

Flash



Design
Lab⁺

Flash设计师 高薪秘籍

[日] デザインラボ編集部 著
彭彬 译

因为专业，所以不凡！
引领你进入Flash的殿堂，
跻身于专业设计师的行列。

通过全Flash网站的制作过程，充分展现Flash灵活多样的表现手法和
无与伦比的力量。

从如何构造Flash画面到如何使用工具全面讲解了ActionScript 3.0，
帮助你快速掌握专业Flash设计师必备的技能！

详细剖析了极具表现力的18个超级经典案例，包括读取外部文件、
Global Navigation、PhotoGallery、动作XML、波形编辑等主题。



DOWNLOAD

本书示例的所有
内容均可下载。

图书在版编目 (C I P) 数据

Flash设计师高薪秘籍 / 日本デザインラボ編集部著
; 彭彬译. -- 北京 : 人民邮电出版社, 2011. 2
ISBN 978-7-115-24136-8

I. ①F… II. ①日… ②彭… III. ①动画—设计—图
形软件, Flash IV. ①TP391.41

中国版本图书馆CIP数据核字 (2010) 第209228号

版 权 声 明

Flash Design Lab

Edited by Design Lab Team

Copyright © 2008 Softbank Creative Corp.

Chinese translation rights in simplified characters arranged with Softbank Creative Corp., Tokyo through Japan UNI Agency, Inc., Tokyo

本书中文简体版由日本Softbank Creative公司经由日本UNI版权代理公司授权人民邮电出版社独家出版。未经出版者书面许可, 书中任何部分不得以电子和机械等任意形式, 通过翻拍、记录等手段复制、伪造、传播。

版权所有, 侵权必究。

内 容 提 要

本书分为基础篇和实践篇两部分, 基础篇包括了第1~5章, 介绍了Flash的基本操作、符号的层次结构、补间动画、制作全Flash动画网站的基础、ActionScript 3.0开发和辅助工具的使用等; 实践篇包括了来自不同资深设计师创作的18个有代表性的综合实例。

本书适合网页设计师、多媒体设计师、动画设计师、网络广告设计师、网络游戏设计师等互联网从业人员阅读参考, 也适合相关专业的院校师生和培训机构用作参考书。

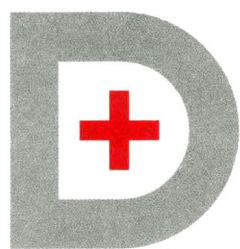
Flash 设计师高薪秘籍

- ◆ 著 [日] 日本デザインラボ編集部
译 彭 彬
责任编辑 杜 洁
 - ◆ 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>
北京画中画印刷有限公司印刷
 - ◆ 开本: 880×1230 1/16
印张: 10
字数: 501千字 2011年2月第1版
印数: 1~4000册 2011年2月北京第1次印刷
- 著作权合同登记号 图字: 01-2010-1201号

ISBN 978-7-115-24136-8

定价: 45.00元

读者服务热线: (010)67132705 印装质量热线: (010)67129223
反盗版热线: (010)67171154



Design
Lab+

Flash设计师 高薪秘籍

[日]デザインラボ編集部 著
彭彬 译

人民邮电出版社

北 京

基础篇

Basic Knowledge

❖ Lesson 01

Flash概要和基础

- 006 Flash的5大功能/Flash播放器的普及率
- 007 Flash界面的构成
- 008 时间轴/帧和帧频/帧和关键帧
- 009 各种帧的状态/图层
- 010 基本面板
- 011 编辑面板/动作面板/各种面板的显示方法
- 012 Flash的绘图工具/使用Flash所制图形的特征/图形信息的独特设计
- 013 Flash图形巧妙的选择方法/基本图形的绘制方法(矩形、椭圆、多边形)/“填充”和“线”的编辑
- 014 元件和实例/元件的种类/元件的转换(添加)
- 015 元件的使用/实例的属性/元件的编辑1/元件的编辑2

拥有分层结造的元件

- 016 拥有分层结构元件的优点/使用影片剪辑播放嵌套
- 017 按钮的使用/熟练运用嵌套

动画和缓动

- 018 所谓动画补间/动画补间的设定方法/动画补间的详细设定
- 019 缓动/自定义缓入/缓出

❖ Lesson 02

通过Full Flash Site 学习Flash的基础知识

- 020 Full Flash Site的概要/首先尝试制作

整理Full Flash Site

- 030 在按钮上配置影片剪辑
- 031 在各页面的Contents中配置元件

❖ Lesson 03

ActionScript 3.0入门

- 034 什么是ActionScript
- 035 ActionScript的编写区/动作面板的首选参数
- 036 ActionScript的编写/特定实例
- 037 目标路径的利用/认识目标路径的设置
- 038 使用变量/变量的定义
- 039 运算符/算术运算符/赋值运算符/比较运算符和逻辑运算符/文字的连接

ActionScript的数组和控制语句

- 040 数组(Array)/条件分支(if句型和if ~ else 句型)
- 041 运用多个条件式(else if句型)/循环处理(for句型)
- 042 尝试编写函数/根据用户的操作事件调用函数
- 043 ActionScript的调试/检查代码编写错误

❖ Lesson 04

- 044 ActionScript实际操作讲座
- 046 外部文本的读取
- 048 进度条的设置
- 050 跟随鼠标移动的图片剪辑实例
- 052 外部AS文件的使用

❖ Lesson 05

方便操作的辅助功能和工具

- 056 “贴紧”功能/“网格”功能
“辅助线”功能
- 057 “对齐”面板/放大镜工具(工作区的扩大和缩小)/
手形工具(工作区的移动)/时间轴的移动
- 058 快捷键的使用

目录

Contents

实例篇 Case Study

- 060
实例 01
显示加载进度的进度条



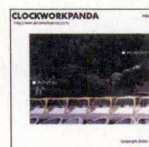
- 064
实例 02
通过加载文本文件动态更新内容



- 068
实例 03
Web网站和外部文件的链接



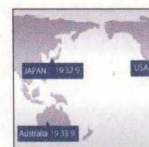
- 072
实例 04
图像的放大与缩小



- 076
实例 05
用Flash实现图像处理



- 082
实例 06
制作获取时间的世界时钟



- 086
实例 07
通过计算差分扩大或缩小图像



- 090
实例 08
可用作演示文稿的幻灯片切换



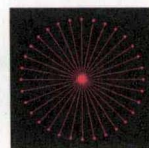
- 98
实例 09
Flash中FLV文件的播放



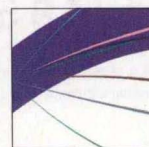
- 106
实例 10
显示从XML文件中读取的菜单



- 112
实例 11
使用物理法则“加速度”和“空气阻力”的动画



- 116
实例 12
飘带的模拟实验



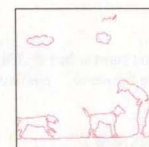
- 120
实例 13
用扩大、缩小的圆点表示输入键



- 126
实例 14
读取外部图像文件的Photo Gallery



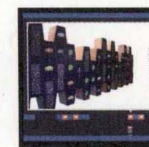
- 132
实例 15
使用Motion XML实现一笔画



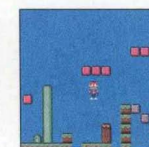
- 138
实例 16
波形编辑应用程序



- 146
实例 17
实现键盘操作模式的游戏内容



- 154
实例 18
实现拖放功能



本书的使用方法

How To Use

本书将讲述使用 Flash 制作各种网站内容和创建网站的方法。

范例文件（需下载）中收录了基础篇（部分）和实例篇中讲解的所有实例。请在深入阅读本书基础之上用作参考。

※ 范例文件下载

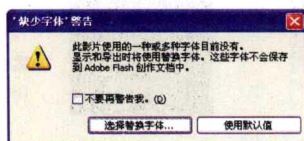
在本书中讲解的范例资料，可以从以下 URL 地址下载。下载的范例文件中包含了基础篇（Basic Knowledge）中的部分范例和实例篇（Case Study）中的所有范例。

<http://isbn.sbcr.jp/48521/>

下载后的范例文件中的资料都含有著作权。不能公开其图表、部分画像或者全部内容，也不能在修改后使用。另外，请注意由于使用下载资料而发生任何损失，作者以及 SoftBank Creative 有限公司一律不承担责任。

※ 关于警告画面的处理

在打开包含在下载范例文件中的 FLA 文件时，有可能会出现类似右图这样的画面。这是因为在 FLA 文件里使用的字体在使用的电脑中没有被安装的原因。当显示该画面时，请选择“使用默认值”。这样就将其换成了电脑中安装的字体，和本书的截图相比可能存在差异，但是并不影响使用。



※ 必要的系统环境

在利用本书的范例资料时系统环境要求如下。

Windows

- 配备了 Intel® Pentium® 4、Intel Centrino®、Intel Xeon®、或者 Intel Core™ Duo 处理器的计算机
- Microsoft® Windows® XP (Service Pack 2)、Windows Vista™ Home Premium、Business、Ultimate 版本或者 Enterprise 版本（推荐 32 位版本）
- 512MB 以上的内存（推荐 1GB 以上）
- 2.5GB 以上可用容量的硬盘（需要安装的情况下还需要一些额外的空容量）
- 支持 1024 × 768 以上的画面分辨率的显示器
- 因为需要 Product Activation（许可认证），所以要能够连接到互联网

Macintosh

- Power PC® G4、G5 或者是 Intel 的处理器
- Mac OS X v10.4.8-10.5 (Leopard) 版（但是，不是对所有 Leopard 都适用，详情请参见与 Mac OS X Leopard 相对应的 Creative Suite3 的应用有关的 Adobe 计划（PDF：295K）。
- 512MB 以上的内存（推荐 1GB 以上）
- 支持 1024 × 768 以上的画面分辨率的显示器
- 因为需要 Product Activation（许可认证），所以要能够连接到互联网

Flash

基础篇

Basic Knowledge

Flash 概要和基础

从业余爱好者到专业的设计师，Flash 都是很多用户使用的行业标准网络应用开发工具。接下来将介绍在学习 Flash 之前必要的 Flash 概要知识和基本结构。

Flash 的 5 大功能

作为网页内容的制作工具，Flash 由于其实用性而广泛普及。在 Flash 普及的背景下，本篇列举了由于宽带化的进步使通信环境变得充实、因矢量数据化的动画片而使通信环境问题得以解决、还有运用插件解决浏览器和 OS 不同而出现的显示差异，使初学者也可以轻而易举地编制 Flash 内容等例子。

另外，Flash 在网页制作上包含各种非常便利的功能，这也正是 Flash 的特征之一。主要的功能有：用于绘图“图形化功能”、图像动作的“动画功能”、用于的读取其他软件所制图像的“输入功能”、在动画中添加代码使其拥有双向传递性（根据用户的操作）的 ActionScript 功能、用于制作网页最适合文件的“发布功能”等。

其中 Flash 的最大特点是使用称为 ActionScript 的脚本语言，根据用户的操作便利地制作各种双向式信息传递结构。

此外，现在的 Flash 拥有很多功能，可以实现很复杂的功能，但并不是要求所有人必须掌握所有的操作，由于所有的功能都按照用途划分，所以即使是初学者也可以轻松使用 Flash。



使用一个软件，就可以编制特化 Web 的内容。

图形化功能

将制图作为 Flash 内容的功能。它是以矢量的形式存储数据信息，所以比起其他动画软件可以使用更少的信息量更简便地制作动画。

动画功能

将制作好的图像和外部获取的图像利用补间方式制作简单的动画效果。

输入功能

支持添加其他软件制作的画像、音乐、影像。使用该功能可在网上处理各种各样的内容，同时制作更富魅力的信息内容也将成为可能。

ActionScript 功能

使用 ActionScript 功能，在动画中添加代码，可以实现从按用户选择信息内容变化到时间轴的控制、物理模拟实验等各种各样的功能。除了动画动作，包括客户端的联动等高难度技术都能比较轻松地完成。

发布功能

支持生成最适合网页使用的文件的功能。此外该功能还不限于 Web，还可以采用各种方式输出，所以用途也就不仅仅局限于 Web。

Flash 播放器的普及率

使用 Flash 制作的内容在浏览器上播放需要添加“Flash 播放器”插件（为提供功能而在应用软件上添加的小程序）。当没有配置 Flash 播放器，需要显示 Flash 内容时必须事先从 Adobe 公司的网站下载并安装。所以一些企业网站在内容制作时会避免使用 Flash 内容，就在于安装单独的播放插件会带来使用上的麻烦。

现在因为下载操作简单化、Flash 内容的增加，Flash 播放器也广泛地普及起来。根据 Adobe 公司的调查，在日本 Flash 播放器的普及率 FlashPlayer8 达到 98.1%，最新的 FlashPlayer9 也达

96.5%（2008 年 3 月），基本上所有人都能毫无问题地浏览 Flash。

另外，根据制作（发布）好的版本，需要 Flash 播放器版本是不同的。例如，用 FlashPlayer9 发布的 Flash 内容不能用 FlashPlayer8 播放。通常用最新的版本制作不一定效果就好，所以制作时有必要稍加注意。



表示 Flash 播放器的图标和表示 SWF 文件的图标。安装了 Flash 播放器的话就可以播放 SWF 文件。

制作 Flash 的版本	ActionScript 的版本	Flash 播放器的版本	普及率
Flash MX	1.0	Player6	97.4%
Flash MX 2004	2.0	Player7	98.5%
Flash 8	2.0	Player8	98.1%
Flash CS3	3.0	Player9	96.5%

※ 2007-2008 adobe.com: Worldwide Ubiquity of Adobe Flash Player by Version
http://www.adobe.com/products/player_census/flashplayer/version_penetration.html

Flash 界面的构成

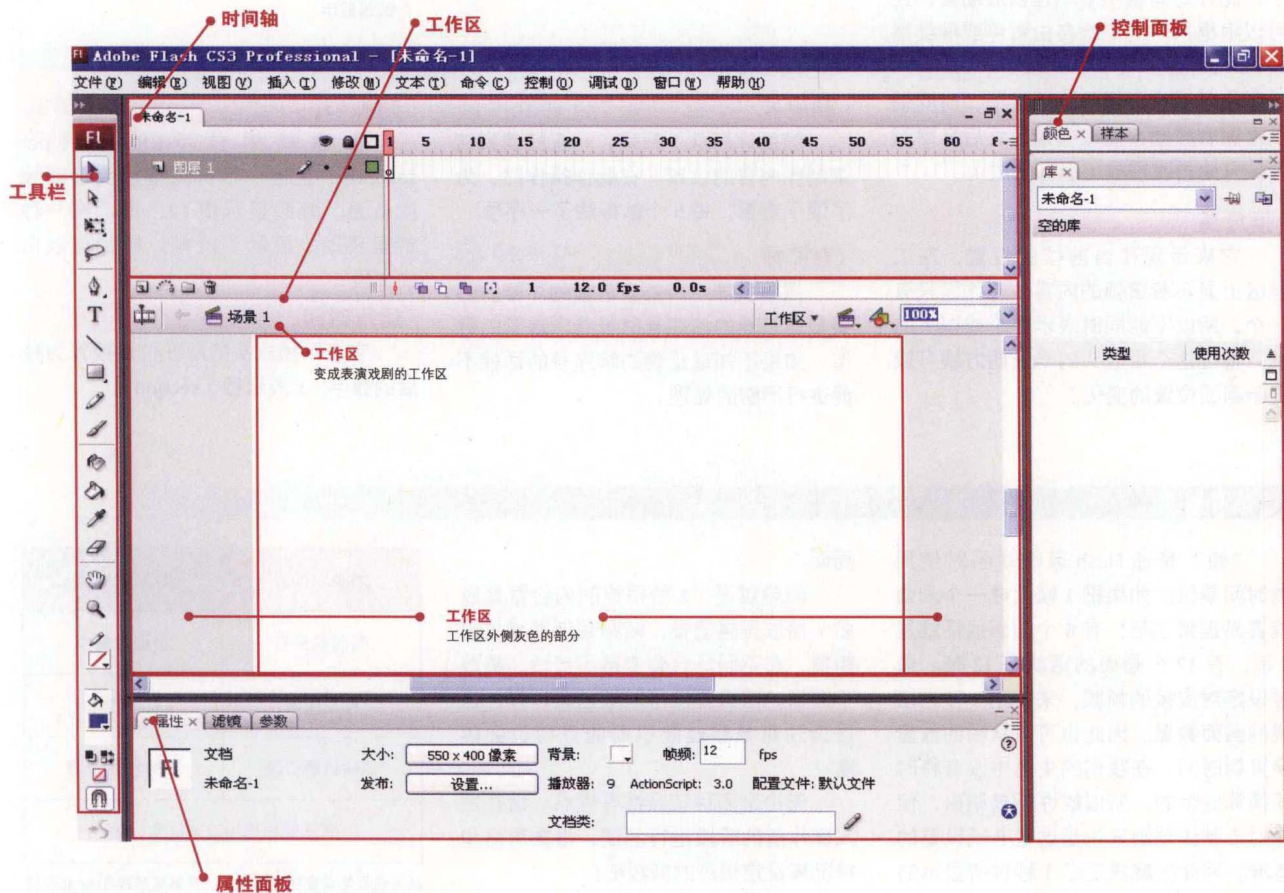
无论在任何应用过程中制作画面都是存在的,有的把它叫做画布,有的把它叫做画面,有各种各样的称呼。在Flash中,制作画面称为“工作区(Stage)”。工作区(Stage)的字面意思对于大家来说是舞台的工作区或者是演奏会的工作区等,脑海中是不是会浮现愉快地观看某东西的情景呢?实

际上Flash也是一样,可以将它想成在“工作区中制作戏剧,并将它发布”的过程。

对于阅读的人来说只能看到工作区的内容。但是,在这其中的音箱、照明、工作区监控、演出人员等是与各种各样的人相关的。即使是Flash也一样,它将

这些内部操作作为功能添加,完成一部戏剧。例如,从工作区外侧显示所有内容的动画需要使用一定的工作区域。

在Flash界面中有控制时间的时间轴、主要用于绘画功能的工具栏、各种各样属性变更的属性设置,为我们辅助操作的控制面板等,若使用该软件,就可看见这些必备的工具。首先逐一认识一下必备的各个部分吧!



TIPS >> 工作区

在制作Flash内容时,不经意间很容易就会只关注工作区,但在制作动画时需要灵活运用工作区后台。Stage是工作区,所以Work Area则为工作区后台。在工作区后台中等候着为演出而准备的演员等。Flash也是这样,在工作区中显示之前需在工作区后台做好准备。暂时在工作区后台待命,必要时可移到工作区内的动画也是其中的一个例子。



首先在工作区后台中放置动画素材。最初隐藏在看不到的工作区后台。虽然在操作画面上看得见它,但是实际播放时是看不到的。然后,在制作动画时将素材移至工作区上,即可作为动画播放出来。

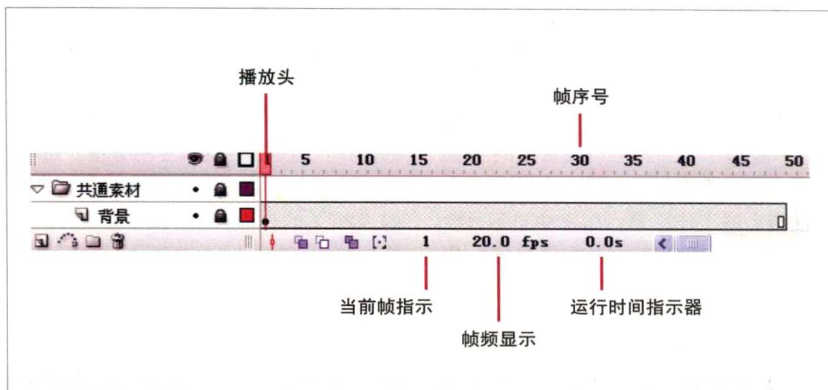
使用 Flash 播放动画时需要使用时间轴。时间轴是用来管理 Flash 内容的时间关系。当播放动画时，播放头以从左到右的方向移动，显示包含在当前帧中的内容。

任何运动的事物都伴随着时间的流逝。我们生活的世界也是通过时间来控制的。经过的时间、速度调整、运动状态，运用这些就可以播放好看的动画了。此外时间轴不单单是播放动画，还可以根据用户的控制自由暂停或继续播放。

在什么也不操作的情况下，时间轴从左朝右移动，播放到最后之后又回到开头开始再次播放（循环播放）。

播放头

它表示现在当前帧的位置，在工作区上显示着该帧的内容。工作区只有一个，所以不能同时表示两个或以上的帧。拖拉这个播放头的半透明方块可以显示画面位置的变化。



帧序号

帧的序号从 1 开始，一直持续到结束制作内容的位置。在制作操作时，为了便于查看，每 5 个帧标注了一序号。

当前帧

播放头表示所处当前帧的序号。想要知道正确的帧序号时可参考这里的数值。如果不知道正确的帧序号的话就不能进行严密的处理。

帧频 (fps)

所谓帧频显示 (fps:frame per seconds) 是指一秒钟内播放头移动帧的数量。帧频显示值 12，表示在一秒钟里移动（显示）12 帧。帧频可以自由变换。

运行时间

它显示播放头的所在位置换算为秒后的数字。s 表示秒 (second)。

帧和帧频

“帧”是在 Flash 运行动画时使用的单位。如果把 1 帧看成一个画面就容易理解了吧！有 6 个画面的话就是 6 帧，有 12 个画面的话就是 12 帧。通过设定对应帧的帧频，来决定 1 秒内出现的画面数量。因此也可以从帧的数量换算到时间。在我们的生活中没有将时间换算成帧的，所以或许不易明白，但是对于制作动画来说是这是必须设置的基准。因此帧就决定了 1 秒钟所显示的

画面。

帧频越高（1 秒播放的内容数量越多）播放得越流畅，帧频越低就播放越粗糙，在不同场合会变得不灵活，播放不顺畅。粗看会觉得帧频越高越好，但是请注意帧频越高机器的负荷也会越高。

无论是哪种帧频都有优点。请在进入操作前的阶段进行测试，根据用途和状况来设定相应的帧频吧！

帧频数值高	帧频数值低
画面数量多	画面数量少
移动流畅	移动粗糙
消耗机器资源	节省机器资源
有必要根据用途来改变帧频	

帧频是非常重要的考虑因素，是熟练掌握 Flash 必须的知识。在充分理解帧频特征的基础上开始进行操作吧！

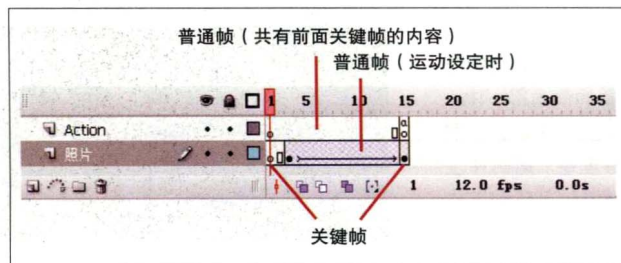
帧和关键帧

前面说到的“帧”在其他 Flash 中有的也被叫做“关键帧”。关键帧是用于制作新场景的帧，常在定义巨大变化时使用。在改变场景和编写 ActionScript、分配帧标签时，关键帧就变得非常必要了。使用关键帧展开画面，直至下一关键帧之间改变或移动图形，这就是 Flash 动画的基本形式。

帧就像电影中的帧一样，与前面的关键帧共享相同的内容。因此修改关键帧的内容，也会给后面的帧带来影响。

例如，想象从画面左端出现的角色一直走到右端的场景，需要定义在左端的状态和在右端状态的 2 个关键帧，中间移动的状态使用普通帧来显示。接

下来会讲述在使用动画补间功能等场合时，关键帧是扮演了很多重要角色的帧。就如名称所示，它是操作时关键的帧。



前面关键帧的内容都会共有到下一关键帧。中间灰色的帧处于普通状态，是共有前面关键帧内容的帧。箭头显示的淡紫色帧处于设定动画补间动画时的状态。

各种帧的状态

显示在时间轴上的帧在不同状态时，其表现形式也不尽相同。如不事先理解它们的含义，就无法正确地进行时间轴和帧的操作，所以这里先介绍一下帧的状态。

打开 Flash 新文件时，处于帧上没有放置任何东西的状态，就如 **1** 那样显示着半透明的帧。每 5 帧显示一帧灰色，所以在进行插入关键帧等操作时非常方便。另外在帧上还能带有像 **3** 这样的标签。因为带有了标签，既可以一看便知这个帧表示什么意思，也能在 ActionScript 中作为指示的对象。

1		5		9	
	在工作区上什么也没操作， 什么都没有配置的帧		配置了空关键帧， 之后持续相同状态的帧		设定形状补间帧
2		6		10	
	关键帧中设定了 ActionScript 的状态		在工作区上配置了一些项目的关键帧		没有正确设定补间时， 显示的不是箭头而是虚线
3		7		11	
	在关键帧上设定了标签名的状态		在工作区上有配置了项目的关键帧， 之后持续相同状态的帧		帧中插入了声音源的帧，显示出波形
4		8			
	在工作区上什么也没有配置的 关键帧状态（空关键帧）		设定动画补间的状态		

图层

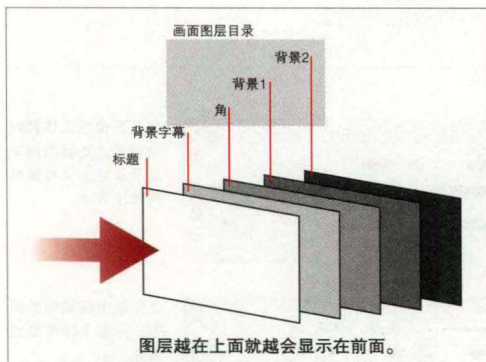
图层是如透明文件一样的东西。它可以将图形和动画像地壳那样重叠好几层来显示。通过把图形和动画分层，可以将操作细化处理，制作工作也由此变得简单。

图层越在上面就越会显示在前面。任何时候都可以改变图层的顺序。在改

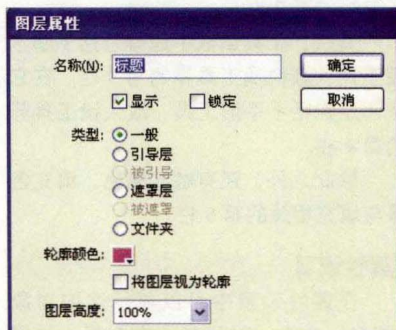
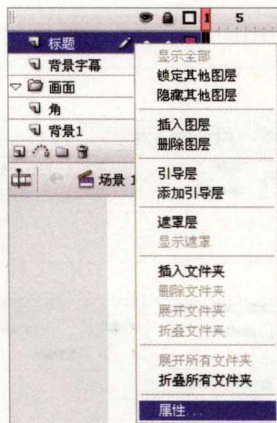
变顺序时，拖曳图层名部分，上下移动即可改变显示的次序。

各个图层都有时间轴，虽然构成不同的帧，但是时间轴是共同的。图层的显示与隐藏等各种控制都是使用图层的按钮来完成的。另外，控制图层的状态（隐藏或显示所有图层）、（锁定/解锁所有图层）、

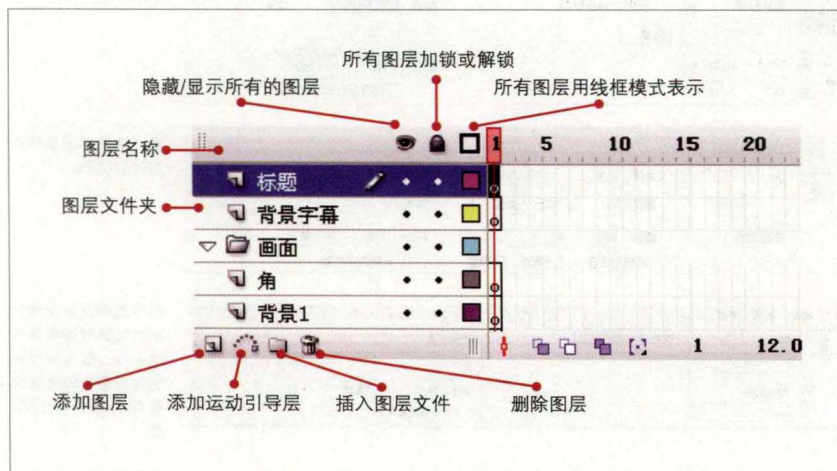
（用图层的线框模式隐藏所有图层）的按钮，点击各个图标的话，操作效果会反映到所有的图层上。想对各个图层进行设定的时候，分别点击各个图层的图案部分进行更改。使用图层属性的话，可以统一进行设定。通过改变属性，各个图层的状况都会有所变化，所以请务必记住这一点。



如果是接触其他制图软件的人是不是感觉有见过图层呢？图层是被很多制图软件采用的共通概念。



在图层上点击右键从菜单中选择“属性”，就显示了属性窗口。在该窗口中可以统一改变图层的状态等。



添加运动引导层

运动引导层可在动画补间时使用。

图层文件夹

它是将两个或以上的图层以组群等形式归纳的文件夹。在文件前面有一个三角形的开闭符号，所以不必要的时候就可以关闭。当素材很多时，整齐分开归类的话，操作也就变得简单了。

全部的图层用线框模式表示

它指将图层的内容转换为用线框的形式来表示。当对象变为透明等时也就看不见了，所以可以用线框模式显示来进行确认。用配置于各图层的颜色来表示线框。

虽然 Flash 可以进行各种操作,但是另一方面因为面板很多,所以一看到操作界面瞬间就感到头疼的用户也不少吧!实际上如果自己配合使用功能,限定显示面板的话,界面也会变得清爽很多。通过配合用途转换界面,就能更容易使用了吧!另外,在面板中包含经常使用的工具和只能在特定场合使用的工具,所以这里先介绍一下。

右图所示“工具面板”和“属性设置”这两个被称为基本面板。所谓的基本面板是指在使用 Flash 时最初使用的面板。无论任何操作都会频繁使用它,所以要牢牢记住哦!

工具面板和属性设置在生成和编辑图形时使用。特别是属性设置,它在动画的控制等操作中必定会用到。

工具面板

在工具面板中包含 5 栏,按照重要程度从高至低依次排列。

说到各栏,首先是包括了选择工具、任意变形工具等的第 1 栏。在它下面是包含了线条工具、文本工具、铅笔工具等的第 2 栏。

其次,工具面板中还有包括了墨水瓶工具、颜料桶工具等的第 3 栏。在它下面是包括了手形工具、放大镜工具等的第 4 栏。

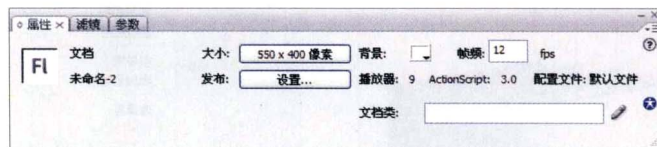
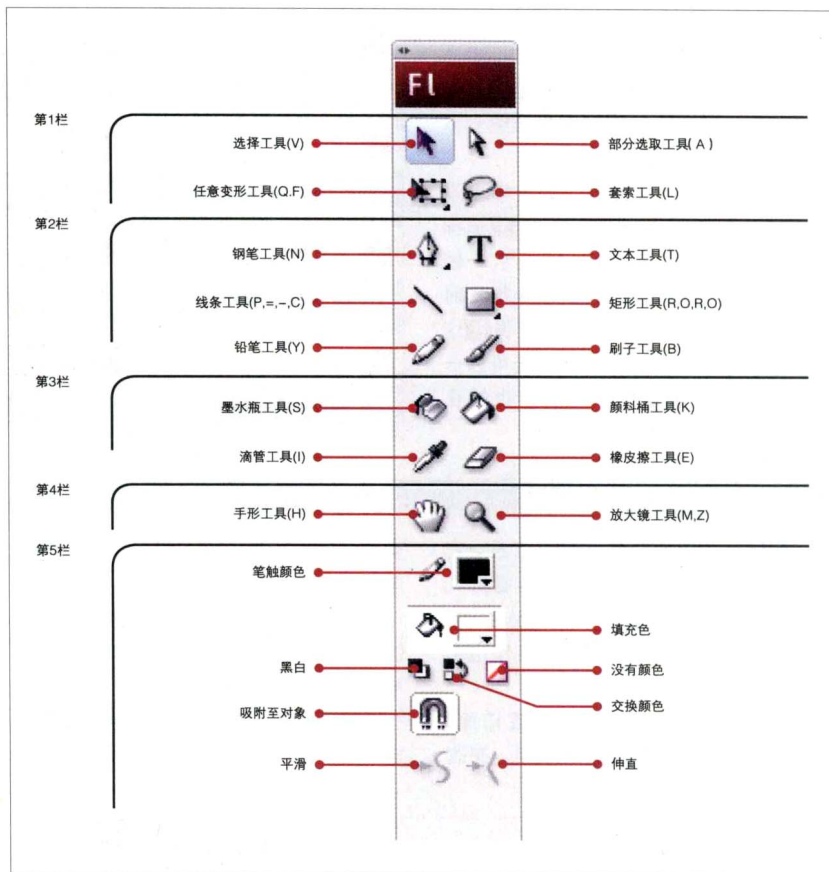
除此之外,还有触笔颜色、填充色等与填充相关的第 5 栏。

属性设置

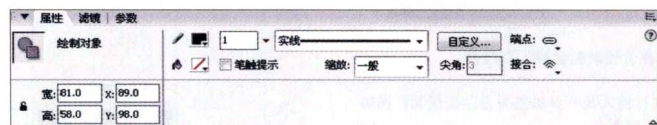
在属性设置中可以显示选中对象(图形、文本、时间轴、实例等)的属性。因此,通常可以根据所选内容改变显示的状态。属性显示的是所选择对象的特征。因此,它可以编辑每个对象,当然也能改变其内容。

Flash 的操作是选择一帧或选择工作区的对象进行操作。原打算选择某帧,但却选择了工作区的某对象,或者相反情况时都是有可能的。因为只要确认属性设置就可以知道选择了什么,所以至少在尚未熟练期间必须要设法做到确认,这样做的话错误也会减少。

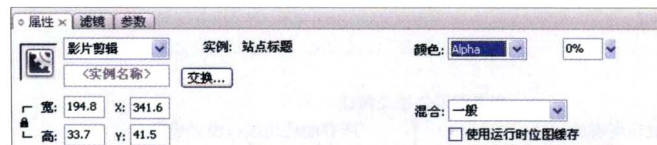
例如,即使选中了时间轴上什么也没插入的帧,也可以改变属性设置的状态。由于属性是根据被选中的部分发生变化,所以好好确认吧!



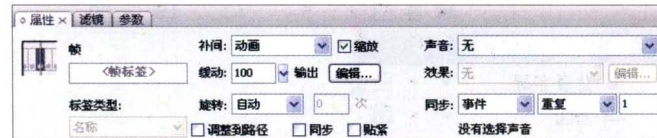
表示在点击工作区时的状态。此状态用来显示设定主文档属性的操作方法。



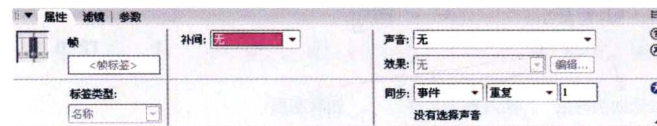
表示选中所绘制的图形,并显示该图形的属性。



表示选中工作区中实例的内容。



表示选中设定动作后的帧的状态。

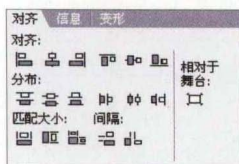


表示选中没有设置和配置帧的时候所显示的内容。显示帧中不包含对象也没有被设置任何属性时的状态。

编辑面板是在工作区上编辑图形和实例的面板组群。它运用于将制成的作品进行后期编辑或者执行动画时。它是非常重要的面板组群。

对齐面板

该功能主要是在将图形对齐时使用。该功能可以相对于工作区对齐，也可以排列同种类的图形。这是在制作界面和对齐多种项目时非常有用的面板。该功能有左对齐、居中对齐等一般功能，所以也可以用于整理分布、匹配大小等高级的项目。



在以舞台为基准时就选中“相对于舞台”按钮。

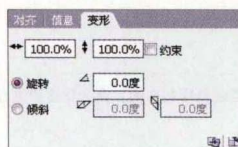
信息面板

该面板用坐标表示选择图形的大小和配置场所。另外，它还可以使用滴管工具确认颜色。



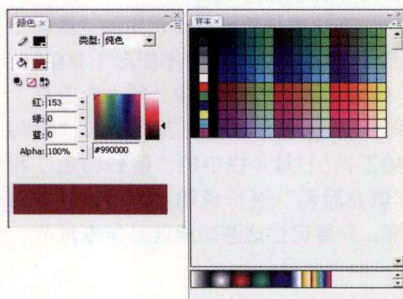
变形面板

该面板在改变图形的大小、旋转、倾斜等操作时使用。



混色器以及颜色样本

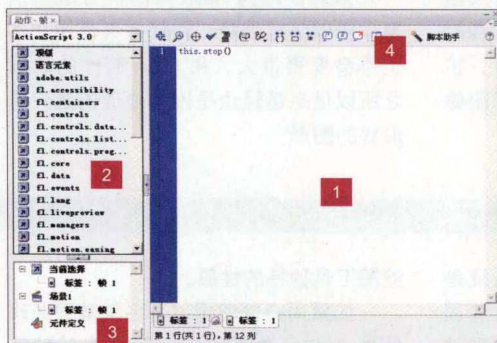
该面板是用于改变图形的颜色。而且，在混色器面板中可以进行颜色层次的设定。并且，不仅可以从窗体中选取样本颜色，也可以输入十六进制数直接指定颜色。



动作面板

动作面板是 ActionScript 特殊化后的面板。ActionScript 是 Flash 特有的脚本语言。为此，在 Flash 中准备了专用的面板。

因为在该面板中拥有支持编写代码环境的设定和编程等功能，所以相比使用文本编辑等方式的编程，它可以更方便、更迅速地进行程序编写。即使对于不太擅长编程的用户来说，这种方法也比学习其他的编程简单很多。



1. “代码编辑面板”

编写代码的区域。

2. “动作工具箱”

ActionScript 的各种项目被分类在不同种类中。从这里也可以调出所需的项目。

3. “命令导航”

显示编有代码的帧和选择的帧。

4. “输入支持功能”

上部的按钮工具群各自分别担当着支持输入代码的功能。

7. “调试选项”

该按钮用于想确认代码执行情况的位置设置“断点”。

8、9、10. “代码折叠的各种功能”

当代码变长，只显示想看的部分时，可以折叠代码或者显示代码。

11、12、13. “注释的各种功能”

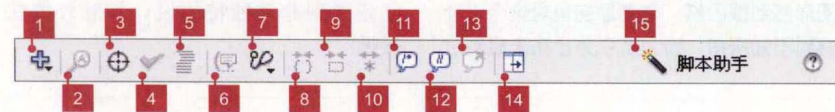
该功能用于想要在代码中添加说明文字和将文字用作注释的时候。另外也可以用于取消注释。

14. “显示/隐藏工具箱”

该功能可以转换左侧工具箱和命令导航窗口的显示/隐藏状态。

15. “脚本助手”

将显示一个用户界面，用于输入创建脚本所需的元素。



1. “将新项目添加到脚本中”

该按钮用于显示与动作工具箱相同的内容。单击后将显示菜单，可以从此处选择要添加的项目。

2. “查找”

该按钮可以查找脚本编辑区域中的字符串。

3. “插入目标路径”

该按钮可以将选择目标实例的目录(目标路径)插入到代码中。

4. “语法检查”

该按钮用于检查是否含有语法错误。

5. “自动套用格式”

该按钮用于在代码中插入字符缩进功能等，格式化代码以便于阅读。另外，当有语法错误时无法进行排版，所以也可用于检查简单的语法错误。

6. “显示代码提示”

该按钮用于根据当前的输入显示一个可能的提示选择框作为输入候选。

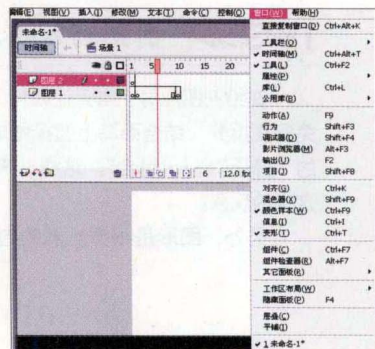
各种面板的显示方法

在这里介绍的面板是一般的面板。当然也有在其他书中没有介绍的面板。如果面板仅显示使用工具的话，操作界面就不会显得杂乱无章。在显示使用的工具，改变各种配置的同时，来寻找自己喜欢的面板设置吧！

各种面板都可以通过菜单栏的“窗口”菜单调出。用户有必要自己进行调整操作界面。不管是什么工具，如果能

好好配置，方便自己使用的话，就不会存在使用不便，浪费时间的情况了。制作工作是从操作场所的设定开始的。

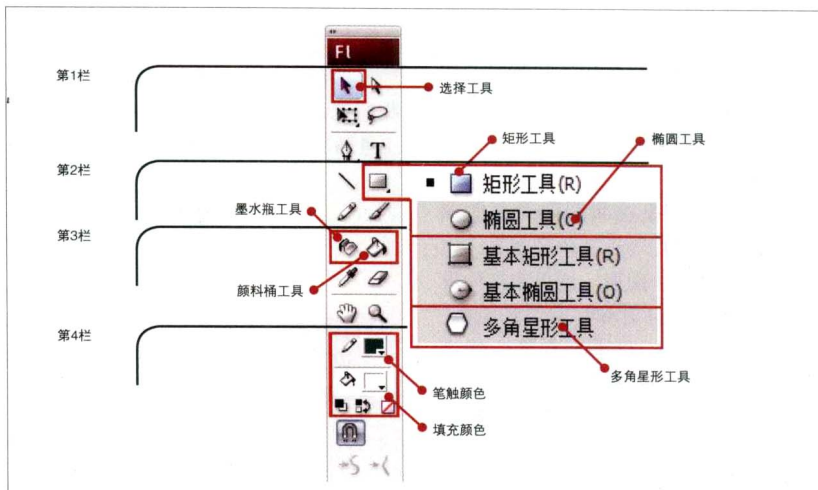
在Flash中配有各种面板，尝试打开不了解的面板是最好的学习方法。当没有必要的时候可以关闭面板。



Flash 的绘图工具

Flash 与其他绘图软件一样也具有绘图功能。使用的方法也与其他软件相同，从工具面板中选择绘图功能作成图形。之前在基本面板的讲述页面中已对工具面板进行了说明，在这里再对图形制作时频繁使用的图形制作工具和选择工具等功能进行说明。

在这里将关于第 1 栏中的“选择工具”、第 2 栏中的“基本图形工具组群”（矩形工具、椭圆工具、多边形工具）、第 3 栏中的“墨水瓶工具”和“颜料桶工具”、第 4 栏中的“笔触颜色”和“填充颜色”进行说明。为制作基本图形，只要记住这些功能就足矣应付了。



红色框圈起来的工具会在这里进行讲解。

使用 Flash 所制图形的特征

使用 Flash 作成的图形都拥有矢量形式的信息。矢量形式是在显示图像时进行内部计算来表现画面的。因此，扩大或缩小等变形操作并不会损坏图像信息。

另外，因为 Flash 内部拥有数值信息，所以信息量轻，即使多次使用文件也不会变得很大。用 Flash 制作的动画之所以信息量轻也是因为处理的是矢量形式的图形。

矢量形式的另一特征是擅长“笔触颜色”和“填充颜色”分明的图形，相反却不擅长显示照片和图画。因此，Flash 配备了从外部导入素材的功能。

图形信息的独特设计

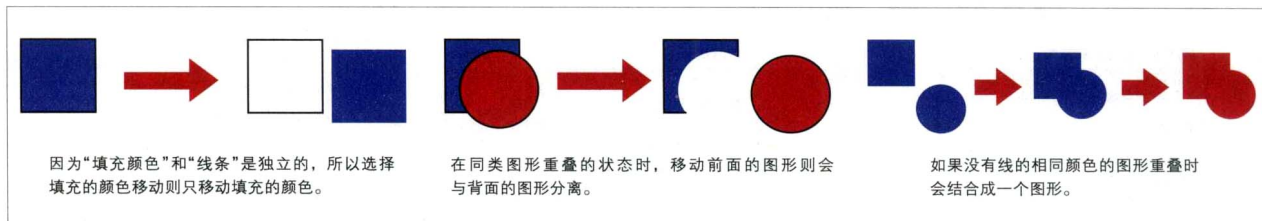
使用 Flash 制作图形的另一特征是一起画的“填充”和“线”是独立管理的。而且，同类图形在重叠或者图形被反复拆分、结合时，该图形本身采用独特的信息存储方式这一点，也是 Flash

绘图工具独特的性质。

如果用户曾使用过著名矢量图形软件 Illustrator，很有可能会对 Flash 的独特图形感到很困惑。笔者最初也对这个设计感到不知所措，因为至今为止尚未碰到过

这样的图形软件所以感到很新鲜。

如果能好好理解拥有该特征的 Flash 图形，并能熟练运用的话，会不会感到非常独特的设计和意外的快乐呢？



如果能理解图形的特征，就能理解为何将使用 Flash 绘图工具制成的图形称之为独一无二的图形了。

TIPS >> 图形的组合化

在绘制图形时，当无论如何必须重叠绘制，或者为避免反复拆分、结合而马上选择做成的图形，就执行菜单中的“修正”→“组合”。这样一来，即使重叠也可以避免这样的现象。

另外，图形是根据所画的内容依次显示在用户面前

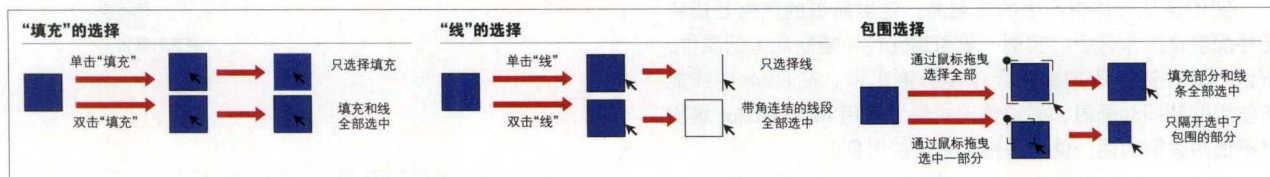
的。在组合化时，想要更换前后的图形时，选择对象图形，从菜单上选择“修改”→“合并对象”，再选择需要的项目。但是请注意合并对象的功能不能适用于一般的图形，并且组合化还可以使用菜单的“修改”→“取消组合”返回到原来图形的状态。

Flash 图形巧妙的选择方法

图形的制作有一个操作流程, 包括了“制作”→“选择”→“编辑(加工、补充、修改)”3种基本操作。在制作图形素材时, 必然会进行这种操作。即使在使用原来做好的素材时, 也不会破坏“选择”→“编辑”的操作形

式。也就是说, 有必要确切选择需要的素材。以下图示为 Flash 图标的选择方法。在 Flash 中选择图形则可以使用在第一栏中的“选择工具”。此工具本身与 Illustrator 非常相似, 但是可以选择的图形却是不同的。因此习惯使用

Illustrator 的人在适应 Flash 独特的绘画系统之前会觉得处理很不方便, 但是一旦习惯了之后, 它就是非常便于使用的工具。在开始阶段鼓励大家制作各种各样的图形, 感受一下实际的操作。



实际上, 通过绘制图形和“选择”工具的操作就可以了解所谓的独一无二的绘画系统。

基本图形的绘制方法(矩形、椭圆、多边形)

在基本图形的制作时使用第二栏的基本图形工具组中的“矩形”、“椭圆”、“多边形”工具。通过事先在属性设置中改变图形属性, 可以制作变形的图

形。只要记住仅这3种图形制作工具, 就能绘制各种各样变形的图形。以下图示是基本图形的绘制方法。当然也还配有其他的制作工具在这里先暂时省略。

如果记住了基本的操作方法, 自己就可以边制作各种图形边尝试确认就好了! 如果了解了基本操作, 那么操作本身就变得非常简单了。

	从工作区的任意位置拖曳后放置	按住“Shift”键拖曳后放置	按住“Alt”键拖曳后放置	按住“Shift”和“Alt”键拖曳后放置
矩形工具	制作长方形	制作正方形	从中心制作长方形	从中心制作正方形
椭圆工具	制作椭圆	制作圆	从中心制作椭圆	从中心制作圆
多边形工具	制作多边形	将多边形的顶点朝向垂直方向制成多边形	基本的图形都能按照相同的方法制作	



根据基本图形的属性设置来可以绘制各种各样形状的图形。

“填充”和“线”的编辑

至此本书已讲述了基本的制作工作和选择工作。这里将讲述作为基本“编辑”工作——改变图形“填充”和“线”设定的方法。

Flash 的图形中颜色“填充”和“线”是相对独立的。因为 Flash 也配有

基于这些特征的编辑功能, 所以也将对此进行讲述。

“填充”和“线”的各自内容就如图右图所示。“线”除了颜色以外还有线的种类和形状等很多可以编辑的要素, 但是“填充”只能编辑颜色。

- 填充的编辑内容 · 颜色
- 线的编辑内容 · 颜色、线宽、线的种类、线端的形状、结合(线段转角的形状)

根据属性设置而设定的工作图

“填充”和“线”的设定

铅笔符号...“线” 线宽和线种类的设定

颜料桶符号...“填充” 颜色、笔触提示、描边、一般、尖角、圆角、斜角

工具控件 图形制作时颜色的设定以及制作完成之后颜色的变更。

属性设置 图形制作之后的颜色变换以及设定内容

表示共同的颜料调色板。

线段形状 结合(线段转角的形状)

“填充”以及“线”的制作

在只有线或只有填充的图形里可以制作新的线和填充物。

工具控件“墨水瓶工具”

工具控件“颜料桶工具”

若在没有线的图形颜色填充部分单击的话, 图形外部则添加了笔触线条。

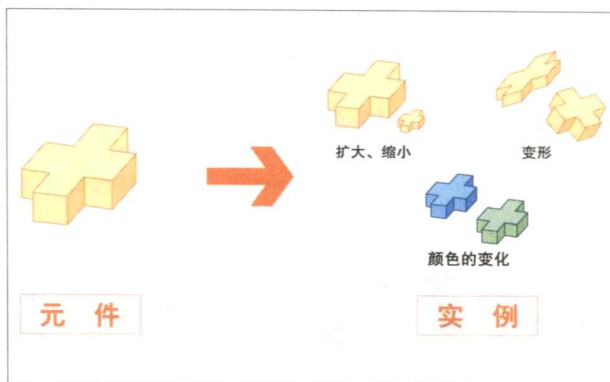
若在没有填充颜色的图形内部单击的话, 图形内部则填充了颜色。

基本的图形可以采用与 Illustrator 等其他绘图工具一样的绘制方法制作。

在 Flash 的重要思维中存在“元件”和“实例”的两大概念。只有准确理解这两个概念才能理解 Flash。虽说它们是稍有难度的两个概念，但请不要放弃准确加以了解。

元件就像零件一样，一旦做成，不管在什么影片里都能再次利用。如果做成了元件，就可以将它作为收纳在库面板中的零件来调用。

实例是从元件中产生的复制品，这里所说的产生是指将元件配置在工作区中。实例一般引用元件（复制品）的属性，所以对元件进行添加编辑操作会影响实例。在 Flash 中播放所使用的补间动画时，必须使用实例。在用 ActionScript 编程时不使用实例的话，就无法指定特定的对象。



元件的种类

在 Flash 中配有“影片剪辑元件”、“按钮元件”、“图形元件”3种元件。请根据自己的需要加以选择。

影片剪辑元件

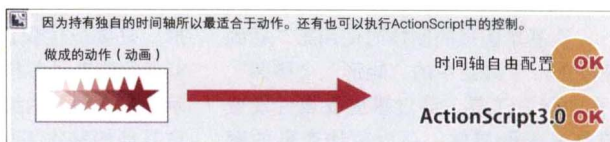
影片剪辑元件能够在单独的时间轴上使之运动，而且能够编入 ActionScript 等，它可以完成更高级的动画制作。这是使用频率最高的元件。

按钮元件

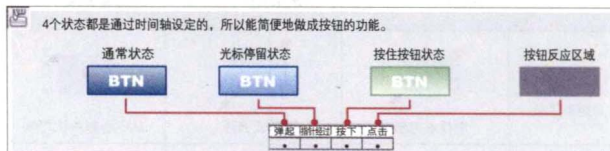
正如名字所示，这是用作按钮的元件。在使用 Flash 内容制作菜单、根据按钮添加控制功能等情况下，都要使用该元件。

图形元件

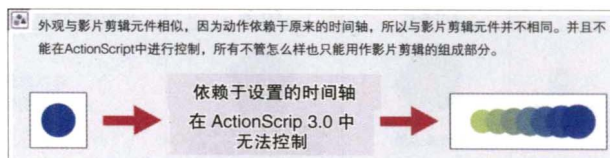
作为元件使用这一点没有改变，但是图形元件的动作依赖于配置在工作区中的时间轴，而且也不能添加 ActionScript。因为在早期的 Flash 中，时间轴的动画是主流，由于其容易理解，而被广泛地使用，但是现在影片剪辑元件能覆盖其所有功能，所以也就无需再使用图形元件了。



使用频率最高的是影片剪辑元件。不管是作为图像、按钮还是动画都可以的高通用性元件。



这是按钮功能的特殊化之后的元件。按钮元件的时间轴不同于影片剪辑元件，需事先在按钮上设定必要的项目。

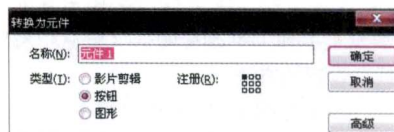


图形元件现在是利用价值很低的元件。没有特别使用的必要，所以也就很少被用到。

元件的转换（添加）

在 Flash 中显示、变换图形，制作动画时，必须将对象的图形作为“元件”添加。在 Flash 中，如不作为元件添加的话，就不能使用 ActionScript 等对对象进行操作，所以要做图首先就要变换元件。

在将图形转换成元件时，在工作区上选择要转换成元件的图形，从菜单栏中选择“修改”→“转换为元件”，制作过程中会频繁进行这样的操作。但不必每次都从菜单栏中选择“转换为元件”，而可以使用【F8】快捷键转换元件，这样就非常方便了。当出现“转换为元件”对话框时，就输入各个项目，然后点击“确定”按钮即可。当作为元件添加时，该对象就被收纳在库面板中。并且，添加了元件的工作区上的图形被认为是实例。如果查看它的属性设置，即可知道它是实例。



在将制成图形变换成元件时，从菜单栏中选择“修改”→“转换为元件”。在上图“转换为元件”对话框中设定制成元件的“名称”、“类型”并设定“注册”点，然后点击“确定”按钮。“注册”点是指旋转元件时的轴心点，注册点信息在这之后也可以改变。



在 Flash 中使用的元件都可以在库中自动添加。因此通过确认库也可以确认元件的名称和形状。想要更改元件时，也可以从库中选择想要变更的元件。请注意，如果改变元件的话，那相对应的实例也都会发生改变。

元件的使用

使用元件时，可以从库面板中将元件拖放到工作区上。但是所配置的元件并非是元件，而是该元件的一个实例。在工作区上可以配置无数个实例。元件和实例就像是零件和零件复制品的关系。若元件是零件的话，实例就是它的复制品。



将元件配置到工作区上的对象可以叫做实例。

实例的属性

实例可以从属性设置和变形面板中改变颜色和形状。就像相同父母的孩子，不同的孩子有不同的特征那样，从同一个元件做成的实例可以改变各自不同的特征。即使实例的属性改变了，元件也不受任何影响。



对于从同一个元件做成的实例添加各自不同的变化。即使改变了实例，添加在库中的元件自身也不会发生改变。

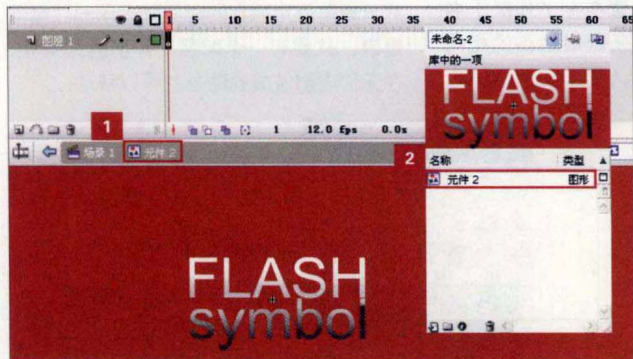
元件的编辑 1

在编辑实例的前身元件时，从库面板中双击想要编辑的元件标签或图像部分，进入元件的编辑页面进行编辑。但是元件的编辑界面和工作区外观并不会改变，所以请仔细确认显示的是工作

区还是元件的编辑界面。

在编辑条中显示的是界面的位置 1。“场景”1 指的是工作区。场景的后面如果显示元件名称的话，表示正处于元件编辑界面。习惯此类操作之后也可以在

元件中将实例设置为嵌套。那时请参考该编辑条。在嵌套之后在右侧将会显示要素名称。



编辑元件时，打开库面板，双击元件 2，就进入了元件的编辑界面。请务必确认一下编辑条 1。



编辑元件的话，包括已经完成的对象，所有的实例都将被改变。在上述的例子中，因为将元件的文本“SYMBOL”变成了“FLASH”，所以所有实例的文字都发生了变化。请不要忘记实例通常是引用元件的。

元件的编辑 2

双击工作区上的实例也可以显示元件的编辑界面。这种方法适用于在显示工作区的状态下编辑元件，所以想要一边确认内容显示位置，一边对元件进行操作时，建议采用该方法。细心观察就会发现关于元件的编辑界面，也会不经意间编辑和删除元件，所以请经常确认编辑条，确认是在编辑实例还是在编辑元件之后，再进行操作。实际上通过反复编辑和删除也可以了解元件和实例之间的关系。刚开始时也会出现始料未及的状况，但是因为“只有失败才是掌握的捷径”，所以请在测试环境中尝试各种操作。



双击实例，就会保持工作区内容的显示并转换至元件的编辑界面。