



附光盘

JAVA

范例实战

数位文化 编著

```
HTML document ready http://www.tsinghua.edu.cn/
<SCRIPT language=JavaScript>
</>
if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else
document.write("frame col width 125 * border=0 frameborder=1")
document.write("frame src http://www.tsinghua.edu.cn/scrolling.htm noresize name=side merg
if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else
document.write("frame col width 125 * border=0 frameborder=1")
</SCRIPT>
<FRAME name="top" src="http://www.tsinghua.edu.cn/scrolling.htm"
src="http://www.tsinghua.edu.cn/scrolling.htm"
src="http://www.tsinghua.edu.cn/scrolling.htm"
HTML document ready http://www.tsinghua.edu.cn/
<SCRIPT language=JavaScript>
</>
if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else
document.write("frame col width 125 * border=0 frameborder=1")
document.write("frame src http://www.tsinghua.edu.cn/scrolling.htm noresize name=side merg
if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else if (navigator.userAgent.indexOf("MSIE")
document.write("frame col width 125 * border=0 frameborder=1")
else
document.write("frame col width 125 * border=0 frameborder=1")
</SCRIPT>
```



清华大学出版社



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

JAVA 范例实战

数位文化 编著

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>
上由“馆藏检索”该书详细信息后下载，
也可到视听部复制

清华大学出版社

(京)新登字 158 号

北京市版权局著作权合同登记号：01-2001- 2265 号

本书繁体字版由第三波资讯股份有限公司出版，版权归第三波资讯股份有限公司所有。本书简体字中文版由第三波资讯股份有限公司授权清华大学出版社出版。专有出版权属清华大学出版社所有，未经本书原版出版者和本书出版者的书面许可，任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部。

内 容 简 介

本书是一本专门为已经具备 Java 程序语言基础的读者所设计的范例学习指南，通过现成的范例，从 Java Applet 程序的解析、程序设计重点与技巧分析等层面来切入，直接从实例中吸取设计的精华，充实自己的程序设计经验。

文中范例程序语法公开，毫不保留；通过程序流程图来剖析程序，帮助读者彻底了解 Applet 的设计方法，并结合程序源代码展示、解释，轻松地掌握 Applet 程序在网络上的运用，能让你拥有更丰富的设计经验与技巧。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

书 名：JAVA 范例实战

作 者：数位文化 编著

责任编辑：陈仕云

出 版 者：清华大学出版社(北京清华大学学研大厦，邮编 100084)

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

印 刷 者：国防工业出版社印刷厂

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 **印张：**25 **字数：**605 千字

版 次：2001 年 8 月第 1 版 2001 年 8 月第 1 次印刷

书 号：ISBN 7-900635-86-6

印 数：0001~5000

定 价：39.00 元(附光盘)

第 1 章 初尝香醇的 Java 咖啡

Java 是由美国的 Sun Microsystems 公司发展出来的一套面向对象程序语言 (Object Oriented Programming Language), 因为它的面向对象、分散运算、跨平台等等的特性, 语言本身又支持了多线程 (Multi-Thread), 再加上它简单易学的特性, 使得许多用过的人都爱不释手。

随着 Internet 的发展, Java 因为能设计出在网页中执行的嵌入式程序 (也就是本书的重点 Applet), 于 1995 年 Sun 公司发表之后, 在很短的时间之内, 就把因特网变得充满朝气活力, 竞相在平面网页上争艳群芳。许多公司, 包括了当时浏览器使用率最高的 Netscape, 还有 IBM、微软等大公司, 都迫不及待地想要啜饮这一杯香醇可口的咖啡 (Java 也是一种咖啡的名称)。

Java 的出现, 其玲珑婀娜的身影, 惹得原本一片静态 HTML 格式的万维网沸沸腾腾起来, 同时也为网络世界注入了一股新的生命力。

1.1 带领网络前进的尖兵——Java

因为 Java 的魅力, 和其在因特网上的成功发展, 使得愈来愈多的人想要接触 Java 这个香醇可口的咖啡。而 Java 在 Sun 公司不断研发的过程中, 也由当初只是一套新的程序语言, 成为现在各种产品的完整解决方案, 小到在 PDA (便携式信息终端)、手机、Set-top box 等小型运算设备上的使用, 大到整个商业环境、分布式运算环境, 它都提供了相当完善的程序设计空间, 让程序设计师针对每项产品的不同特性努力研发。它为什么会散发出如此大的魅力呢? 让我们来认识一下它。

1.1.1 什么是 Java?

Java 的故事开端其实只在 20 世纪 90 年代初期, 那时候 Sun 公司的工程师们为了要发展出一套可以在各种消费性的电子产品 (就是我们家里的电视、电饭锅、电冰箱等) 使用的标准 (共享的作业平台), 秘密进行一些计划, 中间当然是一波三折啦。在开发这个标准系统的过程中, 他们首先想到选择一套他们熟悉的语言来使用, 在那个年代里, 程序设计师使用的程序设计语言语大部分都是 C/C++, 可是后来却发现 C/C++ 在结构上仍嫌不足, 所以他们决定开发新的一种语言, 取名为 Oak (橡树)。因为 Oak 这个名字当时已经有别的产品拿去用了, 所以他们就在某天到咖啡屋去喝咖啡的时候, 灵机一动, 想到了出

产颇负盛名的咖啡的印尼爪哇岛 (Java)，所以就把这个 Oak 改名为 Java!

但是读者可能会觉得奇怪，既然这是开发使用在那些电子商品上的东西，和 Internet 又有什么关系呢？其实因为在发展的过程中，曾经有一度认为这个点子不能为 Sun 公司赚钱，打算要停止这项计划，工程师们为了要挽救它，想到了利用 Java 结构上的优点，以当时方兴未艾的 Internet 为目标，来开发一个能在网络上执行的环境——一个浏览器，这个点子让 Java 能在网页中执行，这不是很棒吗？的确，正因为这样，使得那时候的网页，从平淡无奇的静态展示，拥有了各种不同的可能和想像空间，随着设计者天马行空的创意，让网页翩翩起舞。这种可以嵌入在网页中执行的小程序，它的名字就叫做 Java Applet，因为如此不凡的吸引力，使得几乎每个网页制作者，都想认识 Java，各方人士自然而然也都开始注意起 Java 来，在逐步的探索中慢慢发现了 Java 的好处与美感。一旦知名度打开了，这股 Java 热就愈烧愈大了。

1.1.2 Java 的优点

在日常生活中，有一位值得相交熟悉的朋友，想必是他身上拥有某部分我们自己缺乏的优点。要介绍读者认识这个网页程序语言——Java，当然也是因为它有以下几个优点：

- ◆ 简单易写。
- ◆ 分布式运算环境。
- ◆ 跨平台。
- ◆ 程序安全性佳。
- ◆ 程序执行效能高。

1. 简单易写

Java 是由 C++ 修改而来的，而且把 C++ 比较复杂的部分都帮我们处理掉了，例如 C/C++ 令人头痛的指针，Java 会帮助你解决这个烦恼。在 C++ 中每次声明一个指针就得指定一块内存，但是在用完后还要记得把它 delete 归还给系统；Java 就聪明多了，内存问题可以交给它的 Garbage Collection 机制管理，设计师只管专心写程序，等到内存不用时它就会主动收回。而且它还提供了一套 Class Library 可以让你不用重复写一些相同的数据结构或程序，节省了很多时间。尤其它支持面向对象的许多特性，更让设计师在开发程序时，特别是大型程序，可以写起来更加方便轻松。

2. 分布式运算环境

Java 设计之初就已经是网络兴起之时，在设计方面当然不会忘记将网络这项因素给考虑进去，所以若要发展分布式的程序，Java 是很好的选择。

3. 跨平台

Java 程序编译后，是产生一种 bytecode，而不是机器码，所以，不论到哪一种结构的

机器上，只要有 Java Virtual Machine，都不需要经过重新编译，就可以直接执行。即使是不同的机器、不同的操作系统，它都帮你解决了这些中间的差异性，让同一个程序可以在 X window、MS Windows、Macintosh 各种操作平台上执行无误。

4. 程序安全性

Java 既然已经处理掉 C/C++ 的指针，也就不必担心指针指错而造成程序无法执行，甚至死机的大灾难，而它的数组也成为一种对象，设计师不用再担心数组驻标超出范围，造成无可挽救的错误。

5. 程序执行效能

Java 除了直译执行之外，其实也可以将 bytecode 转成机器码，让它具有不逊色于 C++ 程序的效能。另一方面，因为线程已经成为 Java 语言内置的对象，需要写多任务程序时可以更加方便——因为有了多线程的程序，可以把界面和运算部分交给不同的线程执行，如此一来，运算的工作就不会完全占用 CPU 资源，让使用者呆望着一动不动、看起来好像死机的程序，完全束手无策，不知道该如何是好。

1.1.3 执行 Java 的工作环境

原来 Java 有如此令人称赞的优点，真是应该来实际操作一番，否则很难体会到它的神妙之处。作文书处理有 Word，写网页靠 FrontPage 或 Dreamweaver，那么需要什么样的环境才能让我们发展出活蹦乱跳的 Java Applet（或者其他的 Java 程序）呢？

- ◆ JDK 编译程序。
- ◆ 文字编辑器。
- ◆ 稳定的操作系统。

1. JDK 编译程序

JDK (Java Development Kit)：这里面包含了编译 Java 程序的 Compiler、执行 Java 程序的 Java Virtual Machine，还有发展 Java 程序时会用到的一些小工具和所需要的 Library 等等。读者若有需要可以在国内许多大学的 FTP 站里找到，也可以直接到 Sun 公司的网站下载最新版的 JDK。本书中采用的是 JDK 1.3.0_01 版本。以下是几个可提供相关服务的网址。

<http://java.sun.com/j2se/>

<ftp://ftp.csie.nctu.edu.tw/pub/Language/>

<ftp://ftp.ice.ntnu.edu.tw/program/java/>

2. 普通的编辑器

利用 Windows 内建的“记事本”，或是“Ultra Editor”等都可以拿来修改或撰写 Java

程序。

3. 操作系统

在 MS-Windows、Un*x、Macintosh 等各种不同的平台上都可以支持，但是你必须下载安装不同平台的 JDK，并且可能根据各平台的不同需要会有一些不同的设置，这一部分请参考 JDK 下载时所附的安装说明。

若计算机操作系统是属于 MS-Windows95/98 的话，安装完毕后，只要记得把 JDKx.x.x\Bin 的路径（这个路径要根据你安装 JDK 的路径而定）加入你的系统变量 Path 里面就可以了。

1.2 嗨！Java！（和 Java 的第一次接触）

现在我们就来写一个最简单的 Java Applet 程序给大家看看吧。这个程序非常简单，只在 Applet 上显示出一个字符串，Hello, Java!，但是这个基本的程序可以让我们看出 Java Applet 程序的基本结构。

1.2.1 初次见面请多指教

这个程序只是很单纯的取得网页里面给的参数，若是网页里面没有设定的话，那么就用我们程序里面设定好的字符串，来显示在 Applet 上。在这个程序里面我们可以看到，Java 程序能执行与否只需要在 `init()` 里面把一些变量设定好之后，浏览器就会按照需要调用 `paint()` 这个 method（方法），来画出我们想要在 Applet 上呈现的东西了。当然，目前这个只是最基本的形式，在接下来的几个章节里面，将会根据需要用到线程，并且让我们的 Applet 看起来更加多姿多彩，如图 1-1 所示 Hello Java 的执行结果。

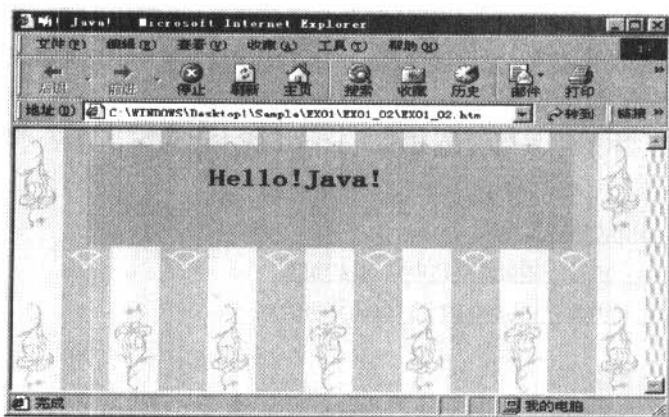


图 1-1 Hello Java 的执行结果

程序代码展示

```
\Sample\EX01\EX01_02\Hello.java

10 import java.applet.Applet;
20 import java.awt.* ;
30
40 public class Hello extends Applet
50 {
60
70     String showText ;
80
90     public void init( )
100    {
110
120        showText = getParameter("ShowText") ;
130        if ( showText == null ) showText = "Hello ! Java!" ;
140
150    }
160
170    public void paint(Graphics g)
180    {
190
200        g.setFont(new Font("Serif", Font.BOLD, 24));
210        g.drawString(showText, 100, 40) ;
220
230    }
240
250 }
```

程序代码解析

- 90~150: 这里是只要浏览器加载 Java Applet 程序之后就会调用的 `method`, 所以通常我们把初始化变量, 或者一些必要的准备工作, 都把程序代码写在这里。
- 120: 这里的 `getParameter()` 是取得在网页上所给定的参数, 请参考相对应的 `Html` 文件。
- 170~230: 这里是每当我们的 Applet 需要被重绘时, 例如这个窗口被其他窗口盖住后, 又回到这里时, 或者是 Applet 加载完成后, 浏览器也会立即调用这个函数, 所以我们把画图的工作集中在这里。

1.2.2 编译 Java 的程序

编译这个程序非常简单，只要在 DOS 窗口（如果你使用的是 Windows 的话，则必须开启 DOS 窗口）输入：（但是必须要注意字母的大小写）

```
javac -target 1.1 Hello.java
```

如图 1-2 所示。

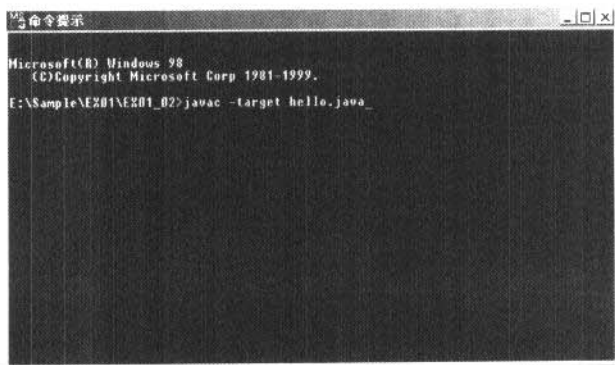


图 1-2 编译 Java 程序

其中 `javac` 就是 Java 的编译器（java compiler），而 `Hello.java` 就是我们写的程序，这里必须要注意的是，Java 程序的文件名必须要和里面的 `class` 名称一样（包括大小写），否则等一下会不能运行。另外一个 `-target 1.1` 这个参数也是必要的，因为日前的 IE 浏览器只支持到 Java 1.1 版，所以，若是程序内不加上这个参数的话，那么我们的 Applet 将不能在浏览器上执行。

1.2.3 执行 Java 的程序

执行 Java Applet 程序有两种方法，一种是直接开启相对应的 `html` 网页，那么马上就能看到了。

另外一种方法是利用 JDK 里面所附的 `appletviewer`，这也是和 `javac` 一样，是一个在命令行（Command Line）激活执行的程序，它是为了方便开发 Applet 所设计的一个工具。以这个 Applet 为例，直接在命令列中输入：

```
appletviewer EX01_02.htm
```

接下来所看到的，它就会跳出一个窗口显示程序的执行结果了，如图 1-3 所示。

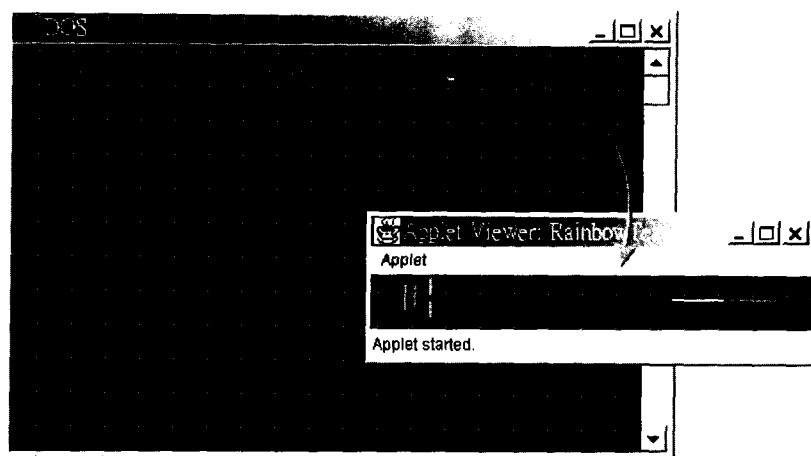


图 1-3 视窗显示程序执行结果

1.3 本书注意事项

本书的程序都是在 Windows 操作系统下，用 JDK 1.3.0_01 编译测试过的，读者的计算机中若已经安装了 JDK，那就只要把光盘里面所附的 Sample 目录复制到硬盘里面，然后依照前一小节所提到的方法编译执行即可。

本书的介绍方式是以 Applet 种类来安排的，所以各章节之间并没有一定的阅读顺序，读者可以选择自己感兴趣的章节先开始阅读。而且每个章节的编排方式是先简介一下各种特效的做法，接着会有程序的展示，然后是程序的解析，最后则会提出一些各个程序需要注意的地方，或者是可以由读者加以改写扩充的地方。

最后，希望能给读者们在了解 Java Applet 程序部分有相当的帮助，以提高程序设计的功力。

第2章 跑马灯

跑马灯是许多网页上经常可以见到的特效，通常都是用来显示一些信息给浏览网页的使用者观看，虽然有点像跑龙套的角色，但是设计精美的跑马灯，往往能为网页带来一股活泼的气息，给观者留下深刻的印象。

2.1 可暂停的跑马灯

一般的跑马灯，所要传达的文字信息，都是一溜烟似地跑个没完没了，有时候真是得凝神观看方才明白上面到底写了什么东西，挺累人的！本节范例将介绍的跑马灯，就比较懂得体贴人，若鼠标在跑马灯上单击一下，则可暂停跑马灯的移动，再单击一下可以恢复跑马灯的动作，如图 2-1 所示。



可以暂停的跑马灯

图 2-1 可以暂停的跑马灯

跑马灯的暂停与恢复执行的激活，都是因为鼠标的单击动作，所以在这个程序中，可以看到如何拦截 / 处理鼠标事件。另外，由于这个程序是以线程的方式运作，因此暂停与恢复都是采用直接调用线程的方法。

程序流程图如图 2-2 所示。

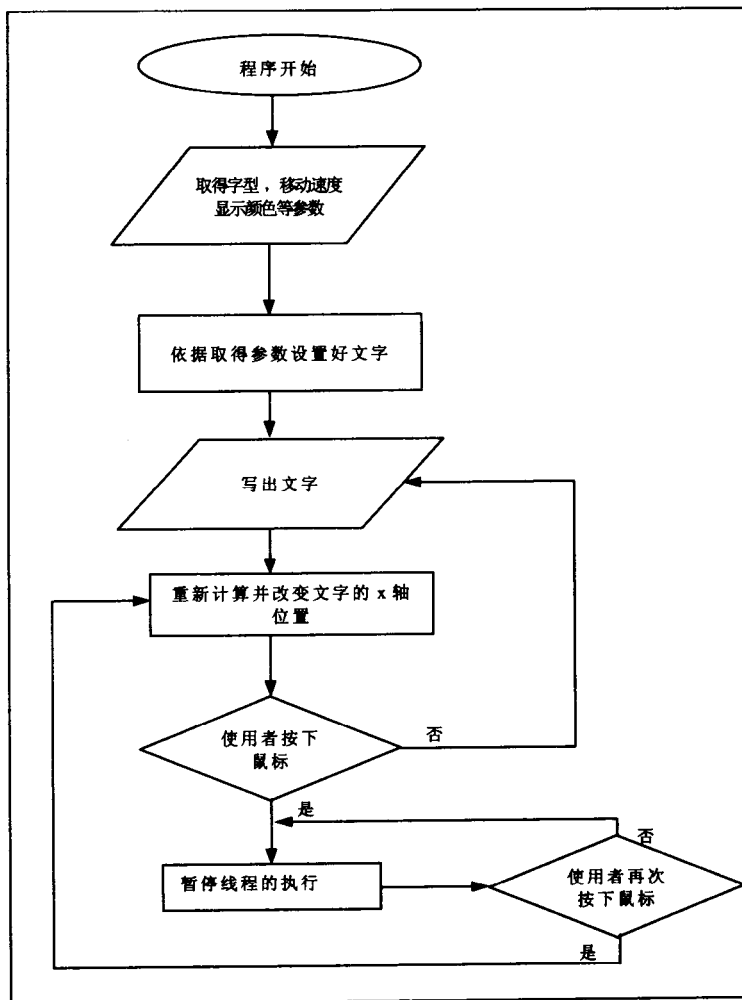


图 2-2 程序流程图

程序代码展示

Sample\EX02\EX02_01\Scroll_1.java

```
10 import java.applet.Applet;  
20 import java.awt.* ;  
30 import java.awt.event.*;  
40  
50 public class Scroll_1 extends Applet implements Runnable  
60 {  
70  
80     String text ;
```

```
90     int xpos, realLength, realHeight, width, height, velocity ;
100    boolean threadSuspended = false ;
110    Font font ;
120    Color fgcolor, bgcolor ;
130    Thread mover ;
140    Image offScreen ;
150    Graphics osGraphics ;
160
170    public Scroll_1( )
180    {
190
200        xpos = 400;
210        width = 400;
220        height = 40;
230        velocity = 5;
240        threadSuspended = false;
250
260    }
270
280    public void init( )
290    {
300
310        String str = getParameter("text") ;
320        text = (str != null) ? str : "No text" ;
330        str = getParameter("fontsize") ;
340        int fontSize = str == null ? 24 : Integer.parseInt(str) ;
350        str = getParameter("fontstyle") ;
360    int fontStyle = Font.PLAIN ;
370        if (str != null)
380        {
390
400            str.toUpperCase( ) ;
410            if (str.compareTo("ITALIC") == 0)
420                fontStyle = Font.ITALIC ;
430            else if (str.compareTo("BOLD") == 0)
440                fontStyle = Font.BOLD ;
450
460        }
470
480    font = new Font("Serif", fontStyle, fontSize) ;
490        str = getParameter("velocity") ;
500        velocity = str == null ? 5 : Integer.parseInt(str) ;
```

```
510     str = getParameter("fgcolor") ;
520     int c = str == null ? 0 : Integer.parseInt(str, 16) ;
530     fgcolor = new Color(c) ;
540     str = getParameter("bgcolor") ;
550     c = str == null ? 0xffffffff : Integer.parseInt(str, 16) ;
560     bgcolor = new Color(c) ;
570     width = getSize( ).width ;
580     height = getSize( ).height ;
590     offScreen = createImage(width, height) ;
600
610 }
620
630
640 private void paintText(Graphics g)
650 {
660
670     g.setColor(bgcolor) ;
680     g.fillRect(0, 0, width, height) ;
690     g.clipRect(0, 0, width, height) ;
700     g.setFont(font) ;
710     g.setColor(fgcolor) ;
720     FontMetrics fontmetrics = g.getFontMetrics( ) ;
730     realLength = fontmetrics.stringWidth(text) ;
740     realHeight = fontmetrics.getHeight( ) ;
750     g.drawString(text, xpos, (height + realHeight) / 2) ;
760
770 }
780
790 public void paint(Graphics g)
800 {
810
820     paintText(offScreen.getGraphics( )) ;
830     g.drawImage(offScreen, 0, 0, null) ;
840
850 }
860
870 public void start( )
880 {
890
900     if(mover == null)
910     {
920
```

```
930         mover = new Thread(this) ;
940         mover.start( ) ;
950
960     }
970
980 }
990
1000 public void stop( )
1010 {
1020
1030     mover = null;
1040
1050 }
1060
1070 private void setcoord( )
1080 {
1090
1100     xpos = xpos - velocity;
1110     if(xpos < -realLength) xpos = width;
1120
1130 }
1140
1150 public void run( )
1160 {
1170
1180     while(mover != null)
1190     {
1200
1210         try
1220         {
1230
1240             Thread.currentThread( ).sleep(10L);
1250
1260         }
1270         catch(InterruptedException _ex) { }
1280         setcoord( ) ;
1290         repaint( ) ;
1300
1310     }
1320
1330 }
1340
```

```
1350     public void update(Graphics g)
1360     {
1370
1380         paint(g) ;
1390
1400     }
1410
1420     public boolean handleEvent(Event event)
1430     {
1440
1450         if(event.id == Event.MOUSE_DOWN)
1460         {
1470
1480             if(threadSuspended)
1490                 mover.resume( ) ;
1500             else
1510                 mover.suspend( ) ;
1520             threadSuspended = !threadSuspended;
1530
1540         }
1550         return true;
1560
1570     }
1580
1590 }
```

程序代码解析

- 50: 实现 Runnable 界面, 使 Scroll_1 可以由一个线程来执行。
- 240~490: 用 `getParameter()` 来取得各个需要的参数, 当如果无法取得这个参数 (因为在网页里没有以 `<param>` 卷标给定参数) 时, 就自动给予程序一个默认值。
- 260~270: 取得要在跑马灯上显示的文字, 若是没有设置则显示 No text。
- 280~290: 取得字号, 若是没有设置则大小设为 24。
- 300~360: 取得字型样式是一般、斜体或粗体字。
- 400~450: 设置前景和背景颜色, 这两个参数读入的字符串以 16 进位来表示, 所以使用 `Integer.parseInt(str, 16)` 来把它转换为整数。例如给定 “0000FF” 这个字符串, 则转换后的数字会代表蓝色; 若是 “FF0000” 则代表红色; 若是 “00FF00” 则代表绿色, 依此类推。
- 480: 准备一块图像, 作为背景画布, 然后将要画出来的东西都先画在这块画布上, 再一次显示到画面上, 以避免直接在画面上做画会造成闪烁的情形。这就是所谓的双缓冲的技巧。
- 590~620: 用 `FontMatrix` 取得这个字符串的大小, 把它画在垂直居中的位置。

- 670~680: 先画在背景画布上, 再把背景画布直接画到屏幕上。
- 910~1040: 这个是让线程做事最主要的地方, 先以 `Thread.currentThread().sleep(10L)`, 让这个线程停止 10 微秒, 然后再以 `setcoord()` 把现在的坐标往左移, 并以 `repaint()` 更新画面。
- 1060~1090: 重写 `update()`, 使它直接更新, 而不是先清掉画面再绘制, 如此可以避免画面的闪烁。
- 1110~1220: 处理当鼠标单击时, 若本来是字在跑动则让线程停止, 则字也会停止, 若再一次单击鼠标, 再让线程继续开始 (`resume()`) 使文字跑动。

本范例程序利用捕捉“鼠标事件”的方式, 设计可以暂停的跑马灯, 其中暂停的方法, 是利用中断线程的方式来完成的。此外, 当用户再次单击 Applet 则多线程会自动再度激活, 恢复跑马灯的进行。简而言之, 这个特效是通过多线程的暂停与否, 使跑马灯产生行进和暂停的效果。

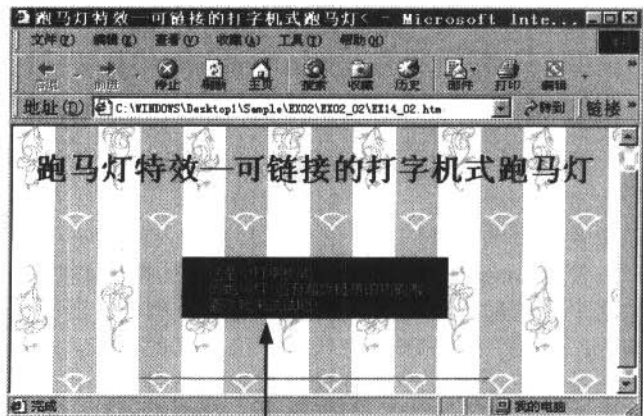
2.2 可链接的打字机式跑马灯

这个小节我们把要显示的文字, 每次只显示一个字, 并且配上了像打字机一样“滴滴答答”的声音。如此一来, 这个跑马灯便会像打字机一样显示出文字, 如图 2-3 所示。

因为需要把想要显示的文字一个个地显示, 而且要配合 Applet 的长度来分行, 所以把所取得的文字做分行处理是很重要的。这个程序利用 `StringTokenizer` 对象来切割读入的文字, 这个对象有非常强大的功能, 它的建立方式如下:

```
new StringTokenizer(要切割的字符串, 切割的字符字符串);
```

后面切割字符的字符串可以由使用者指定, 我们要用什么字符来切割字符串, 可以是“`\n`”这个分行符号, 或是其他的字符, 当然也可以一次指定多个字符在一个字符串中。



像打字机般显示出所有的文字

图 2-3 可链接的打字机式跑马灯