

主编 张廷国

主审 蒋立源

JavaScript

程序设计及应用



西北工业大学出版社

12JA

(陕)新登字 009 号

【内容简介】 JavaScript 是最近才出现的一种新的程序设计语言。与传统的超文本标记语言 HTML 相结合,JavaScript 语言采用基于对象的思想,为 Web 设计者提供了制作具有动态主页所必需的各种对象和一组功能强大、使用灵活的程序控制结构,与仅使用 HTML 的效果相比较,JavaScript 能使用户制作出更为生动、精美的 Web 主页。

本书以通俗易懂的语言、循序渐进的方法,论述了 Internet 和 WWW 服务、超文本标记语言 HTML、JavaScript 语言的语法结构、对象的建立和使用、JavaScript 的内部对象、Netscape Navigator 浏览器所提供的对象、使用 JavaScript 语言在 Internet 上提供多媒体信息服务等内容。本书的附录还提供了 HTML 和 JavaScript 语法索引。

本书可作为具有计算机基础知识人员的学习资料,也可供有一定程序设计经验的读者进一步提高之用,还可作为 HTML 和 JavaScript 的索引工具。

本书所有示例都以较艺术的方法组织在另附的软盘上,该软盘包括例题演播室、热线站台等内容。使用此软盘,读者可直观方便地学习 JavaScript 程序设计的主要知识,也可以快捷地进入我们提供的热线站台进行浏览。有需要此盘的读者可向西北工业大学出版社联系购买。

JavaScript 程序设计及应用

主 编 张延园

主 审 蒋立源

责任编辑 雷 军

责任校对 钱伟峰

*

©1997 西北工业大学出版社出版发行
(710072 西安市友谊西路 127 号 电话 8491147)

全国各地新华书店经销

西安陆军学院印刷厂印装

ISBN 7-5612-0945-2/TP·129

*

开本: 787×1092 毫米 1/16 印张: 11.375 字数: 271 千字
1997 年 4 月第 1 版 1997 年 4 月第 1 次印刷

印数: 1—6 000 册 定价: 16.00 元

购买本社出版的图书,如有缺页、错页的,本社发行部负责调换。

前 言

全球性的信息网络——Internet 与人们的工作、学习和生活越来越密切,这与 WWW 的广泛使用是分不开的。今天,Internet 上分布着许多用于提供 WWW 服务的 Web 服务器,它们向广大的 Internet 用户提供多种形式的信息服务,包括普通文本、声音、图像等。Internet 用户只需一个标准的浏览器软件,就能在全球各地享用这些多媒体信息。Internet 上的这些多媒体信息服务正在给社会生活带来巨大的影响。所谓无纸贸易、企业网络(Intranet)、电子购物、电子影院、网上学校等新事物已经或正在成为现实。

目前,上述网络多媒体信息的主要载体是所谓网页(Web Page)。以往,网页主要用超文本标记语言 HTML 来描述和设计。现在,人们越来越多地使用由 Netscape 公司推出的底稿式程序设计语言 JavaScript 和由 Sun Microsystem 公司推出的 Java 语言,并与 HTML 相结合,以便设计功能更加强大、更有活力的网页,在 Internet 上提供 WWW 信息服务。

随着我国越来越多的单位通过 CERNET、CHINANET 等全国性计算机网络连入 Internet,许多人希望能使用 HTML、JavaScript、Java 等语言设计网页,并在 Internet 上提供自己的 WWW 信息服务。目前,国内在这方面的资料、书籍较少,为了满足广大网页设计人员的需要,我们组织编写了这本介绍 JavaScript 和 Java 语言的书。

本书主要介绍了 Netscape 推出的 JavaScript 语言。第一章为绪论,一方面对 JavaScript 语言作一个概要性的介绍,另一方面讲述了与 JavaScript 语言相关的概念及有关的背景知识,为读者阅读本书的其他章节提供必要的基础。第二章简单介绍与 JavaScript 语言关系非常密切的超文本标记语言 HTML。第三章通过简单示例介绍 JavaScript 语言的初步知识,使读者对该语言有一个感性认识。第四章描述 JavaScript 语言的语法成份,包括组成 JavaScript 语言的基本元素、JavaScript 的控制流、JavaScript 的函数以及 JavaScript 的函数在 HTML 事件处理器中的应用。第五章讲述如何在 JavaScript 程序中建立并使用对象。第六章介绍 Netscape Navigator 所提供的重要对象之一——文档(document)对象。第七章讲述如何用 Netscape Navigator 所提供的、用于建立格式化交互界面的 form 对象,建立能与用户进行交互的网页。第八章讲述如何使用 Netscape Navigator 的窗口对象、历史对象、定位对象,使网页具有更加友好的用户界面。第九章是一个综合实例,展示如何综合使用 JavaScript 语言建立功能强大、具有友好用户界面的 Web 主页,以期进一步提高读者编写 JavaScript 程序的能力。书末是三个附录,用作 JavaScript 语言的资料索引。附录一给出了 HTML 较为详细的描述;附录二用 BNF 范式给出 JavaScript 的语法定义;附录三给出了 JavaScript 的层次结构。

本书语言流畅、内容完备、示例丰富,并采用循序渐进的方法引导读者逐步掌握 JavaScript 语言。既适合网页初学者用作 JavaScript 语言的入门书,又可帮助有经验的网页制作者进一步提高,同时还是一本讲述 JavaScript 语言的很好的教材。本书所有示例都以较艺术的方法组织在所附的软盘上。

本书由西北工业大学软件工程中心张延园主编,参加编写工作的有吴健、康慕宁、高集荣、陈启安、李战怀、刘敏、孙夕凌、李红燕、李静和马满福等同志。其中,张延园同志对本书编写工

作进行了全面的规划,吴健同志对各章的内容作了精心的组织,陈启安同志对综合示例进行了设计和调试。在本书编写过程中,胡正国教授从国外寄来了必要的参考资料,蒋立源教授对全部书稿进行了审阅和仔细的修改加工。我们特表衷心的感谢。此外,西北工业大学出版社的许多同志对本书的出版给予了大力的支持和协助,在此也深表谢意。

由于 JavaScript 是一种较新的程序设计语言,加之作者的水平有限,本书疏漏之处在所难免,欢迎广大读者批评指正。

编 者

1996 年 11 月于西安

目 录

第一章 绪论	1
1.1 Internet 和 WWW 简介	1
1.2 Web 浏览器和 HTML	2
1.3 HTML、Java 和 JavaScript	3
1.3.1 Java 语言	4
1.3.2 JavaScript 语言	5
第二章 用 HTML 制作 home page	7
2.1 HTML 概述	7
2.2 HTML 文档结构	7
2.3 HTML 标记简介	9
2.3.1 超文本链接	9
2.3.2 内嵌图像	10
2.3.3 接受用户输入的 FORM	10
2.3.4 加载其他语言的脚本代码	10
2.3.5 HTML 注释符	10
2.3.6 格式控制标记	10
2.3.7 字体控制标记	11
第三章 走进 JavaScript	13
3.1 JavaScript 程序示例	13
3.2 JavaScript 和 HTML	16
3.2.1 在 HTML 中引入 JavaScript	16
3.2.2 处理顺序	17
第四章 JavaScript 程序设计基础	18
4.1 JavaScript 的基本语法元素	18
4.1.1 注释	19
4.1.2 文字	19
4.1.3 标识符	21
4.1.4 关键字	21
4.1.5 分隔符	22
4.1.6 变量定义	22

4.1.7	构造表达式及使用运算符	24
4.1.8	小结	29
4.2	JavaScript 的控制流	29
4.2.1	使用块和注释来组织代码	30
4.2.2	使用缩排法	31
4.2.3	使用 if-else 语句	32
4.2.4	用于重复执行代码的 for 循环	34
4.2.5	使用 while 循环	35
4.2.6	使用 break 和 continue 控制循环	35
4.3	JavaScript 的函数	36
4.3.1	在 JavaScript 程序中创建函数	36
4.3.2	在函数中使用 return 语句	38
4.3.3	允许改变参数个数	38
4.3.4	使用函数和事件处理器	39
4.3.5	使用事件处理器触发函数	40
第五章	在 JavaScript 中创建并使用对象	42
5.1	JavaScript 的对象模型	42
5.1.1	访问对象性质	43
5.1.2	调用对象方法	45
5.2	JavaScript 的内部对象	46
5.3	JavaScript 中创建和使用对象	50
5.4	创建和使用对象的一个例子	55
第六章	创建文档——Document 对象	63
6.1	Netscape Navigator 对象简介	63
6.2	Document 对象	64
6.2.1	HTML 中等价的 Document 对象	64
6.2.2	Document 对象的性质	66
6.2.3	Document 对象的方法	73
6.2.4	文档对象事件处理程序	75
6.2.5	小结	76
第七章	编制交互式文档	77
7.1	Form 对象	77
7.1.1	Form 对象的结构	77
7.1.2	Form 对象的性质	78
7.1.3	Form 对象的方法	80
7.1.4	Form 对象的访问	80

7.2 HTML 与 JavaScript 中的交互成份	84
7.2.1 Form 的元素	84
7.2.2 Form 对象元素中的事件处理器	91
7.3 Form 示例	93
第八章 page 中的其他对象	95
8.1 Window(窗口)	95
8.1.1 Window 对象的 HTML 标记	95
8.1.2 Window 对象的性质	96
8.1.3 Window 对象的方法	100
8.1.4 Window 对象的事件处理器	103
8.2 History(历史)	103
8.2.1 History 对象的性质	103
8.2.2 History 对象的方法	103
8.3 Location(定位)	104
第九章 一个完整的 JavaScript 程序示例	105
9.1 例子的运行环境及安装	105
9.2 例子设计中用到的技术	105
9.2.1 多文本选择	105
9.2.2 返回按钮	107
9.2.3 进入新窗口	109
9.3 主页	109
9.4 软件工程中心简介	111
9.5 例子演播室	111
9.6 热线站台	113
9.7 与作者交朋友	115
9.8 完整的源程序清单	117
附录一 HTML 举要	148
附录二 JavaScript 语法定义	162
附录三 JavaScript 层次结构	165

第一章 绪 论

1.1 Internet 和 WWW 简介

本书旨在向读者介绍 JavaScript 程序设计语言。由于目前该语言主要用于制作 Internet 上的网页,从而在 Internet 上为用户提供信息服务,因此有必要对 Internet 和其上最受欢迎的 WWW(World Wide Web)服务作一个简单介绍。

当前,Internet 是最热门的话题之一,而 WWW 作为 Internet 上最受欢迎的服务,几乎成了 Internet 的同义语。于是,人们自然会问:何谓 Internet? 何谓 WWW 服务?

要给 Internet 下一个贴切的定义是十分困难的,因为每一个人从不同的角度可以给出不同的表述。这里,我们只能通过考察 Internet 的发展历史来理解什么是 Internet。

1957 年,苏联发射了世界上第一颗人造地球卫星,这使美国政府大吃一惊。作为对此事的反应,美国急切希望消除在空间竞赛方面与苏联的差距,建立了 ARPA——高级研究计划局,建立这个局的目的是研究高级技术。鉴于美国国防部是当时最大的计算机用户,因此,一些人建议 ARPA 应支持计算机科学中大规模的基础理论研究。在 60 年代初期,国防部又委托建立 ARPAnet 用于研究网络协议,这应该算是 Internet 的始祖了。到 1969 年,由于美国国内的某种原因,须要明确表明国防部的经费已真正用于军事目的,遂将 ARPA 改名为 DARPA(D 是国防的缩写),相应地,ARPAnet 也被更名为 DARPAnt。这一时期的主要特点是只有为数不多的研究人员为军事目的在使用 ARPAnet。

ARPAnet 的初期运行情况即已表明,这种计算机网络应该有一种标准化的通信协议。为此,在 1973 年推出了 TCP/IP 协议来满足这种需要,这就是现在仍在 Internet 上用来传输信息的协议。TCP/IP 协议或称为“传输控制协议/网际协议”,是一个能在不同网络平台间进行信息交换的协议族,TCP(传输控制协议)和 IP(网间协议)是该协议族中的两个最重要也是最著名的协议。TCP/IP 应视为一种与硬件无关的协议,运行在基于硬件的协议之上。这意味着可以使用诸如以太网、ATM 以及任何正在使用或正在开发的硬件环境来传输信息。20 年来 ARPAnet 的运行经验表明,TCP/IP 是一个非常可靠且实用的网络协议。

Internet 历史上的重大事件之一是高速 NSFnet(美国国家科学基础网)的出现。80 年代末期,美国国家科学基础局(NSF)建立了 5 个超级计算机中心。当时,世界上最快的计算机只有研制武器的人员和少数大公司的研究人员才能使用。通过建立超级计算中心,NSF 想让这些资源能用于各种学术研究。之所以仅建立 5 个中心是因为这些超级计算机太昂贵了,只能为相关人员所共享,这就出现了通讯方面的问题。一方面这 5 个中心需要互连,另一方面要使这些中心的用户能访问它们。开始,NSF 想借助于 ARPAnet,但由于官僚主义繁琐的官样文章,这一愿望未能实现。因此,NSF 决定基于 ARPAnet 的 IP 技术,建立自己的网络,这就是 NSFnet。NSFnet 是用 56Kbps 的电话线将 5 个中心连接起来所构成的计算机网。NSF 鼓励各大学也加入其 NSFnet。显然,每个主机(简单地说,一个具有 IP 地址的计算机就称为 Internet

上的一台主机)都用 56Kbps 的电话线连接到这 5 台超级计算机上是不现实的,因为这要使用很多、很长的电话线路,因此,NSF 利用了已有的局域网。这些局域网相互就近互连,最后再接入 NSFnet。可见,NSFnet 的意义是它使得大量的计算机用户实现了互连,再加上具有较高的通讯速率,所以,原来 ARPAnet(也就是 DARPA net)的业务也就逐渐为 NSFnet 所取代,到 1990 年 3 月,ARPAnet 便永久退出了历史的舞台。

现在,Internet 上除了具有运行 TCP/IP 的网络外,运行其他协议的网络也逐渐加入了进来,例如 USEnet、BITnet、DECnet 等,这些网络一般都使用所谓网关连入 Internet。

目前,全世界大部分国家都已通过各种方法连到了 Internet 上,因此,Internet 上的主机数量在急剧膨胀。1983 年,Internet 上只有 500 台主机;由于 NSFnet 的建立,1986 年底,有了 5 000 台 Internet 主机;1989 年,Internet 上的主机超过了 10 万台,NSFnet 也升级到了 1.5 Mbps 的 T1 线路;到 1992 年,Internet 上有 100 万台以上的主机,而这才是其急剧增长的开始。到 1994 年底,Internet 所连的主机超过了 400 万台;1995 年,NSF 解除了对 NSFnet 的委托,启动这项新技术的政府行为便告结束,而私有企业将继续发挥作用。1995 年 3 月,com(商业系统的 Internet 地址)的正式数量首次超过了 edu(教育系统的 Internet 地址)的正式数量,这标志着未来 Internet 的商业模式的真正开始。

大量主机(即大量的用户)连接到 Internet,共享其上的信息资源,同时也享用着 Internet 上的各种服务。这些服务有电子邮件(E-mail)、文件传输(FTP)、远程终端登录(Telnet)以及电子公告牌(BBS),等等。其中,一项历史不长但引人注目的信息服务方式是 WWW 服务,WWW 意即全球网。实际上,WWW 服务的主要特点是为用户提供全球范围内的超文本信息服务。

所谓超文本信息,是指在文本中有指向其他超文本信息的指针,用户一般要使用专门的软件(如本书后面将要提到的各种浏览器)才能阅读超文本信息。在阅读超文本信息的同时,用户还可通过超文本指针阅读另一个超文本信息,该超文本信息既可以在本地,也可以在 Internet 的另一个 WWW 服务器上。现在,超文本信息往往具有多媒体的形式。

Internet 上 WWW 服务方式是:在一些愿意提供 WWW 服务的处所(如政府部门、大学、甚至个人)设立一台(或数台)WWW 服务器,其中存放超文本信息,并将其 WWW 服务器联接在 Internet 上。用户可在自己的计算机上使用专门的客户软件(如浏览器等),通过 Internet 向自己所希望的 WWW 服务器请求信息服务。WWW 服务器通过向客户端发送超文本信息(或其他形式的信息)作为对上述请求的响应。

最初的 WWW 超文本信息服务,是由欧洲粒子研究实验室的 Tim Berners-Lee 开发并提供的,那时它只是一个基于文本的界面。直到美国伊利诺州的国家超级计算应用中心(NC SA)为 Web(WWW 服务又称为 Web 服务)开发了一个图形界面之后,WWW 才变得像今天这样强大,这个界面就是所谓的 Mosaic。今天的 WWW 服务能提供诸如普通文本、图像、声音等各种媒体形式的多种信息服务。

1.2 Web 浏览器和 HTML

用户在 Internet 网络的客户端使用浏览器便能获得 WWW 信息服务。当前较流行的浏览器有 Netscape Navigator、Microsoft Internet Explorer 等。目前,浏览器从 WWW 服务器那里

得到的通常是用 HTML 语言编写的文件(称为 HTML 文档,或称为网页——Web Page)。

HTML(HyperText Markup Language)是一种超文本标记语言。一个 HTML“程序”(准确地说应称为 HTML 文档)实际上是在一个普通的 ASCII 文本文件中加入了一些标记(tag)和指令而形成的,用以控制浏览器如何在客户机上显示信息。HTML 的简述和详细资料可分别参见本书的第二章和附录,这里只对它的主要特点和相关概念作一简单介绍。HTML 文档中,在客户端表现的主体仍是文件中的文字、图像、声音等信息,其中的标记仅用来控制客户端浏览器如何表现这些信息。例如它可以控制文字在表现时的大小、字体、风格;控制文本的换行、段落安排;控制图像表现时的大小、位置;将信息组织成表格显示;在用户屏幕上安排多窗口;指示多种协议(ftp、gopher、telnet、http 等)的超文本连结;提供用户与 Web 服务器(即 WWW 服务器)进行信息交互的手段,等等。

HTML 已形成了标准,且已发布了多种版本。目前,HTML 已成为在 Internet 上描述多媒体超文本信息的标准,它能做到与平台无关。除了具有平台独立的特性以外,HTML 还具有很好的可扩展性,这主要体现在许多浏览器厂商所提供的浏览器能够忽略 HTML 文件中所不识别的新标记,且不影响它对 HTML 文件的解释和表现,从而有利于 HTML 的升级和扩展。

此外,HTML 还提供与用户交互的能力,这种能力具有客户/服务器模式。开发人员可在 Web 服务器一端开发 CGI 程序,而客户端的 HTML 文档可接受用户的输入信息,并发送给服务器端的 CGI 程序。这些信息一般是用户对 CGI 程序的一个信息请求,CGI 程序对该请求作适当的处理,并把结果以 HTML 文档的形式返回给客户浏览器,从而进一步表现给用户。例如,当用户从客户端发出查询某单位电话号码的请求时,Web 服务器上的 CGI 程序经响应和处理后,将查询得到的电话号码以 HTML 文档的形式发送给客户。CGI(Common Gateway Interface)是一种通用网关接口规范。在 Web 服务器上接受客户请求并做出响应的程序可称为一个 Web 网关(Gateway)。Web 服务器软件一般都提供了开发网关的支持环境,该支持环境应遵循一定的标准,这就是所谓的 CGI 规范。遵循 CGI 规范的网关软件一般称为 CGI 程序。CGI 程序可用 C、Perl 等语言编写,只要遵循 CGI 规范来使用 Web 服务器的环境变量,接受并处理服务请求即可。

当然,CGI 程序处理用户请求的能力尚不够强大,现在的趋势是将 Web 服务器与数据库服务器结合起来提供动态信息服务,这是当前很热门的一项研究课题。在将 Web 服务器与数据库服务器集成起来的过程中,Web 网关起到了非常重要的中间桥梁作用。一方面它接受 Web 服务器的数据库查询请求,另一方面它到数据库服务器中进行信息查询,并把结果组织成 HTML 文档交给 Web 服务器,最后由 Web 服务器将此 HTML 文档作为结果,发送给最初发出查询请求的客户浏览器。现在,许多公司和研究机构正致力于研制将 Web 服务器与数据库服务器集成起来的软件,即 Web 网关软件,并已推出了很多这方面的产品。

1.3 HTML、Java 和 JavaScript

HTML 的另一个重要特点是可在 HTML 文档中嵌入 JavaScript 程序或 Java 小应用程序(Java applet),从而极大地增强了 HTML 文档的表现能力。下面分别对 Java 和 JavaScript 作一个概要介绍。

1.3.1 Java 语言

1991 年 4 月, Sun 公司组织了一个从事商用电子产品开发的小组, 命名为 Green 小组。Green 小组研究智能型(电视)选台器、CD-ROM 等商用电子产品。在研制过程中, 他们发现须要开发一种通用嵌入式软件开发环境, 为他们的商用电子产品配备软件系统。一开始他们就认识到需要一种独立于平台的开发环境, 因为公众对他们的机器内是什么处理器并不关心, 而只要机器能很好地工作就满意了。换言之, 针对特定平台的开发将是一种不可取的方案。Green 小组中的 James Gosling 最初着手试图扩展 C++ 编译器以达此目的, 但不久就认识到这需要进行大量的工作。James Gosling 继而又为 Green 项目开发了一种新的语言——Oak。之所以这样命名, 是因为当 Gosling 正在为这一新语言注册时, 他抬头看到窗外有一棵橡树(Oak)。但这一名称未能通过商标核查, 于是就改成了 Java。这就是 Java 语言的诞生过程。

Green 小组曾用 Java 及相关支持环境对 Time-Warner 公司的智能型选台器操作系统和可控视频技术的设计方案进行投标, 但不幸失败了。几乎与此同时, Internet 上的 WWW 服务正风行起来, 于是很快便发现, Java 语言及相关技术非常适合这一领域。此后, Java 就成了这一领域令人瞩目的程序设计语言, 遂于 1995 年 5 月 23 日, 由 Sun 公司在 SunWorld'95 上正式公布。

Java 在结构上容许把中间代码动态地加载到像 Internet 这样的异构网络上并加以执行, 从而改变了 Internet 和 WWW 的被动特性。Java 通过把下面的特性加入其体系结构而提供这种改变。

(1) 可移植性 这意味着它可在任何配置了 Java 解释器的机器上运行。一种语言要想应用于 Internet, 这个特性是很重要的, 因为在 Internet 上, 一块以太网卡所连接的可以是任何平台;

(2) 健壮性 语言的特点和运行时环境应保证代码的运行是良性的(即具有良好行为), 这是努力做到可移植性和努力满足坚固性应用要求的必然结果;

(3) 安全性 除了要避免使客户受到无意的攻击之外, 还应使客户免遭有意的攻击。Internet 对病毒真是太友好了, 因为它允许各种程序被下载运行;

(4) 面向对象特性 该语言在根本上是面向对象的, 允许静态和动态继承及代码重用;

(5) 动态特性 Java 的动态特性是面向对象设计的扩充, 它允许运行时的扩展;

(6) 高性能 Java 语言支持几种高级性能, 比如多线程、即时编译(指运行时的编译)和使用本地目标码;

(7) 易使用 因为该语言可被视为来自于 C 和 C++, 所以对熟悉后者的程序设计人员来说极易掌握。同时, 该环境容错性能较好, 例如对指针和内存管理方面的错误。

Java 是一个独立的程序设计语言, 它需要一个开发环境 JDK(Java Development Kit)来“翻译”及提供开发中所需要的程序库(因 Java 是一个面向对象的语言, 其程序库实际是类库)。在这里, 我们所说的翻译是指由 JDK 将 Java 源程序转换为一种中间代码, 尔后在用户访问 Web 服务器时, 服务器便把这个中间代码传送给客户浏览器, 浏览器再解释执行这个中间代码, 从而向用户展示执行结果。第一个能执行 Java 程序的浏览器是 Sun 公司的 HotJava, 现在 Netscape 的浏览器、Microsoft 的浏览器、以及其他绝大多数新的浏览器也都能执行 Java 程序。Java 程序有两种形式: 一种称为独立程序, 它可在 Java 开发环境中执行; 另一种称为 Java

小应用程序,它可嵌入在浏览器中执行。

Java 并不局限于特定的浏览器平台。比如,现在的浏览器只能接受并显示 HTML 文档中的 GIF 和 JPEG 格式的图像;而使用 Java,对浏览器的这种依赖将不复存在,因为作为一种独立的程序设计语言,开发人员完全可以用 Java 语言编写自己的程序,显示各种格式的图像文件。

总之,使用 Java,可以开发更灵活、功能更强且独立于平台的网络应用程序,这就是 Java 在 Internet 上大受欢迎的重要原因。

1.3.2 JavaScript 语言

JavaScript 是一种底稿(script)式语言,首先在 Netscape 2.0 版中实现,其目标是用来提供一种小型、动态、基于对象(注意不是面向对象)的语言。它既可用于建立能嵌入在 HTML 文档中并在客户端建立交互的独立的底稿程序,又可以用来与 Java 的小应用程序协作从而在网页中引入对象。另外,JavaScript 几乎与现今的 CGI 一样,可建立用于服务器端的底稿程序。这一功能为 HTML 作者提供了创建强大网页的灵活性,而不须要依赖复杂的 CGI 程序;在处理用户输入时也可不必与 Web 服务器打交道。

JavaScript 实际上已在 Netscape 及许多其他的浏览器中实现,而且是一个与 Java 完全独立的语言。在其第一个 Beta 测试版中,该语言称为 LiveScript——这与 Netscape 公司为其 HTTP(传送 HTML 文档所用的超文本传输协议,它显然是一个应用层协议)产品系列开发的服务器协议 LiveWire 有关。不久就逐渐发现,Java 可用来接管服务器软件的许多工作,而有一个客户端编程工具将会是很有用的。因为 LiveScript 能很好地满足这一点,所以 Netscape 公司将该语言重新命名为 JavaScript,并作为 Java 语言的一个伴生工具推向市场。Java 和 JavaScript 之间的主要联系是它们在语法上相似,并且 JavaScript 能够访问 Java 类的公共成员。JavaScript 自身就是一个很有用的语言,它可用于建立具有交互能力的复杂程序,而并不依赖于运行 CGI 程序所用的网络传输。

JavaScript 是一种嵌入或集成在标准 HTML 文件中的解释型语言。在 HTML 文档被加载时,Netscape 的浏览器解释 JavaScript 程序并执行指定的操作。JavaScript 只能在实现了 JavaScript 解释器的浏览器中运行。

JavaScript 实际上是在 Java 这样功能齐全的语言和 HTML 这样仅能控制浏览器的简单底稿语言之间的一种折中。它填补了这两者间的空缺。JavaScript 语言有一些重要特点使其在开发客户端网页方面有很强的能力。

(1) 简单 JavaScript 很简单,而且以 Java 的语法为基础,这使该语言对那些已习惯 Java 和 C++ 的人很容易学习,它也可作为学习 Java 的一个阶梯;

(2) 动态 JavaScript 是动态的,它能对用户通过浏览器给出的输入作出反应而毋须依赖服务器端的程序,比如 CGI 程序;

(3) 基于对象 JavaScript 是基于对象的,这是指它能实现自身的对象。它也能与浏览器中的对象和插件(Plug-in)对象进行交互。

JavaScript 的这些特点,使它与 HTML 和 Java 这两个标准都建立了联系。这两个标准要么已成为开发网页的事实标准(HTML),要么距此目标还很远(Java)。重要的是了解这些特点,并在设计、构造以及维护网页时能用这些特点满足自己的需要。下面更进一步分述

JavaScript 的上述特点。

首先,JavaScript 是简单的。JavaScript 语言具有 Java 的基本语法要素和控制流结构,以便这两种语言间能进行快速转换。举例来说,JavaScript 的所有控制流语句在 Java 中都有,诸如 if...else、while、for 循环等。还有,所有的算术表达式、逻辑表达式以及字符串语句都从 Java 移到了 JavaScript 中。这意味着程序员可使用这两种语言的公共成份同时建立 Java 程序和 JavaScript 程序。

JavaScript 并不是一种强类型语言,因此变量不须要像在 Java 中那样进行说明。在 Java 中,每当建立一个变量时,都必须告诉编译器该变量是什么类型,此后编译器将进行检查,以确保此变量具有它应有的类型,这称为强类型检查。JavaScript 中没有这种强类型检查,变量可以自由地与任何类型的数据建立联系。因此,在说明变量时,没有必要指明该变量的类型。这种随意性有利有弊,不过就大多数底稿型语言来说,像 Java 那样严格的类型检查并非总是必要的。

除了宽松的类型检查外,JavaScript 也不强求建立类,也不强求具有面向对象的形式。这可使程序员尽可能以最有效的方法使用该语言,不须要不断地查询类层次图和类的继承性。这是因为并不须要为一个非常简单的函数去建立类、成员方法和变量,程序员只须编写此函数本身,并在适当事件发生时调用该函数即可。

JavaScript 是动态的。JavaScript 的一大优点是能在其环境中很好地处理动态事件。通过与嵌入到 HTML 中的格式输入(form),以及与 Netscape 浏览器所支持的 JavaScript 对象进行协作,所建立的程序能以事件驱动的方式对用户输入作出反应。这种能力对开发客户端软件是很有好处的,它无须太复杂,也无须依赖于对服务器的网络调用。

最后一点,JavaScript 是基于对象的。虽然严格地说,JavaScript 不是一个真正的面向对象的程序设计语言,但它可视为基于对象的,就是说,它的许多功能来自于程序与一些对象的协作,而这些对象对程序设计环境来说是开放的。它们包括 Netscape 所提供的 Window、Location、History、和 Document 对象,还有随网页加载进来的 Java applet 对象,以及集成到 Netscape 浏览器中的任何插件对象。通过与这些对象的协作,JavaScript 语言能够把这些高级程序所提供的更加强大的功能集成进来。由 Netscape 提供给 JavaScript 解释器的对象,使程序员能收集运行环境方面的信息,并能改变文档的显示方式。例如,可以访问浏览器会话文档的历史,并根据用户已到达的位置来控制下一步访问何处。另外,还可以控制诸如背景颜色这一类显示特性。不过当使用 HTML 时,对每个文档都应作一次设置;若使用 JavaScript,则可立即规定这些改变而毋须从服务器上重新调入一个网页。

从某种意义上说,JavaScript 与 HTML 非常接近,HTML 中的每一种标记都可成为 JavaScript 的一种对象。此外,JavaScript 还有一些对象在 HTML 并没有相应的标记。因此,就所建网页之表现能力及功能而言,使用 JavaScript 比使用 HTML 将更为有效。

第二章 用 HTML 制作 home page

由于 JavaScript 代码是嵌入在 HTML 文档中被解释执行的,所以作为本书的入门,我们首先对 HTML 作一概括介绍,然后论述 HTML 文档结构和 HTML 标记,最后在此基础上,介绍用 HTML 制作 home page 的方法。

2.1 HTML 概述

超文本标识语言 HTML 是 WWW 的一种描述型语言,是标准通用标识语言 SGML 的一个简化版本。HTML 打破了以物理地域为组织原则的信息管理方式,使 home page 制作者能按信息内容的相关性将分布于整个网络的资源(包括文本、图像等)有效地组织在一起。

1989 年,欧洲粒子物理实验室 CERN 的 Tim Berners-Lee 设计开发了 HTML。后来为了适应 Internet 的发展,HTML 几经修改,增加了图形等复杂的多媒体功能,遂于 1990 年为 Internet 正式接受。此后,HTML 文档便很快成为发表科学论文和交流研究资料等的理想媒体。HTML 作为一种新型语言,能迅速为人们广泛接受,其主要原因就在于 HTML 提供了强大的文本描述能力,具体表现在以下 4 个方面。

1. 创建独立于平台的文档

用 HTML 书写的文档能在 Internet 的任何浏览器上运行。相对于 Internet 的复杂性而言,这一特点显得尤为重要,因为它可使文档的制作者无须关心自己书写的文档的运行场所。

2. 创建和 Internet 上其他文档的连接

使编写文档的作者能把文档里的文字或图形与另一个文档连接在一起,而不必考虑这些文档是保存在同一计算机里,还是保存在网络上的其他地方。这就使用户不用记忆复杂的命令,也无须记忆枯燥的网络地址,而只须在文字和图像的引导下,用鼠标双点选取,便可进入自己感兴趣的内容。

3. 引入图形和多媒体

HTML 支持在 home page 中展示图形、动画、声音等多媒体效果。

4. 具有和 Internet 上其他资源连接的功能

从表现形式上看,我们可以把 HTML 看作是一种排版语言。它具有各种字形、字体;能显示表格,设置背景、前景;还可实现网上 mail 等功能。同时,HTML 又是 JavaScript 和 Java 的宿主语言,可将 JavaScript 或 Java 的语句嵌入在 HTML 文档中。WWW 和 HTML 的结合,可使 Web 页利用 HTML 来组织文档,建立文档间的连接,创建易于在平台上移植的超级文档。可以说,HTML 把我们带进了一个崭新的网络世界。

2.2 HTML 文档结构

HTML 文档是一系列标记(tag)和其他文字信息组成的文件,它是各种浏览器解释

HTML 语言的最小单位。标记是由符号“<”和“>”及置于两者之间的文字构成的,大多数标记符号成对出现,前一个表示开始,后一个是在“< >”中加“/”,表示此标记的辖域到此结束。下面的例 2.1 可用来具体认识一下 HTML 标记和 HTML 文档的结构。

【例 2.1】

```
<HTML>
<HEAD>
<TITLE>Sample Of HTML</TITLE>
</HEAD>
<BODY>
这是一个用 HTML 组织的页!
</BODY>
</HTML>
```

上述 HTML 文件可在任何一个文本编辑器上编辑,标记中的英文字母用大小写均可。例 2.1 中的程序由浏览器解释执行并显示结果如图 2.1 所示。

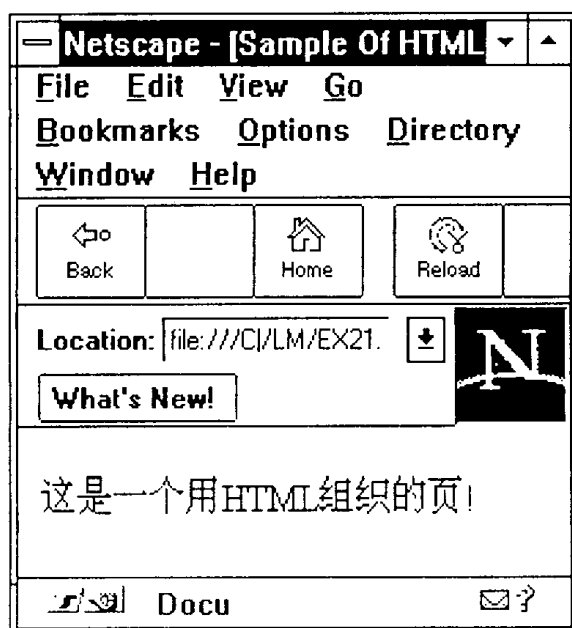


图 2.1 一个简单的 HTML 文档

标记一般是成对出现的,最常用的几种标记的含义是:

(1) <HTML>告诉浏览器由此开始是一个 HTML 文件,</HTML>表明 HTML 文件结束,通常 HTML 文件的扩展名是.html 或.htm;

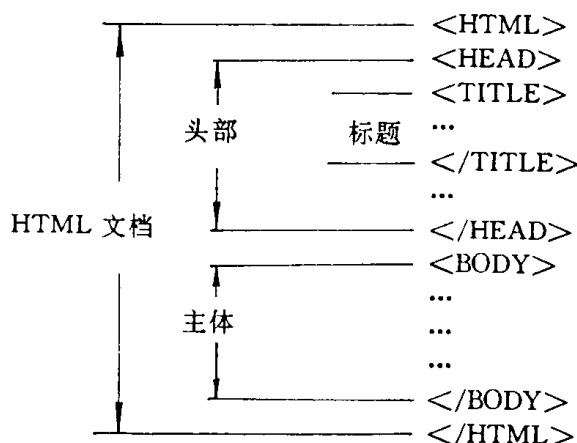
(2) <HEAD>表示 HTML 文档的头部开始,</HEAD>表示头部结束,头部必须包含对文档标题的定义;

(3) <TITLE></TITLE>这对标记用来标识文档的标题,大部分浏览器把其间的 content 显示在窗口的标题栏。在 WAIS 查找服务器时要用到标题,因此在命名时,应力求简短、精炼而又能准确、全面地描述文档的性质和含义,以便通过它使更多的用户能了解文档中的内容,标题一般不应超过 64 字节;

(4) <BODY>以下就是 HTML 文档的主体,通常在这部分定义页(page)的内容,

<BODY>可带一些性质参数(请参见附录一),</BODY>表示主体部分结束。

在 HTML 中,配对的标记(只有极少数标记不是成对出现的)与置于其中的文字所构成的 HTML 语言代码称为该标记的元素(element)。例如,在例 2.1 中,有 title、body 等元素。元素是构成 HTML 文档结构的基本单位。元素的顺序或嵌套使用,构成了 HTML 文档,但 HTML 要求一个文档至少应包含 html、head、title、body 这 4 种元素,即 HTML 文档的基本结构是



2.3 HTML 标记简介

上节我们已经认识了一个 HTML 文档中所应包含的<HTML>、<HEAD>等标记,下面再继续介绍几种有特殊功能的 HTML 标记,以便为后续章节的学习奠定基础。

2.3.1 超文本链接

最重要的 HTML 标记是建立一个超文本的链接,即将超文本链接到 HTML 文档的其他部分或 Internet 上的其他资源,其一般格式为:

<A option[,option]>text

其中,text 会在浏览器上进行显示,[,option]是可选项,即<A>标记可有多个性质参数,如 HREF,NAME 等。

HREF 定义了锚(anchor,<A>标记的“A”即取自其首字母)所链接的对象,例如:

Welcome to 1108

当 Welcome to 1108 被选取后,就会转入到西北工业大学软件研究中心的 home page 中。

也可以用超文本链接在页内移动,首先用性质参数 NAME 定义一个位置,需要时,利用超文本链接可很快跳到 NAME 所定义的位置。例如,在文档起始处给出如下的定义:

则在需要的地方(如文档末尾),可用 HREF 参数作链接:

to the beginning

注意,必须写上“#”。于是,当选中 to the beginning 时,便会很快回到页面的起始处。

2.3.2 内嵌图像

没有图形,WWW 就不会如此生动、漂亮,如此吸引人。在 HTML 中,使用标记把图像嵌入 home page 非常方便,例如,使用标记

```
<IMG SRC="image.gif" ALT="1108">
```

可以把一个名为 image.gif 的图像插入当前位置,自然,此名为 image.gif 的图像文件应在 HTML 文件所在的目录之中。大多数浏览器可以显示 X Bitmap、GIF、JPEG 等格式的内嵌图像。然而,在某些情况下,例如所使用的浏览器不能显示图像或用户嫌加载图像费时,可使用 ALT 性质参数屏蔽图像的显示,而只显示由用户指定的图像替代文本。例如,在本例中,若浏览器不能显示 image.gif 图像,则显示"1108"这一字符串。

2.3.3 接受用户输入的 FORM

FORM 是 HTML 供用户与 Web 服务器进行信息交换的一种手段。如:

```
<FORM METHOD=method ACTION="action">
<INPUT TYPE="type" NAME=name VALUE=value>
</FORM>
```

在<FORM>标记内指定了对用户输入信息的处理方式:method 可以有 GET 和 POST 两种方法。GET 将输入信息转至 action 指定的程序进行处理;POST 将信息发往 action 指定的文件存储,留待后用。

INPUT 标记表示一个用户输入域,用其必选性质参数 TYPE 来区分域的类型,type 可以定义为 text、checkbox、radio、button,等等,分别表示文本框、检查框、单选按钮、按键等,它们可以用来接受用户的输入。INPUT 标记的其他性质参数的选择取决于 TYPE 的值。关于 INPUT 标记和 FORM 标记的具体描述请参见附录一。

2.3.4 加载其他语言的脚本代码

用<SCRIPT></SCRIPT>这对标记可使 JavaScript 程序嵌入到 HTML 文档中去。<SCRIPT>用性质参数 LANGUAGE 来指定所嵌入程序的种类,如欲嵌入 JavaScript 代码,则写成:

```
<SCRIPT LANGUAGE="JavaScript">
.....
</SCRIPT>
```

而欲嵌入的代码则应全部写在该标记之内。

2.3.5 HTML 注释符

在 HTML 中,用<!--和-->把作为注释的字段括起来,以便浏览器在对 HTML 文件作解释时跳过注释部分。因为在 SCRIPT 标记之内用注释符括起来的内容具有特殊意义,将在第三章作具体说明。

2.3.6 格式控制标记

这类标记用于控制文档的显示格式,下面简述它们的功能,具体使用方法参见附录一。