.1 什么是 JavaScript.....	6
1.3.2 JavaScript 的历史.....	7
1.3.3 为什么选择 JavaScript.....	8
1.3.4 Java 与 JavaScript.....	9
1.3.5 服务器端的 JavaScript.....	9
1.4 JavaScript 程序的编辑与调试.....	10
1.5 小结	11
第 2 章 在 HTML 中嵌入 JavaScript.....	12
2.1 怎样嵌入 JavaScript.....	12
2.1.1 使用<SCRIPT>标识.....	12
2.1.2 <SCRIPT>标识的属性值.....	13
2.1.3 由 SRC 属性带来的困惑.....	14
2.2 处理老式浏览器	15
2.2.1 掩盖 JavaScript 程序代码.....	15
2.2.2 使用<NOSCRIPT>标识.....	16
2.3 第一个例程	17
2.3.1 最简单的编辑方式.....	17
2.3.2 用 Netscape 完成相同的工作.....	19
2.4 小结	20
第 3 章 JavaScript 的语法结构.....	21
3.1 变量和数据类型	21
3.1.1 变量.....	21
3.1.2 数据类型.....	23
3.1.3 其他特殊符号	25
3.2 表达式与操作符	25

3.2.1	逻辑运算符和关系运算符	26
3.2.2	算术运算符	27
3.2.3	位运算符	28
3.2.4	字符串运算符	28
3.3	控制流和函数	29
3.3.1	循环语句	29
3.3.2	条件转移语句	30
3.3.3	continue 和 break 语句	31
3.3.4	函数的使用	31
3.3.5	有关程序风格	34
3.4	使用事件触发	34
3.4.1	事件触发的概念	34
3.4.2	在 JavaScript 使用事件触发	35
3.5	举例: 简易计算器	37
3.6	小结	41
第 4 章	用户自定义对象	42
4.1	对象和实例	42
4.2	在 JavaScript 中使用对象	43
4.2.1	构造对象的属性	43
4.2.2	为已有对象增加属性	44
4.2.3	为对象加入方法	45
4.2.4	对象的嵌套定义	47
4.2.5	使用 with 引用对象属性	47
4.2.6	另一种循环语句	48
4.2.7	再谈访问对象属性	49
4.2.8	一般对象	50
4.3	数组对象	50
4.3.1	预定义数组对象	51
4.3.2	自己定义数组对象	54
4.4	举例: 汽车对象扩展	54
4.4.1	扩展汽车对象的属性	55
4.4.2	扩展汽车对象的方法	55
4.4.3	完成 HTML	58
4.5	小结	61
第 5 章	内置对象和函数	62
5.1	String 对象	62
5.1.1	生成字符串对象	62
5.1.2	字符串对象的属性	63

5.1.3	查找字符和子串	66
5.1.4	改变字符串的大小写	68
5.1.5	形成 HTML 文本格式	68
5.2	使用 Math 对象	72
5.2.1	Math 对象的属性	72
5.2.2	Math 对象的成员方法	73
5.2.3	Math 对象与字符串、数组对象的比较	75
5.3	使用 Date 对象	75
5.3.1	创建日期对象	76
5.3.2	get 方法组	77
5.3.3	set 方法组	78
5.3.4	与字符串有关的成员方法	78
5.3.5	有关日期到数值的成员方法	79
5.4	其他内置对象	80
5.4.1	Number(数值)对象	80
5.4.2	Boolean(布尔值)对象	80
5.4.3	Function(函数)对象	81
5.5	预定义函数	81
5.6	举例: 记事本	83
5.6.1	构造对象结构	84
5.6.2	“查找”函数的定义	84
5.6.3	记事本中的其他函数	85
5.6.4	各功能合并	85
5.7	小结	89
第 6 章	Navigator 对象树	90
6.1	Navigator 对象层次	90
6.1.1	Navigator 各对象简介	90
6.1.2	Navigator 对象树的图解模型	91
6.2	Document 对象的属性	91
6.2.1	Document 的数值成员属性	92
6.2.2	关于锚点	94
6.2.3	关于图像	97
6.2.4	链接数组	101
6.3	Document 对象的方法	105
6.3.1	write() 函数	105
6.3.2	open(), close() 和 clear() 函数	105
6.4	举例: 文档统计配件	107
6.5	小结	109

第7章 详解 Navigator 对象树	110
7.1 Navigator 对象.....	110
7.2 Window 对象.....	114
7.2.1 window, self, parent 和 top	115
7.2.2 使用状态栏	115
7.2.3 对话框	117
7.2.4 打开和关闭窗口	119
7.2.5 窗口的焦点状态	120
7.2.6 setTimeout() 和 clearTimeout()	121
7.2.7 其他有关 Window 对象的内容	123
7.3 History 对象.....	123
7.4 Location 对象.....	124
7.5 举例: 一个小游戏	125
7.6 小结	130
第8章 JavaScript 与 Form	131
8.1 什么是 Form	131
8.1.1 定义一个表格	131
8.1.2 Form 与 Document	133
8.2 Form 对象.....	133
8.2.1 Form 对象的属性	134
8.2.2 submit() 和 reset()	134
8.3 Form 中的其他对象	135
8.3.1 按钮对象	135
8.3.2 文本输入域	139
8.3.3 选择性组件	142
8.4 举例: 报名表格	147
8.4.1 字段级验证	147
8.4.2 构造新文档	149
8.4.3 报名表完成了	150
8.5 小结	151
第9章 JavaScript 和 Frame	152
9.1 定义 Frame	152
9.1.1 框架标识	152
9.1.2 其他有关标识	154
9.2 Frame 对象.....	155
9.2.1 Frame 对象的属性和方法	155
9.2.2 frames 数组	158
9.3 框架窗口间的通信	159

9.3.1	框架窗口之间的相互引用	159
9.3.2	框架通信方式	161
9.4	举例: 读者问题调查表	165
9.5	小结	169
第 10 章	JavaScript 的高级技术概述	170
10.1	使用 Cookies	170
10.1.1	什么是 Cookies	170
10.1.2	Cookies 管理器	171
10.1.3	一个 Cookie 程序	173
10.2	使用 JavaApplet	175
10.2.1	什么是 JavaApplet	175
10.2.2	怎样在网页中嵌入 JavaApplet	176
10.2.3	JavaScript 中使用 JavaApplet	176
10.3	使用 ActiveX	178
10.3.1	什么是 ActiveX	179
10.3.2	加入 ActiveX	179
10.4	使用插件	180
10.4.1	什么是插件	180
10.4.2	在网页中嵌入类型文件	181
10.4.3	plugins 和 mimeTypees	182
10.4.4	利用 JavaScript 控制插件	183
10.5	增强 JavaScript 游戏功能	184
10.5.1	加入声音	184
10.5.2	使用动画	185
10.6	小结	187

第 1 章 JavaScript 简介

在正式开始学习之前，有些准备工作要做。在这一章里将不会介绍 JavaScript 的任何编程方面的知识，但是这一章是必不可少的。我们主要讨论下面的几个问题：

- Internet 的历史与现状
- WWW 与 Internet 的联系
- 什么是 HTML
- 为什么要在 HTML 中加入 JavaScript
- JavaScript 的优点

1.1 从 Internet 到 WWW

从 Internet 说起似乎离我们的主题远了一点，可是这又是不得不做的事，好在这件事并不是十分枯燥。

提起 Internet，国内网友们应该会热血沸腾。之所以这样说，一是因为它提供信息大量而快捷；二是因为大多数拨号上网者的电话费让人激动！无论如何，让我们先来看看一下它的历史和现状吧。

1.1.1 Internet 的起源和现状

Internet 的前身是 60 年代中期美国国防部的 ARPANet，这一点不会有什么疑问，但就 ARPANet 本身的目的而言，的确有各种各样的传说，不过大多数人对这样的争论不感兴趣，到底是政府防御核打击的计划促使计算机网络大行其道，还是某个高级领导人厌倦了桌面上同时放置几台计算机。总之，最后的结果是，PC 机甚至是中型机、大型机不再时髦了，那种“单机”作战的方式已经落伍了，Internet 靠其本身信息量大、方便快捷的优势站稳了脚跟。

有一点让人很感兴趣，在网络中协议的地位十分重要。任何人都可以制定网络协议，关键在于这个协议是否能够被广泛接受，具有商业价值。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。TCP/IP 是一个独立于硬件的协议，提供了不同的网络平台间互联的可能。ARPANet 上的许多机器都用 Unix，因此 TCP/IP 也就成为 ARPANet 普遍使用的协议。在这之后不久，美国国防部统一了 ARPANet 上的网络协议，使 TCP/IP 成为 ARPANet 上的标准。

新概念：协议的目的是使网络中的通信者在进行交流时对自身和对方的行为有所规范，在双方（多方）遵守了协议之后，信息就可以得以顺利地传送和理解。上面的定义并不十分严格，只是为了让读者能有个大致的概念。在众多的协议中，应用最广泛的是 TCP/IP (Transmission Control Protocol/Internet Protocol 传

输控制协议/互联网络协议)。TCP/IP 是一个网络集合, 它的目的是使不同的网络平台得以交换信息。在这本书中, 对协议理解到这个程度就足够了, 如果读者对有关协议的内容还有兴趣, 可以参考相关书籍。我推荐清华大学出版社出版的《TCP/IP 网络原理和技术》。

到了 1995 年, 政府作为 Internet 主管者的使命结束了, Internet 完全成为商业性的。同年 3 月, Internet 上的商业机构(.com)第一次超过了教育机构(.edu)。

Internet 的发展完全是指数式的, 在今天, 完全统计 Internet 的广度和深度是不现实的, 也是不必要的。总之, 现在的 Internet 已经渗透到每个国家的几乎任何行业中。你可以不知道克林顿为何要被弹劾, 但你不能不知道 Internet!

一些传统的 Internet 服务到现在仍被广泛地使用, 这包括:

- 电子邮件(E-mail)
- 新闻组(NewsGroup)
- 远程登陆(Telnet)
- 文件传输协议(FTP)

相信你对这些服务早有耳闻或是已经熟练使用了, 但在 Internet 中被最广泛使用和提及的是 WWW。

1.1.2 什么是 WWW

在 Internet 的初期, 它是如此的神秘, 甚至像古埃及的传说般不可捉摸, 这使得只有熟练掌握 Unix 的专家才可能使用它。出现这一局面的原因是传送不同种类信息的协议各不相同, 人们急切地需要一种统一的协议来使所有的信息传送整齐划一, 这就是 WWW 出现的原因。

WWW 是一个缩写, 完整的写法是 World Wide Web, 我们一般把它译做“万维网”。提到 Web, 就不得不说 Tim Berners-Lee, 人们甚至称他为“Web 之父”。当他在欧洲核研究机构(CERN)工作时, Berners-Lee 发现浏览 Internet 十分困难, 而且根本没有统一的界面。因此, 他发明了一种能在 Internet 上传输多种不同格式文件的协议, 这种协议是 WWW 的萌芽。第一个支持图形用户界面的客户端浏览器是 Mosaic, 它是由美国国家高级计算应用中心开发的, 它的模样和 Netscape2.0 很相似, 实际上这个浏览器就是 Netscape 的前身。

在 WWW 中, 我们同样要遵守一定的协议, 或者可以这样说, WWW 本身就是一个协议。实际上 WWW 与它所遵守的协议 http 常常被当作同义词对待。不妨看一下清华大学的网址:

<http://www.tsinghua.edu.cn>

http 实际就是一个协议名——超级文本传输协议。我们可以这样说, WWW 就是在 Internet 中遵守 http 的那一部分; 而 http 是 Internet 上众多协议中的一种, WWW 与 Internet 中其他部分的区别就是它们所遵守的协议不同。

Internet 中 WWW 得到如此广泛的应用, 以致于一提到 Internet, 人们的第一印象就是 WWW, 或是 NetScape 的界面。这也许是个误会, Internet 的含义比这要宽得多, 但是, 确实是 WWW 使 Internet 获得了生命。

图 1.1 就是一个典型的网页，确实五彩缤纷，是吧？

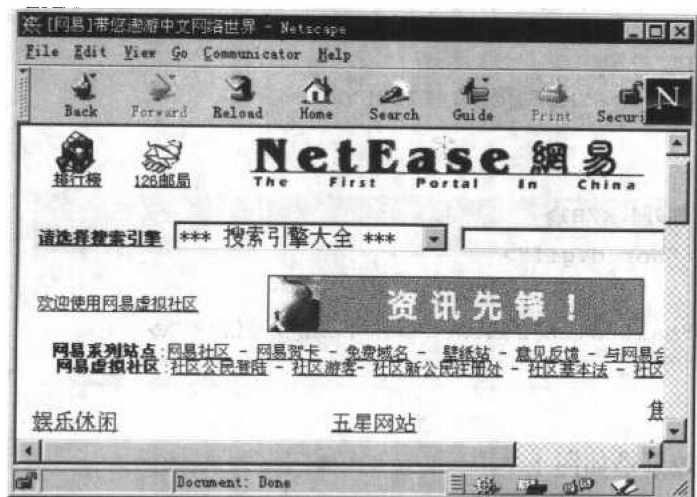


图 1.1 一个典型的网页

1.2 HTML 和 WWW

WWW 中的通用语言是 HTML 语言。通过使用这种充满各种各样标识的语言，就可以实现将一些相关信息组织起来的目的。本书需要读者具有一定的 HTML 基础，但是，如果没有接触过这类知识，也不要灰心，只是请仔细阅读下面这一节。

1.2.1 什么是 HTML

HTML (HyperText Markup Language) 是随着网络普及越来越常用的概念，它的全称是“超文本标识语言”，这是一种驱动 World Wide Web 的新型语言。Web 页都是用专门的语言——HTML 建立的。这种标识语言定义了一系列的标识，每个标识表示一定的意义。我们只要利用这样的标识，就可以按照特定的方式，也就是我们希望的方式，将网页中的数值、文字和图像加以组织，形成一个有自己风格而又有一定统一性的网页。

HTML 和传统的计算机语言很不相同。严格地说，它并不是一种编程语言，不需要对 HTML 语言的文件进行编译和连接，因此由 HTML 语言写出的东西更多的像文档，而不像程序，理解这一点十分重要。不妨这样解释这个过程：

- (1) 写出 HTML 文件。
- (2) 用浏览器装载 HTML 文件。
- (3) 浏览器“读”HTML 文件时，如果遇到标识，就按照标识的意义来显示它的文本或图形。

因此 HTML 语言的文件尽管有时仍被称为程序，但应该注意这只是出于方便的考虑。

你也许还记得在 C 语言教程中的第一个范例程序“Hello, World!”，现在来看看这个向世界问好的程序的 HTML 版本。下面就是它的 HTML 源文件：

【例 1.1】

```
<HTML>
<HEAD>
<TITLE>这是一个 HTML 的例子</TITLE>
</HEAD>
<BODY>
<H1>世界, 你好! </H1>
<IMG SRC="World.gif">
<HR>
<A HREF="mailto:tt@263.net">你可以给我发信!</A>
</BODY>
</HTML>
```

它所产生的效果是如图 1.2。



图 1.2 例 1.1 的运行结果

这个例子的运行结果没有前面的图 1.1 鲜艳、漂亮, 但是它的 HTML 语言源文件却十分典型和简单。你应该注意到以下几点:

- 每个标识都被包含在尖括号中。
- 标识一般是成对出现的, 后面的一个加有斜杠, 这表示结束之前的同名设置, 例如<HTML>和</HTML>。

在上面的例子中, 我们使用了一张图片, 这个语句:

```
<IMG SRC="World.gif">
```

我们只是在文件中告知浏览器图片放在何处 (这个例子中图片文件处于当前目录下), 并没有告知任何有关怎样显示图片的信息, 显示图片的具体工作由浏览器来完成。正是由于浏览器的这种功能, HTML 文件才如此的简单。在例子中, 我们用:

```
<A HREF = "mailto: tt@263.net">你可以给我发信!</A>
```

来表示一个超链接，这个超链接和一般的超链接是不大相同的。不同之处在于一般的超链接后会给出一个以 http 协议开头的 URL，而这个例子中的协议是 mailto，这表示我们用普通的 E-mail 的协议发送邮件，这样就没有必要开发 http 中传送电子邮件的新功能了。如果这个浏览器是有邮件功能的，那么点击这个超链接将激活邮件功能，进行邮件的发送。

这种格式表示一个超链接，它实际提供了一种对其他文件的引用方法。

新概念：超链接，当我们在浏览网页时，会发现鼠标在某些文字上移动时会变成手的形状，这段文字就是一个超链接。当我们点击它时，就会装入其他的网页或图像等。WWW 的浏览对于客户来说，就是从一个链接到另一个链接的过程，最终我们会找到需要的信息。

HTML 是如此的简单，却能完成几乎所有的我们希望能在网页中完成的工作。需要清楚的是这样的一个事实，尽管 C 语言、甚至汇编语言的功能十分强大，但它们要完成上面的工作，却需要成千上万行的代码。正所谓对症下药，HTML 绝对是 WWW 中需要的，它与平台无关，这是 C 语言等无法做到的，而又是 Internet 中所必需的。

1.2.2 HTML 的功能

这一小节是为了没有任何 HTML 语言知识的读者准备的，如果接触过这方面的知识，那么可以跳过这一小节。

从整体来说，一个由 HTML 写出的网页应该具备哪些作用呢？

首先，它应该具备一定的文件格式结构。一般来说，一个 HTML 文档包括这样三个标识对：

```
<HTML>
<HEAD>
</HEAD>
<BODY>
</BODY>
</HTML>
```

请注意这三个标识的嵌套次序：<HTML>和</HTML>之间的文字是 HTML 文档的正文；在<HEAD>和</HEAD>之间的内容一般不会直接影响网页的外观，但是这个部分之间的内容确实是 HTML 文档中最先被处理的部分；<BODY>和</BODY>之间的内容是实际将要显示的网页外观的信息。

其次，这个网页应该可以提供一定的文字或者图形信息，这两个功能的实现很简单。在 HTML 中，只需要在文件中的<BODY>和</BODY>标识之间直接写入文字，就可以显示出这些文字了，而图像的显示方法已在上一小节中介绍了。

再次，在一个 HTML 文档中应该包括与其他文档的联系，这是通过超链接来实现的。这一内容也在上一小节中出现了，而且在后面的章节中还会进一步讲解这个概念，也会看到更多的实例。

另外，在 HTML 中也提供一种称作表格 (Form) 的机制，通过填写表格服务器可以