

Flash
ActionScript 3.0
实训课堂

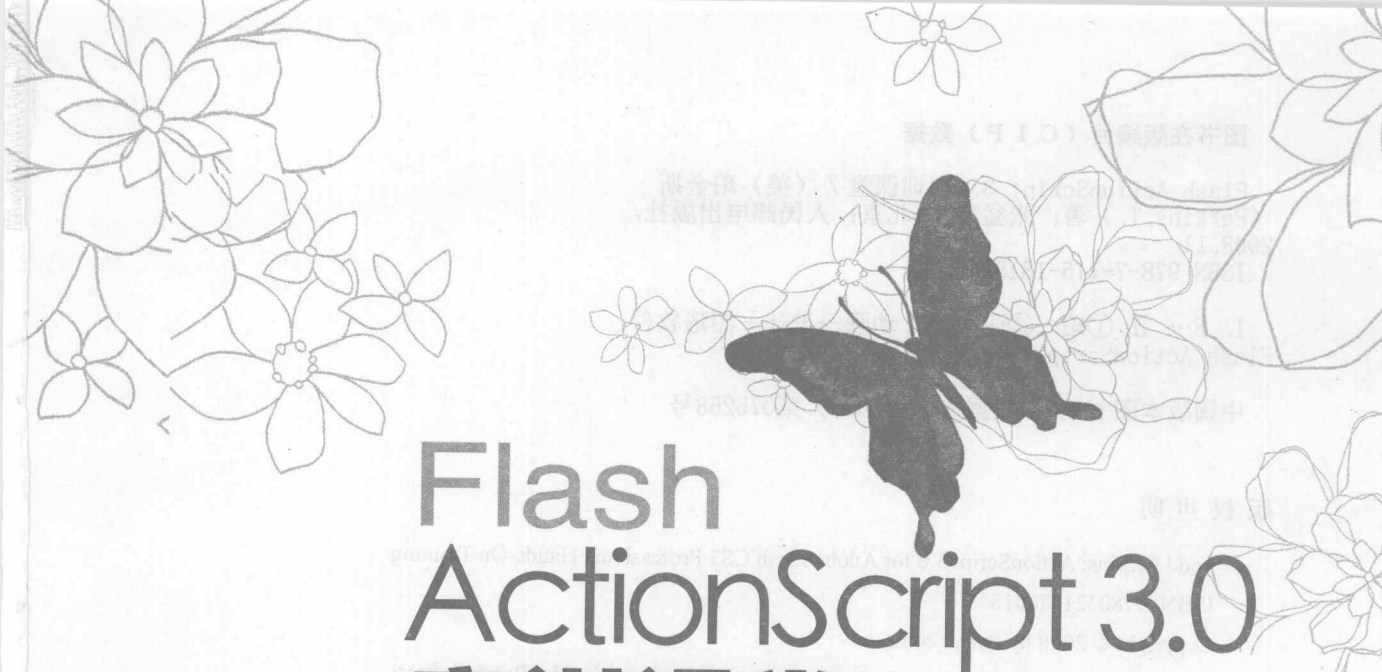
ActionScript 3.0
for Adobe® Flash® CS3 Professional

[美] Todd Perkins 著
张猛 译

源自美国著名培训机构Lynda.com的经典教程；
从实际课堂教学中精选出大量实例，一步步分解剖析；
“手把手”教您快速掌握Flash Actionscript！



人民邮电出版社
POSTS & TELECOM PRESS

A decorative illustration featuring a large black butterfly in the center, surrounded by various line-art flowers and leaves. The background is a light gray with faint, repeating text patterns.

Flash ActionScript 3.0 实训课堂

[美] Todd Perkins 著
张猛 译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

Flash ActionScript 3.0实训课堂 / (美) 珀金斯
(Perkins, T.) 著; 张猛译. —北京: 人民邮电出版社,
2008. 11
ISBN 978-7-115-18185-5

I. F… II. ①珀…②张… III. 动画—设计—图形软件,
Flash ActionScript 3—教材 IV. TP391.41

中国版本图书馆CIP数据核字 (2008) 第075258号

版 权 声 明

Todd Perkins: ActionScript 3.0 for Adobe Flash CS3 Professional Hands-On Training

ISBN: 9780321376015

Copyright © 2008 by lynda. com.

Authorized translation from the English language edition published by Peachpit Press.

All rights reserved.

本书中文简体字版由美国 Peachpit Press 出版公司授权人民邮电出版社出版。未经出版者书面许可,
对本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

Flash ActionScript 3.0 实训课堂

- ◆ 著 [美] Todd Perkins
- 译 张 猛
- 责任编辑 李 际
- 执行编辑 付 飞
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
- ◆ 开本: 800×1000 1/16
印张: 22
字数: 524 千字 2008 年 11 月第 1 版
印数: 1—4 000 册 2008 年 11 月北京第 1 次印刷

著作权合同登记号 图字: 01-2007-4233 号

ISBN 978-7-115-18185-5/TP

定价: 45.00 元 (附光盘)

读者服务热线: (010)67132705 印装质量热线: (010)67129223

反盗版热线: (010)67171154

内 容 提 要

Flash 是目前应用最广泛的一款动画制作软件, Adobe Flash CS3 是其最新版本, 而该软件中内置的 ActionScript 3.0 则是动画实现中的出色编程语言。本书通过实际的示例介绍了该编程语言的基本功能。全书共 12 章, 依次介绍了函数、事件、类等基本概念, 各种判断语句、数学函数、文本和数组, 最后还介绍了高级图形和动画工具, 以及如何处理多媒体和高级交互性。

本书图文并茂, 采用手把手的教学模式, 并用实际的游戏作为示例来综合运用各种知识, 是 ActionScript 初中级读者的必备书籍。

Lynda Weinman 的说明

多数人买计算机书是为了学习，但让人吃惊的是，很少有计算机书是由教师编写的。Todd Perkins 和我可以自豪地说：本书是由有丰富培训经验的资深教师编写的，在本书中，那些精心开发的课程和练习可帮助您学习 Adobe Flash CS3 Professional（最强大、最流行的 Web 动画制作和设计工具之一）中的 ActionScript 3.0。

本书针对的读者是具有初中级水平的 Web 设计师和 Web 开发人员，这些人员需要有合适的工具为 Web 创建强大的、有吸引力的，并且具有很高交互性的内容。采用手把手的教学方法，目的是要让读者在积极参与本书课程的同时迅速掌握 ActionScript。阅读程序代码是一回事，而尝试开发产品并得到有用的结果则完全是另一种体验。我们的座右铭是：“读书、练习，然后掌握程序。”我曾经收到过无数封感谢信，对于这套手把手系列培训图书，我们期望它也能实现同样的效果。

本书并不准备覆盖 ActionScript 的所有方面，也不试图讲授这一极为强大的编程语言所能做的每件事。我们觉得书架上所缺少的是——一本面向过程的教材，一本能够用手把手培训的方式将核心原则、技术和技巧教给读者的教材。

Todd Perkins 和我希望本书能够提高您的 ActionScript 技能。如果真能做到这点，那我们就为所做的工作感到欣慰！

—Lynda Weinman

关于作者

Todd Perkins 是 Adobe 认证讲师，大部分时间他都在培训人们如何使用 Adobe 的 Web 开发软件。Todd 有多年教授不同年龄、不同背景人员的经验，非常擅长用不同的人都能理解的方式讲授复杂的概念。Todd 是

“Adobe 大全：Chad 和 Todd 播客”动态二人组的一员，他已经编写了许多视频培训教程。Todd 喜欢课堂教学，为企业提供咨询服务以及在线进行人员培训，但他最喜爱的还是与他迷人的妻子 Jessica 一起玩视频游戏。

Todd Perkins 的致谢

我要感谢所有帮助本书得以出版的人们。我的朋友 Chad，感谢你将我介绍给 lynda.com 这些可爱的家伙们，感谢你鼓励我学习 Flash。Tanya 和 Lauren，感谢你们的辛勤工作。你们两个人太棒了！Jessica，感谢你在

我编写本书期间给予我的所有支持，感谢你激励我努力工作。感谢 lynda.com 的所有人，我非常喜欢与你们一起工作！最后，我要感谢本书的读者。

如何使用本书

下面介绍的重要信息可以帮助读者更好地使用本书。

本书的内容组成

本书有几个组成部分，包括逐步进行的练习、注释、说明、提示、警告，其中逐步练习有数字编号。

软件界面的屏幕截图

本书的屏幕截图全部来自使用 Windows XP 的计算机。虽然许多 Flash 作者也使用苹果的 Mac 操作系统，但国内的作者使用更多的是 Windows XP 操作系统，本书翻译时也取消了 Mac 机上的操作，希望不会给 Mac 机用户带来不便。

配套光盘内容

在配套光盘上提供了练习文件等有用的资源。

练习文件

配套光盘中为完成练习所需的文件放在名

为 exercise_files 的文件夹内。这些文件按章放在各个文件夹内。开始某章的练习之前，应该将这一章的文件夹复制到桌面上。例如，如果准备开始第 5 章，请将 exercise_files 文件夹中的 chap_05 文件夹从配套光盘复制到桌面。

在 Windows 上, 如果文件来自光盘, 那么该文件会自动变为写保护的, 这意味着不

能修改它们。幸运的是, 更改这个属性很容易。关于完整的操作说明, 请参阅下面的内容。

让练习文件在 Windows 计算机上可编辑

默认情况下, 将光盘上的文件复制到 Windows 计算机上后, 文件设为只读的 (写保护), 因此给使用练习文件带来问题, 因为需要编辑和保存这些文件。您可以按以下步骤操作, 删除练习文件的只读属性。

- 1 打开配套光盘上的 `exercise_files` 文件夹, 将某个子文件夹 (如 `chap_02`) 复制到桌面。
 - 2 打开已复制到桌面的 `chap_02` 文件夹, 选择编辑 > 全部选中。
 - 3 右键单击某个选中的文件, 在上下文菜单中选择属性。
 - 4 在属性对话框中, 选择常规选项卡, 取消选中只读选项, 这样可禁用 `chap_02` 文件夹中所选文件的只读属性。
-

让文件扩展名在 Windows 计算机上可见

默认情况下, 在 Windows 计算机上是看不到文件扩展名的, 如 `.htm`、`.fla`、`.swf`、`.jpg`、`.gif` 或 `.psd`。幸运的是, 可以轻松地改变此设置。

- 1 在桌面上, 双击“我的电脑”图标。
注意: 如果有人改变了我的电脑图标的名称, 那么可能不叫“我的电脑”。
 - 2 选择工具 > 文件夹选项, 打开文件夹选项对话框, 选择查看选项卡。
 - 3 取消选中隐藏已知文件类型的扩展名, 使得所有文件扩展名都可见。
-

目 录

1 入门知识	0	3 使用和编写函数	38
1.1 ActionScript 3.0 简介	1	3.1 理解函数	39
1.2 为什么应该学习 ActionScript 3.0	1	3.2 练习 1 使用方法	40
1.3 ActionScript 3.0 的新功能	2	3.3 练习 2 编写自定义函数	44
1.4 ActionScript 1.0/2.0 与 3.0 之间的差别	2	3.4 理解模块函数	47
1.5 ActionScript 3.0 的元素	3	3.5 练习 3 让函数模块化	48
1.6 超越脚本助手	7	3.6 练习 4 让函数返回值	57
1.7 注意：必需有播放器	8	4 响应事件	62
1.8 超越 ActionScript 3.0	9	4.1 理解事件类型	63
2 与 ActionScript 通信	12	4.2 练习 1 使用侦听器捕捉事件	63
2.1 练习 1 与影片剪辑通信	13	4.3 练习 2 编写事件处理程序	66
2.2 练习 2 修改影片剪辑的属性	19	4.4 练习 3 响应鼠标事件	71
2.3 理解变量	27	4.5 练习 4 响应键盘事件	77
2.4 练习 3 设置变量的数据类型	28	4.6 练习 5 创建到网站的链接	82
2.5 练习 4 使用 Trace 语句	31	4.7 练习 6 使用“进入帧”事件 创建动画	86
2.6 练习 5 使用注释	34	4.8 练习 7 使用“计时器”事件 控制动画	94

5 理解类 100

- 5.1 理解类 101
- 5.2 练习 1 编写自定义类 102
- 5.3 练习 2 扩展现有的类 105
- 5.4 练习 3 将类对象添加到时间轴 108
- 5.5 练习 4 定义文档类 118
- 5.6 练习 5 设置类路径 124

6 判定和重复 130

- 6.1 理解条件语句 131
- 6.2 练习 1 编写条件语句 132
- 6.3 练习 2 使用条件运算符 136
- 6.4 练习 3 设置备选条件 142
- 6.5 练习 4 编写复合条件 146
- 6.6 理解循环 149
- 6.7 练习 5 创建代码循环 150
- 6.8 练习 6 使用循环生成类的实例 153
- 6.9 练习 7 放置循环所创建的实例 157

7 使用算术并爱上算术 ... 164

- 7.1 理解 Math 类 165
- 7.2 练习 1 使用基本算术运算符 166
- 7.3 练习 2 生成随机数 169
- 7.4 练习 3 数字取整 171

8 使用文本和数组 178

- 8.1 练习 1 创建文本字段 179
- 8.2 练习 2 设置文本字段的样式 182
- 8.3 练习 3 从文本字段中获取数据 187
- 8.4 练习 4 装入外部文本 191
- 8.5 练习 5 理解数组 196
- 8.6 练习 6 使用文本和数组创建游戏 ... 202
- 8.7 练习 7 完成文字游戏 210

9 创建记牌游戏 214

- 9.1 练习 1 应用程序规划 215
- 9.2 练习 2 编写记牌类 218
- 9.3 练习 3 编写记牌游戏类 222
- 9.4 练习 4 将图形添加到纸牌 224
- 9.5 练习 5 放置纸牌 230
- 9.6 练习 6 检查匹配 235
- 9.7 练习 7 重置纸牌 240
- 9.8 练习 8 处理不正确的匹配 245
- 9.9 练习 9 确定胜负 249
- 9.10 练习 10 添加更多纸牌 251

10 使用高级图形和动画工具 256

- 10.1 练习 1 使用代码绘图 257
- 10.2 练习 2 创建颜色变化 262

10.3	练习 3 创建随机的颜色变化	267
10.4	练习 4 制作颜色变化动画	272
10.5	练习 5 使用滤镜	275
10.6	练习 6 修改滤镜属性	278
10.7	练习 7 使用滤镜制作动画	281

11 处理多媒体 284

11.1	练习 1 装入外部图像和 SWF	285
11.2	练习 2 与已装入的影片通信	290
11.3	练习 3 装入声音	294
11.4	练习 4 开始和停止播放声音	297
11.5	练习 5 控制音量	302

11.6	练习 6 装入视频	306
11.7	练习 7 控制视频回放	309

12 添加高级交互性 314

12.1	练习 1 创建拖放类	315
12.2	练习 2 检测接触	324
12.3	练习 3 响应接触	328
12.4	练习 4 检测获胜	332
12.5	练习 5 随机放置对象	335

附录 A 故障排除 FAQ 340

1.1 ActionScript 3.0 简介

ActionScript 3.0 是一种面向对象的编程语言，用于增强 Flash Player 的功能。它是一种 ECMAScript 语言，这一点与 JavaScript 类似，所以如果想学习其他 ECMAScript 语言，那么 ActionScript 3.0 就是一个很好的起点。最新版本的 ActionScript 已经完全符合 ECMA 标准。

在 Flash CS3 中，ActionScript 3.0 可以嵌入到 Flash 项目 (.fla) 文件内，也可以写成独立的

ActionScript (.as) 文件，或者用 Flex Builder 创建（如图 1-1 所示）。Flex Builder 是构建在 Adobe 的 Flex 框架上的新工具，为开发人员提供了创建 RIA（Rich Internet Application，富 Internet 应用程序）的环境。本书的重点是用 ActionScript 3.0 增强 Flash CS3 项目。

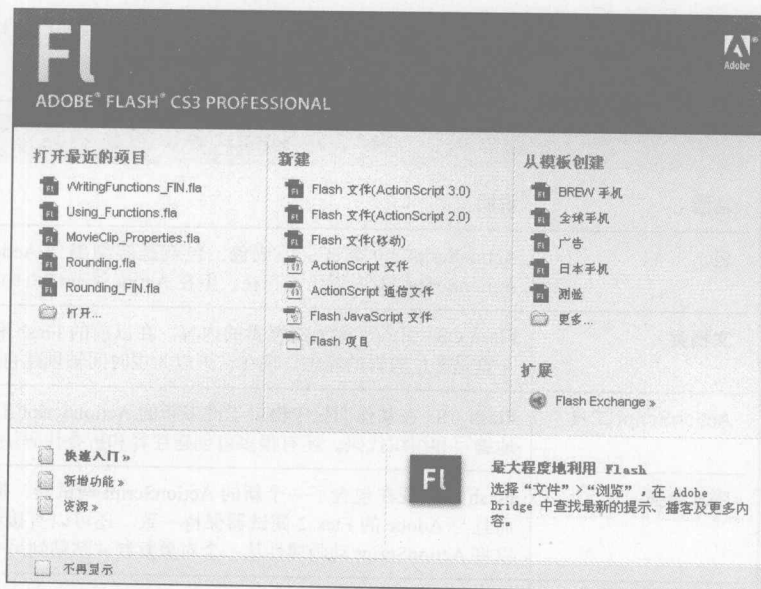


图 1-1

1.2 为什么应该学习 ActionScript 3.0

如果您是 Flash 设计师，以前可能只用过脚本助手。学习 ActionScript 3.0 后可以帮助您跨越这一障碍，创建真正具备完全交互性的应用程序，包括动态 Web 应用程序和交互视频游戏。如果您以前学过 ActionScript 1.0 或 2.0，又畏惧 ActionScript 3.0 中语言的新变

化，那么可以想像 ActionScript 3.0 只是比以前的版本多了几项关键的增强功能，包括更快的下载速度，更精确的视觉控制，更高级的交互性，能够组合位图和矢量图形并包括视频或动画，支持内容缩放和流媒体内容等。

注意



下载速度

ActionScript 由 Flash Player 内的 AVM（ActionScript Virtual Machine，ActionScript 虚拟机）执行。ActionScript 3.0 引入了 AVM2，它的下载速度最高时是以前版本的 30 倍（真是惊人！）。

1.3 ActionScript 3.0 的新功能

ActionScript 3.0 提供了许多新功能，表 1-1 中简要介绍了其中的一些。

表 1-1

ActionScript 3.0 的新功能	
功能	说明
包	ActionScript 3.0 类被组织成包，包就是容纳类似 ActionScript 类文件的文件夹。包在 ActionScript 2.0 中就已经存在，但在 ActionScript 3.0 中用得更多
文档类	Flash CS3 引入了称为文档类的内容。在以前的 Flash 和 ActionScript 版本中，主时间轴一直是影片剪辑的标志。现在，可以为时间轴创建自定义的类
ActionScript 工具	Flash CS3 在动作面板中提供了许多新的 ActionScript 工具，可以帮助您学习如何更高效地编写和组织代码。还有很多可创建注释和折叠代码块的按钮
脚本增强	Flash CS3 现在包含了一个新的 ActionScript 调试器，提供了更好的灵活性和反馈机制，而且与 Adobe 的 Flex 2 调试器保持一致。还可以直接将动画转换成 ActionScript，也可以将 ActionScript 动画属性从一个对象复制并粘贴到另一个对象上
语言的一致性	如果您熟悉 ActionScript 2.0，那么可能注意到 2.0 语言中存在一些不一致的现象。而 ActionScript 3.0 在语法上更加一致，所以了解核心概念之后，这个语言会显得非常直观

1.4 ActionScript 1.0/2.0 与 3.0 之间的差别

从表 1-1 中可以看到，ActionScript 3.0 与以前版本有许多重要的差别。下面这个实例可以更清楚地演示这些差别。

假设要在 Flash 文档中添加一个文本字段。如果使用 ActionScript 2.0，代码如图 1-2 所示



图 1-2

第 1 行代码调用了影片剪辑类。在旧版的 ActionScript 中，所有组件都必须嵌在影片剪辑内，然后使用 createTextField 方法添加文本字段。使用 ActionScript 时，所有组件都必须使用双引号括起来的实例名称。文本字段的属性(宽度、高度以及在舞台上的位置)都放在实例名称后面。

在 ActionScript 3.0 中，实现相同功能的代码则如图 1-3 所示。

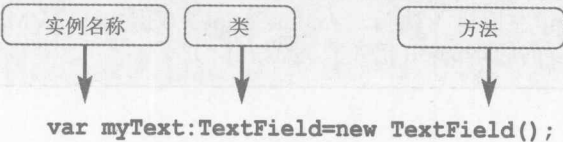


图 1-3

在 ActionScript 3.0 中, 只需创建文本字段类的新实例, 并在以后提供文本字段的属性。

下面是另一个示例。图 1-4 中 ActionScript 2.0 代码在运行时从库面板将影片剪辑元件导入 SWF 文件。

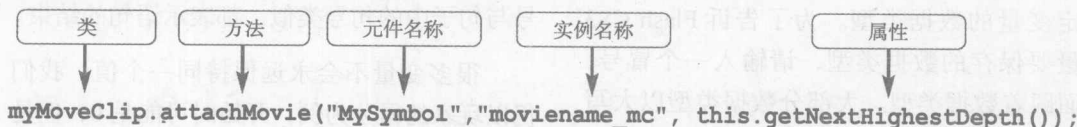


图 1-4

同样, 也必须调用影片剪辑类、attachMovie 方法、元件的名称、新实例的名称以及深度值。

在 ActionScript 3.0 中, 只需创建影片剪辑类的新实例(如图 1-5 所示), 并在稍后添加各种属性, 如实例名称、元件名称、舞台深度。而且它的语法基本上与创建文本字段或影片剪辑的语法相同。



图 1-5

注意, 这只是个别的例子。并非 ActionScript 3.0 中的每个功能所需要的代码都比以前版本减少了。ActionScript 3.0 并不是要成为更简单的语言, 重要的在于标准化。无论创建文本字段、影片剪辑、形状还是其他对象, 方法都是标准的。这种标准化使它更容易被学习和使用。

1.5 ActionScript 3.0 的元素

本书将介绍 ActionScript 3.0 的几个核心元素, 包括:

- 变量;
- 实例;
- 属性;
- 函数和方法;
- 事件、事件处理程序和事件侦听器;
- 类;
- 条件语句。

这些并非是 ActionScript 3.0 的所有元素, 但知道了如何使用这些元素, 就能执行大多数常见的 ActionScript 任务。下面几节可以作为理解本书所用的每个 ActionScript 3.0 元素的参考。如果在后续章节中遇到困难或疑惑, 可以回到本章并复习这些术语。

1.5.1 变量

变量是容纳数据的容器。为了理解变量, 请想象一种玩家可以积分的游戏。玩家得分数的信息(或数据)保存在变量内。玩家得到更多积分时, score 变量中的数值就增加, 因此 score 变量就作为数字(或数据)的容器(或变量)。

变量可以容纳的数据不仅限于数字, 还可以容纳许多其他类型的数据。变量可以容纳文本值, 如用户名、密码或者文本字段的文本。变量也可以容纳 true 或 false 值, 如登录网站的用户是否为管理员。变量容纳的数据的类型称为变量的数据类型。

在 ActionScript 3.0 中, 使用关键字

名称还应该具有描述性,但是要简洁(没有必要太长)。好的实例名称如 `contact_btn`、`nav_mc` 和 `body_txt`。

注意:如果在影片剪辑实例的名称末尾使用 `_mc`、按钮实例的名称末尾使用 `_btn`、文本字段实例的名称末尾使用 `_txt`,那么 Flash CS3 会在“动作”面板中提供代码提示。这样编写代码会更快更轻松。

1.5.3 属性

无论相信与否,您已经熟悉了属性。属性就是元件实例上所附加的变量。如果这么说听起来有点拗口,那么可以将属性当作能够在属性检查器中修改的特性。具体来说,实例的 `x` 和 `y` 位置及其宽度和高度就是实例的一些属性。处理影片剪辑或按钮时,其他属性还包括 `alpha`、混合模式,甚至包括筛选器。在 `ActionScript` 中可以使用点(`.`)语法修改属性,即输入实例名称、点和要修改的属性,然后用等号(`=`)输入要赋予属性的值(就像设置变量的值一样)。修改实例名称为 `my_mc` 的影片剪辑的位置 `x` 并将该位置设为 100 的代码是 `my_mc.x = 100;`。

1.5.4 函数和方法

使用 `ActionScript 3.0` 时,经常要编写多次重复使用的大块代码段。多次复制相同的代码段既折磨人又烦琐。幸运的是,利用函数和方法可以一次性编写大块的代码段,然后多次循环使用。函数和方法本质上是相同的即都是可重用的代码段。

Flash CS3 中已经有了很多函数,如想停止影片的循环播放,可以使用预置的 `stop()` 函数。要使用函数,请输入函数的名称,然后输入一对括号。运行 `stop()` 函数的代码是 `stop();`。Flash CS3 中另一个预置的常用函数是 `gotoAndPlay()`。`gotoAndPlay()`

函数从特定的帧开始播放 Flash 影片。运行 `gotoAndPlay()` 函数时,要在括号内告诉 Flash CS3 从哪个帧开始播放。例如,从第 5 帧开始播放影片的代码是 `gotoAndPlay(5);`。

注意:括号是函数可以运行并且区别于变量的特性。如果输入 `stop`,但是没有输入括号,那么 Flash CS3 会认为正在引用一个名为 `stop` 的变量。

有些函数在 Flash CS3 中没有,所以必须自己创建。编写自定义函数需要几个步骤。首先要用 `function` 关键字告诉 Flash CS3 现在要创建函数,然后告诉 Flash CS3 函数的名称。如果要创建名为 `myFirstFunction` 的函数,就应该输入 `function myFirstFunction`,再输入一对括号。我们学习事件时会了解括号的作用。在括号后面,需要用冒号告诉 Flash CS3 该函数返回的数据类型(与设置变量的数据类型相似)。本书所用的所有函数的返回类型都是 `void`。最后,将需要重复使用的代码块放在大括号内。例如,下面的代码创建一个名为 `myFirstFunction` 的完整函数。

```
function myFirstFunction():void
{
    (代码块写此处)
}
```

运行自定义函数的方法与运行预置函数类似,输入函数的名称,再输入括号对。运行函数 `myFirstFunction` 函数的代码是 `myFirstFunction();`。

1.5.5 事件、事件处理程序和事件侦听器

事件是在 Flash 影片播放过程中发生的事情。事件类型很多,如网站的访客单击按钮、按下键盘上的键或者开始下载文件等。通过在事件发生时运行函数,可以对事件加以利用。事件发生时运行的特殊函数称为事件处理程序。

要编写事件处理程序，只需创建基本的函数即可。事件处理程序函数的不同之处在于：它要接收使函数运行的事件信息。以下代码创建称为 `playMovie` 的事件处理程序函数，对按钮单击进行响应。

```
function playMovie(event:MouseEvent):void
{
}
```

注意：括号中的代码 `event:MouseEvent` 很重要，它的作用是捕捉让函数运行的事件信息。`Event` 部分代表发生的事件，冒号后面指明该事件的数据类型是 `MouseEvent`。单击按钮运行这个函数之前，需要将事件处理程序函数附加到事件（单击按钮）上。

要将事件附加到事件处理程序，需要使用所谓的事件侦听器。事件侦听器等待事件发生，事件发生时就会运行对应的事件处理程序函数。要理解事件侦听器，可以将电台广播音乐当成一个事件。无论听众是不是在收听，它都会广播。打开收音机并调整到一个台，就相当于听到一个事件。听到音乐时，就会选择如何对音乐做出响应，如跟着一起唱、换台或者加大音量。同选中的电台将听众与广播所发出的信号连接起来一样，事件侦听器也将事件处理程序与事件连接起来。

对 Flash CS3 中的按钮或任意对象侦听事件，请使用对象的 `addEventListener` 方法（方法与函数的意义相同）。首先输入实例名称、点，再输入 `addEventListener`，然后在括号内告诉 Flash CS3 实例要侦听的事件，再输入逗号（,），然后输入事件发生时要运行的函数名称。例如，如果有一个实例名为 `play_btn` 的按钮，希望单击这个按钮时运行称为 `playMovie` 的函数，可输入以下代码：

```
play_btn.addEventListener(MouseEvent.CLICK,playMovie);
```

注意：在 Flash CS3 中，事件以事件的数据类型开始（在这个示例中是 `MouseEvent`），后面跟着点（.），然后是具体的事件名（全部用大写）。还要注意的，如果向函数发送多个值，要用逗号分隔这些值。

1.5.6 类

您。类就是事物的总括或者概念。想想影片剪辑元件和按钮元件之间有什么区别吧？影片剪辑拥有近乎无限数量的帧，运行 Flash 影片时就会播放这些帧（除非用 `ActionScript` 指定其他操作）。而按钮的时间轴上只有 4 个帧，但是这几帧并不播放，只有在鼠标移到上面或者单击它们时才播放。`MovieClip` 类是所有影片剪辑的蓝图。虽然每个影片剪辑元件看起来可能各不相同，但所有影片剪辑都有某些共性，如大写字母开始，所有的类也都是数据类型。`Number`、`String` 和 `Boolean` 都是类。

从库面板拖出影片剪辑或按钮元件实例并将它们放在舞台上，这就创建了 `MovieClip` 或 `Button` 类的实例。许多类是不可视的，所以必须在 `ActionScript` 中用 `new` 关键字创建这些类的实例。在 `new` 关键字的后面输入类名和一对括号。`Loader` 就是一个不可视的类，用来加载外部内容。要创建 `Loader` 类的新实例 `myLoader`，应该输入以下代码：

```
var myLoader:Loader = new Loader();
```

注意：该代码创建 `Loader` 类的新实例，它与创建其他不可视类实例的代码类似。在 `new` 关键字后面，注意要有类的名称和一对括号。看起来这与运行函数的语法类似，因为 `new` 就是一个函数，只不过是一种特殊的函数，称为构造函数，负责创建特定类的新实例。

1.5.7 条件语句

使用条件语句可以根据条件的真或假来

运行一个代码块。例如,早晨起来穿衣服(或者随便什么时候想穿衣服),准备既穿鞋又穿袜子,如果已经穿袜子了,那么只要套上鞋就行了,如果没有穿袜子,那么就先得先穿袜子后穿鞋。

在 ActionScript 3.0 中处理条件的方式与上述方式类似。假设有一个变量 socksOn,它的数据类型是 true 或 false 数据类型(Boolean)。要编写条件语句,首先使用关键字 if,然后将运算结果为 true 或 false 的表达式放在括号内,最后将要执行的代码放在大括号内。在下面的条件语句中,如果已经穿了袜子就会穿鞋。

```
if(socksOn)
{
    (穿鞋)
}
```

注意: 如果 socksOn 变量的值为 true,就会运行代码(穿鞋)。关键字 if 后面括号里的内容,其运算结果是 true 或 false 值。

在条件不为真时也可以执行代码块。为此要使用关键字 else,将要执行的代码放在大括号内。还是用前面的代码,如果要在 socksOn 为真时穿鞋,在 socksOn 为假时穿袜子,应该输入以下代码:

```
if(socksOn)
{
    (穿鞋)
}
else
{
    (穿袜子)
}
```

注意: else 关键字后面不需要任何括号。

1.6 超越脚本助手

如果您以前做过许多 Flash 设计,那么肯定用过脚本助手,Flash 中的这项功能可以帮助您编写 ActionScript。知道需要调用的函数或方法,但是不记得正确语法时,脚本助手很有帮助。要打开脚本助手,请执行窗口 > 动作命令,打开动作面板,或者在键盘上按 F9 键,再单击屏幕右上角的脚本助手按钮(如图 1-8 所示)。

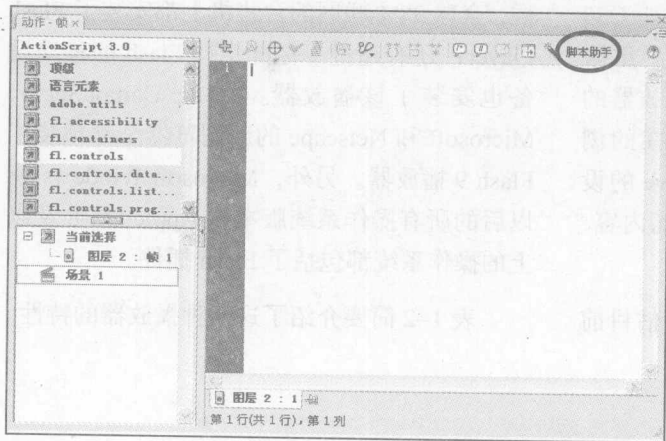


图 1-8