



完全基于互联网丛书



随书赠光盘

本书所用技术:
ASP, PHP, Perl
SQL, Access

网络签证 客户身份验证与管理

摆渡人工作室
编著
www.buydoren.com



机械工业出版社
China Machine Press

完全基于互联网丛书

网络签证——客户身份验证与管理

摆渡人工作室 编著



机械工业出版社

目前，在数字化认证方面主要由两个要素来确定一个用户：用户身份识别号和密码。而在这两个要素当中，密码又是更重要的，因为用户身份识别号码在许多场合都是可公开的，但密码不可公开。这时候，密码本身的安全问题就特别突现了，从而出现了对密码再进行加密存储的办法，即便加密密码被窃取，也不会对密码本身造成安全隐患。

本书的重点不在于告诉读者如何设计一个非常科学的加密算法，而在于如何建立各种应用情况下的认证办法，加密算法只是内容的一部分。从系统级的用户认证到应用程序的用户认证，从简单的用户识别到复杂的网站安全保护，本书内容涉及的范围比较广。书中每一个页面都包含一个或多个认证处理程序，每个程序在某个应用方面都很具有代表性。读者按照书上讲解的步骤实际操作一遍就可以掌握本书的内容。

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

责任编辑：王听讲 余茂祚 封面设计：艾迪艾藤

责任印制：路琳

中国建筑工业出版社密云印刷厂印刷·新华书店北京发行所发行

2001 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·17.5 印张·423 千字

0 001—6000 册

定价：32.00 元（1CD，含配套书）

ISBN 7-900043-89-6/TP·85

E-mail: sbs@mail.machineinfo.gov.cn

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话（010）68993821、68326677-2527

前 言

从 20 世纪 60 年代网络雏形的发展到目前全球化 Internet 的广泛应用, 互联网正逐渐改变我们的生活方式。我国自 20 世纪 90 年代初才开始接入 Internet, 而仅仅过了 10 年左右的时间, 我们周围的生活就显现出了“Internet 化”——从网上资讯浏览、网上购物到网上聊天, 我国网民人数正以日逾万人的速度增长着。

近几年, WWW(World Wide Web)的应用日益成为 Internet 发展中的重点, 不但各种类型的网站应运而生, 而且 WWW 的设计技术也层出不穷。大型 ICP 为了吸引网民的注意力, 不断扩大麾下的个人主页空间, 这为网络爱好者提供了一个可以展示个人风采的大舞台。

因为 WWW 具有易于浏览、可操作性强、使用环境简单、开发容易的特点, 我们可以预见, 未来 WWW 的创意、设计和编程人员将越来越多, 人才需求越来越大, 应用范围也会越来越广, 传统的应用编程也向 WWW 靠拢。

完全基于互联网丛书(Fully Based on Internet, 简称 FBI)主要就是针对 Internet 的发展倾向, 面向广大读者, 从 WWW 的创意、设计和编程的各个方面进行说明。FBI 丛书的主要特点可以用“一个中心, 两个基本点”来概括。

♣ **一个中心——完全基于互联网的发展技术为中心。** 丛书中包括目前几乎所有互联网所涉及到的方方面面, 从网站的整体规划到最细微方面的设计, 囊括了 WWW 领域内时兴的应用点, 如典型的计数器、BBS、聊天室、数据查询与搜索、Web 数据库、各种网页特技、安全控制、电子商务等。换句话说, “网上有什么, FBI 中就讲述什么”。

♣ **两个基本点——便于阅读和便于交流**

(1) **按效果(功能)划分, 方便了读者阅读** 在 FBI 丛书的每一本书中都会向读者讲解几种相关软件的应用技术, 便于读者阅读和应用。为了实现某一种效果(功能), 往往会涉及到很多种应用工具软件, 如在制作网页时, 不但要熟悉 FrontPage、Dreamweaver、Flash、Fireworks、Photoshop 等软件的不同特点, 而且还要学会各种编程技术, 如 Java、ASP(JavaScript/VBScript)、PHP、Perl、SQL/MySQL 等, 这样才能使您制作的网页更加精彩。在同一本书中您可以看到各种不同的软件和语言为了实现同一个效果(功能)协同作用。而采用更有针对性的页面形式编排, 融技术于具体应用, 读者只要按照书中的步骤就可以完成预定的效果(功能)。这是目前市面上仅有的按照效果(功能)划分并以网页形式编排的系列图书。

(2) **完善的售后服务体系, 便于读者与作者之间的交流** 为了方便广大读者与本丛书作者的相互沟通, 我们特为本套丛书建立了专用网站。请记住我们的网址 <http://www.buydoren.com>, 欢迎您指正 FBI 丛书存在的问题, 或者就您感兴趣的问题和我们探讨。我们也将定期对新出现的计算机技术在网上进行实例讲解。

我们都是 Internet 的爱好者, 相信通过我们的共同努力, 可以为互联网在我国的普及尽一点微薄之力。

摆渡人工作室

www.buydoren.com

关于本书

随着计算机网络的发展，人类社会生活的方方面面都出现了很大的变化，数字化生活已经成为趋势，例如金融电子化、电子产品、计算机及网络、企业办公自动化、社区服务等等，尤其是目前热门的电子商务。这些变化使得人们经常要面对身份识别的问题。去银行取钱需要帐户密码、收发电子邮件需要帐户密码、登录网站需要帐户密码、网络商务活动需要密码等等。为了使得这种身份认证能够唯一确定某个实际的用户，就需要在数字认证机制上有好的实现办法。

《网络签证——客户身份验证与管理》是《完全基于互联网丛书》的第十本，主要介绍如何开发安全的网站系统，内容涉及操作系统本身的安全处理和安全认证、Web 站点的安全控制等。

页面 1 介绍如何使用 Cookie 进行网络客户识别；页面 2 通过不同技术、不同方法向读者说明 IP 地址在 Web 身份识别当中的应用；页面 3 主要是就 Windows NT 下的系统安全进行说明，并举一个 VB 开发例子来讲解如何更改 Windows NT 下的帐户密码；页面 4 讲解随机密码的生成及如何把随机生成的密码传给用户；页面 5 为读者开辟了一种思路，即如何采用比较成功的、能够应用的算法来保护自己的数据；页面 6 介绍目前最流行的网站系统中的身份认证；页面 7 在页面 6 的基础上介绍网站个人资料的更改及删除，主要是网站中个人密码的更改；页面 8 本页面为读者讲解 Unix 下的系统认证问题；页面 9 针对目前网络上的页面保护技术，讲解了两个页面保护的例子。

本书通过这九个具体的页面，包括十几个具体的例子，几乎把除了利用人体特征识别用户外的其他网络验证技术全部包含在其中了。读者通过阅读这本书，可以详细地了解到目前网络中应用到的各种认证技术。

编 者

关于我们——摆渡人工作室

“摆渡人工作室”的前身是在两年前成立的“清汉计算机工作室”。

两年多来，我们和机械工业出版社通力合作，精心策划、编辑、出版了《计算机开发与制作实例丛书》（以下简称《实例》）。这套丛书共计 26 本，覆盖了从 Microsoft Office 到应用程序和数据库开发，以及网页制作和多媒体等多方面实用技术和技巧，以其独特的魅力在同类计算机图书中发展成了一个大家族。

我们立足清华大学，广泛了解并研究计算机发展方向，认真进行计算机图书的策划、组织和编辑工作。

我们的努力得到了应有的回报！2000 年我们的实例丛书获得了巨大的成功，得到了读者和出版社的广泛好评。其中《FrontPage 2000 网页制作实例》和《Visual Basic 6.0 数据库开发实例》一年内加印了 4 次，累计销量达到 16000 册，丛书平均销量也达到了近 10000 册。

在实例丛书的具体运作中我们体验了创业的艰辛，成功的喜悦，在成功后我们开始在更深的层次上思考读者真正需要什么。

我们开始思考两个问题：

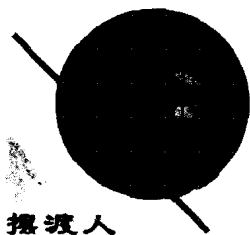
◆ 我们何不推出更具针对性的实例呢？

我们发现实例丛书中的实例虽然是我们精心设计的，但很多却不是“现存的”，也就是说这些实例不是在网上存在的，而是我们为了讲解某些知识专门设计的。这对需要解决实际问题的读者来说还需一个消化吸收的过程。

◆ 何不建立一个更完善的图书售后服务体系呢？

有些读者购买了我们的实例丛书后，针对书中的某些问题或打电话或发 Email 和我们探讨，我们为读者对我们的厚爱感到由衷的高兴，但同时也感到交流的不顺畅阻碍了我们和读者进一步的探讨。读者买到图书，正是学习的开始，我们的工作应该延伸到读者的学习过程中，而不是售出书就万事大吉！

带着这些急需解决的问题，我们考虑了现在最便利的联络方式——互联网，在对现在全球互联网技术现状和未来发展走势的精心调查研究下，我们隆重推出了《完全基于互联网丛书》（Fully Based on Internet，简称 FBI）。这套丛书继承了《计算机开发和制作实例丛书》的特点，以页面的形式讲解当今互联网业涉及的几乎所有的技术。从申请域名建立网站，到实现人机交互和建立自己的电子商务网站等全过程。读者只需要按照我们的讲解，按部就班地学习丛书中介绍的各种技术，不仅可以建立展现自我风采的个人网站，还可以成功地建立完善的商业网站。同时这套丛书还具有“一个中心+两个基本点”的特征。关于这方面的内容您可以从本丛书的前言中看到。相信在您认真阅读了这套丛书后，会对这些特点有更深刻的理解。



随着 FBI 丛书的发展,我们相信一定会与您成为好朋友,因为我们和您一样都是爱好新鲜事物的年轻人,充满着对未来的憧憬。我们努力、创造,我们自信、把握,我们相信机会在自己的手中。

您有什么好的建议或者对我们的丛书有什么意见,请访问我们的网站:

www.buydoren.com。

关注生活, 关注我们的图书!

附录: 关于《实例》丛书

《实例》丛书的特点:

(1) 通过一个个精心设计的实例,用户可以轻松地逐步掌握一种软件。

(2) 书中每一个实例的主题对应该软件的一个主要功能,通过所有实例的学习,您可以全面掌握该软件的精髓。

(3) 虽然《实例》丛书采用一个个相对独立的实例,但我们在丛书的策划中考虑到知识的连贯性和统一性,丛书中的“归纳和注释”可以帮助读者举一反三地掌握书中介绍的内容。使用附光盘的书籍更可以使您的学习和开发如虎添翼。

(4) 随书光盘中还包含和该书籍相关的软件及网络资源,使您在阅读该书的同时还可以涉猎其他相近的领域。

《实例》丛书书目:

1. Word 2000 中文版实例
2. Excel 2000 中文版实例
3. PowerPoint 2000 中文版实例
4. Access 2000 中文版实例
5. Delphi 5.0 开发实例 (附 CD)
6. VBScript 开发实例 (附 CD)
7. ASP 开发实例 (附 CD)
8. C++ Builder 网络开发实例 (附 CD)
9. Flash 5.0 网页动画制作实例 (附 CD)
10. Photoshop 6.0 图像制作实例 (附 CD)
11. FrontPage 2000 网页制作实例 (附 CD)
12. CorelDRAW 9.0 图画制作实例 (附 CD)
13. Photoshop 5.5 图像制作实例 (附 CD)
14. 3D Studio MAX R3.0 动画制作实例 (附 CD)
15. Authorware 5.0 制作实例 (附 CD)
16. Director 7.0 多媒体制作实例 (附 CD)
17. Visual Basic 6.0 数据库开发实例 (附 CD)
18. Visual Basic 6.0 多媒体开发实例 (附 CD)
19. Visual Basic 6.0 网络开发实例 (附 CD)
20. Visual FoxPro 6.0 开发实例 (附 CD)
21. Flash 4.0 动画、网页制作实例 (附 CD)
22. AutoCAD 2000 中文版综合应用实例 (附 CD)
23. Visual C++ 6.0 数据库与网络开发实例 (附 CD)
24. Visual C++ 6.0 多媒体开发实例 (附 CD)
25. Dreamweaver 3.0 网页制作实例 (附 CD)
26. Fireworks 3.0 网页素材制作实例 (附 CD)

目 录

前言

关于本书

关于我们——摆渡人工作室

页面 1	应用 Cookie 识别用户.....	1
页面 2	通过 IP 地址识别用户.....	30
页面 3	NT 帐户密码和系统安全.....	50
页面 4	随机生成密码.....	80
页面 5	Vernum 加密算法.....	90
页面 6	网站系统中的身份认证.....	106
页面 7	网站系统中的密码修改.....	153
页面 8	Unix 帐户密码和系统安全.....	177
页面 9	页面保护处理程序.....	208



<http://cheer.my169.com/js/js-11.htm>

页面效果和技术概要

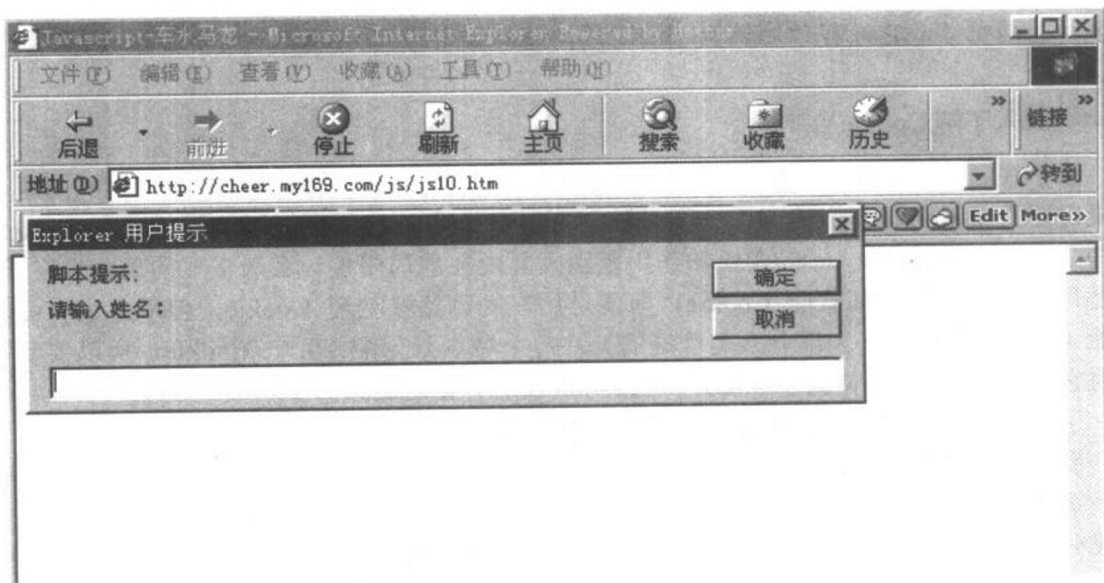


图 1-1 应用 Cookie 进行身份识别

Cookie 由 Web Server 产生，在用户的计算机与 Web 服务器间传输。在下载 HTML 网页时，嵌在其中的 Cookie 文件就写到用户的计算机上。它包含了 Server 所希望放置的任意信息，并且这些信息可以在用户下次登录该网站时被 Server 读回。

对于网络开发者来说，Cookie 能使他们获得有关访问者很多的有用线索，这就有助于开发者确定网络发展战略目标；对于网络用户来说，也可以从 Cookie 当中汲取很多有用的东西。本页面实例将向读者讲述如何使用 Cookie 进行网络客户识别。

实现本页面需要的技术概要包括：

- ① Cookie 的概念及功能
- ② Cookie 的组成及存储机制
- ③ 通过 Web 页面进行 Cookie 的设置和读取
- ④ Cookie 的安全性



实现步骤和技术详解



介绍 Cookie

Cookie 是记录访问读者的站点的人的信息，它其实驻留在用户的硬盘上，即使用户已经离开你的站点，Cookie 在用户的硬盘上仍然存在，如果该用户再次返回你的站点，则该 Cookie 就会被一起发回到你的服务器中，便于统计和处理重复到访者的信息。但是有一点需要注意：Cookies 被引入 Netscape 2.0 和 MSIE 3.0，但是 MSIE 3.0 对 Cookie 的应用不是太完善，可以利用 Cookie 在别的用户的计算机上保存和读取信息，但却不能将 Cookie 保存在自己的计算机上，所以给 Cookie 的测试造成一定的困难。所以如果使用的是 MSIE 3.0，最好将其升级为 MSIE 4.0 或者改用 Netscape。

下面我们看一看一个 Cookie 应用的典型例子，我们在一个网页中设置 Cookie，然后通过别的网页读取它。在使用该例子的时候，想想如果没有 Cookie，如何做到这一点。



深入了解 Cookie

由于 Cookie 牵扯到向用户的硬盘写盘和读取信息，所以就涉及一个保密性的问题。如果需要大量 Cookie，应该阅读一下 Marc Slayton 写的 Cookie 揭密以及重新考察 Cookie。这些文章将告诉你 Cookie 的实质和作用范围及其内在的局限性。其最重要的局限性在于：不是每个人的浏览器都欢迎 Cookie。即便是用户的浏览器欢迎 Cookie，但用户也有可能拒绝 Cookie 的访问（大部分人是欢迎的）。每个域名只分配 20 个 Cookie，所以要节省着用它们。Cookies 不得大于 4 KB，当然 4000B 的容量是足够的了。

了解了这些局限性之后我们开始学习如何设置 Cookie。

设置一个基本的 Cookie 很容易。所需做的只是在一个 `cookie_name=value` 表单中生成一个字符串，然后设置 `document.cookie` 属性。唯一的技巧：Cookie 值中不能有空格、逗号或分号。好在你无需担心这些问题，因为有一系列的函数可以帮你对 Cookie 属性编码和解码：`escape()`和`unescape()`。

下面的简单例子中将你的姓名保存为一个 Cookie：

```
function setCookie()
{
    var the_name = prompt("What's your name?","");
    var the_cookie = "wm_javascript=username:" + escape(the_name);
    document.cookie = the_cookie;
    alert("谢谢，现在转到下一页");
}
```

函数中间的一行是关键：

```
var the_cookie = "wm_javascript=username:" + escape(the_name);
```

如果在提示框中输入了“dave thou”，该行代码将生成一个字符串：

```
wm_javascript=username:dave%20thau.
```

这就是说将把一个名为 `wm_javascript` 的 Cookie 保存到硬盘。该 Cookie 的值是 `username:dave%20thau`。函数 `escape()` 将 “dave” 和 “thau” 之间的空格用 `%20` 做了替换。

当读取 Cookie 时，寻找名为 `wm_javascript` 的 Cookie，提取 “`username:dave%20thau`”，将其用 `unescape()` 解码，去掉 “`username:`”。

```
document.cookie = the_cookie;
```

Cookie 现在就设置好了，很简单。下面我们学习如何读取 Cookie。



如何读取 Cookie

一旦在某用户的硬盘上设置了 Cookie，读取是件很容易的事。下面是读取 Cookie 范例的代码：

```
function readCookie()
{
    var the_cookie = document.cookie;
    var broken_cookie = the_cookie.split(":");
    var the_name = broken_cookie[1];
    var the_name = unescape(the_name);
    alert("你的名字是：" + the_name);
}
```

第 1 行很重要。当你的浏览器打开一个网页时，它调用任何和该网页有关的 Cookie 然后将其载入 `document.cookie` 属性。

读取 Cookie 的技巧在于从中抽取出需要的信息。注意我们所设置的 Cookie 是这样的：`wm_javascript=username:dave%20thau`。在该函数第 1 行之后的所有语句用于从该 Cookie 中提取出用户名（`username`）。

```
var broken_cookie = the_cookie.split(":");
将 Cookie 在分号处分割成两部分。
var the_name = broken_cookie[1];
抓取分号后面的内容 dave%20thau。
var the_name = unescape(the_name);
取消函数 escape() 的编码替换。在本例中重新用空格替换了 %20。
alert("你的名字是：" + the_name);
显示你的姓名。
```

这个例子使用的 Cookie 只保存了很少的信息：用户名，Cookie 最多可以保存多达 4KB 的信息。



复杂的 Cookie 读取

如果想让自己的 Cookie 包含更多的信息，可以将 Cookie 的值设得很长。假设我们要保存某人的姓名，年龄和电话号码：

```
var the_cookie = "username:thau/age:older than the hills/phone:411";
document.cookie="my_happy_cookie="+ escape(the_cookie);
```

用斜杠 / 来分割属性名称，用分号区别不同的属性名称及其属性值。斜杠 / 和分号不是绝对的选择，可以使用任何的字符做分割的标志：

```
var the_cookie = "username=thau&age=older than the hills&phone=411";
document.cookie="my_happy_cookie=" + escape(the_cookie);
```

读者可以自行选择限位器，只要注意在对 Cookie 解码时也使用同样的限位器即可。

设置复杂的 Cookie 时方法要复杂一些。建议使用相关数组来保存所有的信息，假设将该 Cookie 保存到某人的硬盘上：

```
my_happy_cookie=username:thau/age:older than the hills/phone:411
```

可以将这些信息放到一个方便的相关数组中：

```
function readTheCookie(the_info)
{
    // 将 Cookie 存入变量并还原空格
    var the_cookie = document.cookie;
    var the_cookie = unescape(the_cookie);
    // 根据分割符获取数值
    var broken_cookie = the_cookie.split("=");
    var the_values = broken_cookie[1];
    // 分割每个 name:value 数值对
    var separated_values = the_values.split("/");
    // 循环处理每个变量及变量值
    var property_value = "";
    for (loop = 0; loop
```

如果在 JavaScript 中有上面这段代码，可以这样调用它：

```
var cookie_information = new Array();
readTheCookie(cookie_information);
```

然后就会正确设置 cookie_information["username"]、cookie_information ["age"] 和 cookie_information["phone"]。

这些看起来可能有些难以理解，但实际上并不是很难，下面我们一步步分析：

```
var the_cookie = document.cookie;
// 将 Cookie 赋值给一个变量。
var the_cookie = unescape(the_cookie);
// 取消 escape() 的编码
var broken_cookie = the_cookie.split("=");
var the_values = broken_cookie[1];
// 使 the_values 等同于 username:thau/age:older than the hills/phone:411。
var separated_values = the_values.split("/");
// 生成一个包含三个元素名为 separated_values 的数组：
separated_values[0] = "username:thau"
separated_values[1] = "age:older than the hills"
separated_values[2] = "phone:411"
```

```

for (loop = 0; loop
循环调用 separated_values 的三个元素。
property_value = separated_values[loop];
提取当前的 name:value 配对，第 1 个配对是 username:thau。
var broken_info = property_value.split(":");
将该配对分成名为 broken_info 的数组中的两个元素：
broken_info[0] = "username"
broken_info[1] = "thau"
var the_property = broken_info[0];
第 1 次经过这个循环时，the_property 是 "username"
var the_value = broken_info[1];
其值是 "thau"
the_info[the_property] = the_value;
这里开始发回相关数组的便捷功能。它使得 the_info ["username"] = "thau"，所以现在
当需要从 Cookie 中查找 username 时只需：
var the_name = the_info["username"];
每次经过这个循环时，就在 the_info 中加入一个新元素。循环到最后时，
the_info["username"] = "thau", the_info["age"] = "old as the hills"，而 the_info["phone"] =
411。

```

虽然有些烦琐，但是当需要从 Cookie 中输出大量信息时这是一个很好的办法。当然还有别的办法。



读取和编写多重 Cookie

上一步中我们学习了如何将大量的信息包含在一个 Cookie 中，另一种方法是使用多重 Cookies。保存多重 Cookie 的方法很直观，每一个 Cookie 都有一个名称，上一个例子中的 Cookie 的名称是 my_happy_cookie，可以这样操作：

```

var the_cookie = "my_happy_cookie=happiness_and_joy";
document.cookie = the_cookie;
要保存多重 Cookie，只需给每个 Cookie 一个不同的名字。如果读者要加入一个新的
Cookie，设置 document.cookie 时并不会删除前面已经设置了的 Cookies，所以：
var the_cookie = "my_happy_cookie=happiness_and_joy";
document.cookie = the_cookie;
var another_cookie = "my_other_cookie=more_joy_more_happiness";
document.cookie = another_cookie;
现在需要访问这两个 Cookie，有些复杂，所以读者需要明了整个过程。
假设执行了上面的代码，现在想访问 my_happy_cookie。如果查看 document.cookie 的
内容，会看到：

```

```

my_happy_cookie=happiness_and_joy;
her_cookie=more_joy_more_happiness;

```

这样很直观，但是如果想访问某个特定的 Cookie 则有些困难。下面的代码可以帮助找出某个特定的 Cookie：

```
function WM_readCookie(name)
{

//如果没有 Cookie 则返回 false 或者取得值并返回该值
if(document.cookie == "")
    return false;
else
    return
unescape(WM_getCookieValue(name));
}

function WM_getCookieValue(name)
{
// 定义变量
var firstChar,
lastChar;

// 获取完整的 Cookie 字符串
var theBigCookie = document.cookie;

// 寻找具有 name 名称的 cookie 值
firstChar = theBigCookie.indexOf(name);

// 如果找到 Cookie
if(firstChar != -1)
{
    // 去掉 name 和 '='
    firstChar += name.length + 1;

    // 寻找数值串的末尾(例如数值串分割符';')
    lastChar = theBigCookie.indexOf(';', firstChar);

    if(lastChar == -1) lastChar = theBigCookie.length;
    // 返回该 Cookie 值
    return theBigCookie.substring(firstChar, lastChar);
} else
{
    // 如果没有找到相应的 Cookie 则返回失败值
```

```
    return false;
  }
}
```



再次深入了解 Cookie

现在我们已经学会了如何设置和读取基本的 Cookie。然而基本的 Cookie 常常在用户关闭他的浏览器时会被自动删除，有时候这样最好，因为通常的域只允许在用户的机器上保留 20 个 Cookie。但是如果希望将 Cookie 保存在用户的机器上，就需要设置一个 Cookie 失效的时间，它的格式是一种叫做 GMT 的特殊格式，例如：Mon, 27-Apr-1998 00:00:00GMT 要正确设置 GMT 不是一件容易的事，需要计算好某个日期是星期几。好在 Javascript 有一个日期的方法叫做 toGMTString 可以帮助读者。下面是设定远期的某个时间的一个例子：

```
var the_date = new Date("December 31, 2023");
var the_cookie_date = the_date.toGMTString();
```

一旦设置了 Cookie 的失效期，必须在 Cookie 设置的前面加入这条信息。因此完整的 Cookie 应该如下：

```
cookie_name=blah_blah;expires=date
```

通常只在 Cookie 字符串中加入 expires=date，然后用分号分割不同的 Cookie。

下面是一个如何建立有效期直至 Mayan 日历末尾的函数：

```
function setCookie()
{
  // 获取名字
  var the_name = prompt("What's your name?", "");

  var the_date = new Date("December 31, 2023");
  var the_cookie_date = the_date.toGMTString();

  // 构造并保存 Cookie
  var the_cookie = "my_cookie=" + escape(the_name);
  the_cookie = the_cookie + ";expires=" + the_cookie_date;
  document.cookie = the_cookie;
}
```

最后 Cookie 应该如下所示：

```
my_cookie=thau;expires=Fri,
31-Dec-2023 00:00:00 GMT
```

设置好 Cookie 之后，它将在用户的机器中存在直到失效期。如果将某个 Cookie 的失效期设置得比当前时间还早，该 Cookie 实际上不能在用户的机器上存活。



Cookie 路径和域

这是掌握 Cookie 最后的一个障碍：默认情况下 Cookie 只能被在同一个 Web 服务器上同一个路径下设置了该 Cookie 的网页读取。例如，如果在“http://chimp.webmonkey.com/food/bananas/banana_puree.htm”有一段 Javascript 询问了用户的姓名，可能需要在另一个网页中访问一个给定的名字，所以必须设定该 Cookie 的路径，路径“path”用于设置可以读取一个 Cookie 的最顶层的目录。将 Cookie 的路径设置为网页最顶层的目录可以让该目录下的所有网页都能访问该 Cookie。

方法：在 Cookie 中加入“path=/;”。如果只想让“food”目录中的网页可以使用该 Cookie，则加入 path=/food。还有一点：有些网站有许多小的域名，例如网猴可能还在“chimp.webmonkey.com”、“gorilla.webmonkey.com”和“ape.webmonkey.com”域名下有网页。缺省情况下只有“chimp.webmonkey.com”域下的网页可以读取该 Cookie。如果想让“webmonkey.com”下的所有机器都可以读取该 Cookie，必须在 Cookie 中加入：“domain=webmonkey.com”

要将一个 Cookie 设置在“http://chimp.webmonkey.com/food/bananas/ banana_puree.htm”并且让所有网猴的网页都可以利用它，可以这样：

```
function setCookie()
{
    var the_name = prompt("What's your name?", "");
    var the_cookie = "cookie_puss=" + escape(the_name) + ";";
    var the_cookie = the_cookie + "path=/;";
    var the_cookie = the_cookie + "domain=webmonkey.com;";

    document.cookie = the_cookie;
}
```



如何确定客户端是否支持 Cookie 的脚本

先建立一个隐藏域：

```
<FORM>
    <input type="hidden" name="cookieexists" value="false">
</FORM>
```

然后在 html 中的 body 中写如下语句：

```
<body onload="cc()">
然后加上如下脚本：
<script language="JavaScript">
<!--
```

```
function cc()
{
    /* 检查 Cookie*/
```

```

if (document.cookie == "")
{
    alert("请打开 Cookie 选项");
    document.Form1.cookieexists.value = "false"
}
else
{
    document.Form1.cookieexists.value = "true"
}
}

document.cookie = 'killme' + escape('nothing')
// -->
</script>

```



Cookie 技术在 Web 中的应用

在图 1-2 所示的网页中使用了 JavaScript 的 Cookie 技术，并演示和讲述了一个 Cookie 技术的应用示例，但决不会擅自收集和传播任何个人信息。

对于 Cookie 技术，由于在安全性方面的不良利用，使很多人害怕它，其实 Cookie 对建立 Web 站点非常有用。只要用点简单的 JavaScript 或 CGI Script 就可做个 Cookie(甜饼)给用户。要查看到有 Cookie 技术的网页所达到的效果，必须使用支持 Cookie 技术的浏览器软件，如 IE、Navigator 等。IE 将 Cookie 保存在 \$win\cookies\ 目录下，而 Navigator 则将 Cookie 保存在 \$netscape 目录下。

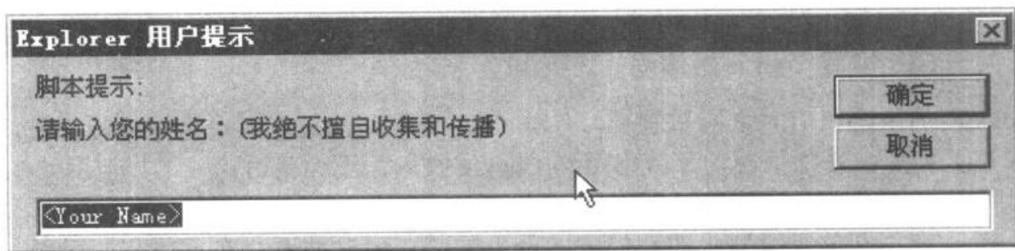


图 1-2 应用 Cookie 技术要求用户输入自己的姓名

在 Web 技术中有一种称为 Cookie 的技术。由于在安全性方面的不良利用，使很多人害怕它，其实 Cookie 对建立 Web 站点非常有用。只要用点简单的 JavaScript 或 CGI Script 就可做个 Cookie 给用户。要查看到有 Cookie 技术的网页所达到的效果，必须使用支持 Cookie 技术的浏览器软件，如 IE、Navigator 等。IE 将 Cookie 保存在 \$win\cookies\ 目录下，而 Navigator 则将 Cookie 保存在 \$netscape 目录下。

在本网页中，演示了一个 Cookie 技术的应用示例。如果你在最近 30 天内第一次访问本网页，该 Cookie 会要求你输入姓名。如果不是第一次访问本网页，网页会自动获得你上一次输入的姓名。从而提供了一种“宾至如归”的服务，希望你能满意！页面如图 1-3 所